Karl Weber
Carl Wolfram Türschmann

FOSBIC Compiler

Ein BASIC-Compilersystem auf FORTRAN-Basis

Haupt

Die Reihe BASIC-Software will

- allgemein
  BASIC-Einsatzmöglichkeiten in Forschung, Praxis und Unterricht
  aufzeigen,

- Sozial- und Wirtschaftswissenschaftlern theoretisch fundierte Applikationsprogramme aus den Gebieten von
  Statistik
  Operations Research
  Betriebswirtschaftslehre
  Volkswirtschaftslehre
  vermitteln,

- Informatiker über
  ANSI/ECMA BASIC-Standards und BASIC-Compiler
  orientieren.

In Vorbereitung sind folgende Bände

BASIC-Transportabilität
BASIC-Anfängerprogramme
BASIC-Mittelschulprogramme
BASIC OR-Programme 1: Prognosemodelle
BASIC OR-Programme 2: Lagerhaltungs- und Warteschlangenmodelle
BASIC BWL-Programme 1: Betriebswirtschaftliche Rechnungsführung
BASIC BWL-Programme 2: Marketing
BASIC VWL-Programme 1: Keynes-Modelle

BASIC-Software Band 1

Herausgegeben von Prof. Dr. Karl Weber

Karl Weber
Carl Wolfram Türschmann

FOSBIC-Compiler

Ein BASIC-Compilersystem auf FORTRAN-Basis

Verlag Paul Haupt Bern und Stuttgart

Die Autoren:
Karl Weber, Dr. oec. publ., M. S. in Accountancy.
1948–1954 Studium der Volks- und Betriebswirtschaftslehre an der Universität Zürich; 1955 Promotion; 1953–1956 Assistent am Handelswissenschaftlichen Seminar der Universität Zürich; 1956–1963 wissenschaftlicher Mitarbeiter am Betriebswissenschaftlichen Institut der ETH Zürich; 1959 Studium an der London School of Economics and Political Science; 1963–1964 Instructor of Accountancy an der University of Illinois in Urbana; 1964–1967 Assistant Professor of Accountancy an der University of Illinois; 1966 Habilitation an der Universität Zürich; 1966 Fellow in Accountancy an der University of Illinois; 1967–1969 Associate Professor of Accountancy an der University of Illinois; seit 1969 Professor für Betriebswirtschaftslehre an der Justus Liebig-Universität Giessen; seit 1977 Geschäftsführender Direktor des Zentrums für Datenverarbeitung der Justus Liebig-Universität Giessen.
Publikationen: Dividendenpolitik, Zürich 1955; Amerikanische Standardkostenrechnung, Winterthur 1960; Planung und Planungsrechnung in Schweizer Unternehmungen, Bern 1965; The Evolution of Direct Costing, Urbana 1966; Amerikanisches Direct Costing, Bern 1970; Planspiel Elektrizitätswirtschaft (zusammen mit H. Lienhard und F. Steiger). Bern und Stuttgart 1975.

Carl Wolfram Türschmann, Dipl.-Ing., Dipl.-Oec.
1964–1970 Studium des Maschinenbaus an der Technischen Hochschule Darmstadt; 1970–1973 Studium der Wirtschaftswissenschaft an der Justus Liebig-Universität Giessen; 1970–1972 akademischer Tutor im Fachbereich Wirtschaftswissenschaft der Justus Liebig-Universität; seit 1972 Wissenschaftlicher Mitarbeiter an der Professur für Betriebswirtschaftslehre V der Justus Liebig-Universität Giessen; seit 1973 freier Mitarbeiter an der Abteilung für experimentelle Kardiologie am Max Planck-Institut für physiologische und klinische Forschung (W. G. Kerckhoff-Institut), Bad Nauheim; seit 1976 Lehrbeauftragter für computergestützte Methoden der empirischen Sozialforschung an der Ruhr-Universität Bochum (Institut für Sportwissenschaft).
Publikationen: OR-Modellbank in BASIC, in: Proceedings in Operations Research 5, Würzburg 1976; BASIC, 2 Bde. (zusammen mit K. Weber), Bern und Stuttgart 1977. Mehrere Artikel, zusammen mit W. Schaper, W. Flameng und B. Wüsten, in Verh. Dtsch. Ges. Kreislaufforschung (ab 1975), European Journal of Physiology (1976), American Heart Journal (1976) und Circulation Research (1976).

ISBN 3–258–02580–0

Vorwort

Die rechnerische Behandlung komplexer Pobleme erfolgt in allen Wissenschaftsgebieten in zunehmendem Masse unter Verwendung elektronischer Rechenanlagen, zu deren Programmierung an Hochschulen bisher zumeist FORTRAN, COBOL oder ALGOL verwendet wurden. Die Erlernung dieser Programmiersprachen erfordert erfahrungsgemäss einen erheblichen Zeitaufwand, so dass die EDV-orientierte Behandlung komplexer Probleme in der Regel erst in höheren Semestern erfolgen kann.
Im Gegensatz zu den vorerwähnten Programmiersprachen erweist sich BASIC als relativ leicht erlernbar und damit bereits im Grundstudium voll verwendbar. BASIC kommt damit tatsächlich die Funktion eines „**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode" zu und vermochte sich aus diesem Grunde an amerikanischen Hochschulen – insbesondere als Dialogsprache – in weitgehendem Masse durchzusetzen. Dagegen fand BASIC an europäischen Hochschulen bisher vergleichsweise geringe Verbreitung, was teilweise auf das Fehlen eines auch für den Stapelbetrieb geeigneten und auf bestehende Grossrechenanlagen übernehmbaren BASIC-Compilers zurückzuführen ist.
Mit der Publikation der vorliegenden Arbeit wird der letzterwähnte Mangel insofern beseitigt, als ein hochkompatibler BASIC-Compiler auf FORTRAN-Basis allgemein verfügbar gemacht wird. Dieser – an der Professur für Betriebswirtschaftslehre V der Justus Liebig-Universität Giessen – von C. W. Türschmann entwickelte FOSBIC (**FO**RTRAN **S**imulated **B**ASIC **I**nterpretive **C**ompiler)-Compiler gestattet es, BASIC auf jeder Anlage mit FORTRAN-IV-Compiler in kürzester Zeit zu implementieren und fremde BASIC-Programme in praktisch unveränderter Form zu übernehmen. Weiterhin besteht die Möglichkeit, die Grundlagen des Compiler-Compiler-Baus mit FOSBIC leicht verständlich darzustellen und weiter zu erforschen. So gilt denn auch der FOSBIC-Compiler nach Professor William F. Sharpe (Stanford University) als „a system of the most modern type".
Um die Verbreitung der lernorientierten Programmiersprache BASIC möglichst weitgehend zu fördern, werden FORTRAN-Source-Decks des FOSBIC-Compilersystems auf unbeschrifteten 80-Spaltenlochkarten oder auf ungelabelten Magnetbändern im BCD-Code zu Selbstkosten an Interessenten abgegeben. Kontaktadresse:

Professur für Betriebswirtschaftslehre V
Justus Liebig-Universität Giessen
Licher Strasse 74
D-6300 Giessen

Die Autoren bitten bei Übernahme von FOSBIC um Information über Maschinentyp, Implementationsdatum, Übernahmegrad des FOSBIC-Compilersystems, eventuell aufgetretene Probleme und um den Austausch entwickelter Programme im Rahmen einer BASIC-User-Group.

Karl Weber
Carl Wolfram Türschmann

Einleitung

Die vorliegende Arbeit schliesst unmittelbar an das in der UTB-Reihe veröffentlichte Werk

BASIC. Lehr- und Handbuch der Programmiersprache BASIC mit wirtschaftswissenschaftlichen Anwendungsbeispielen. UTB 588 und 589, Bern und Stuttgart 1977

an.

Während in der vorerwähnten Publikation Struktur und Einsatzmöglichkeiten der Programmiersprache BASIC behandelt werden, befasst sich der dritte Band der BASIC-Trilogie mit BASIC-Compilern auf FORTRAN-Basis.
Das erste Kapitel vermittelt einen konzisen Überblick über die Entwicklung der Programmiersprache BASIC.
Im zweiten Kapitel werden die BASIC-Compiler UWBIC (**U**niversity of **W**ashington **B**ASIC **I**nterpretive **C**ompiler), FOSBIC (**FO**RTRAN **S**imulated **B**ASIC **I**nterpretive **C**ompiler) und VIEWIT **(View It)** durch entsprechende Hinweise auf Ursprung, Sprachumfang und Compilerstruktur charakterisiert.
Im dritten Kapitel wird das FOSBIC-Compilersystem dargestellt, wobei speziell auf FOSBIC-Entwicklung, FOSBIC-Struktur und FOSBIC-Implementation eingetreten wird.
Literaturverzeichnis und FOSBIC-Programmlisten beschliessen den Band.

1 Programmiersprache BASIC

11 BASIC-Ursprung

BASIC (**B**eginner's **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode) ist eine benutzerorientierte Programmiersprache.[1] Sie wurde in ihrer Grundform 1963/64 von John G. Kemeny und Thomas E. Kurtz – mit Unterstützung der amerikanischen National Science Foundation – am Dartmouth College entwickelt,[2] nachdem Versuche zur Schaffung einer lernorientierten Version von FORTRAN und ALGOL fehlgeschlagen waren.

Bezüglich FORTRAN bestand damals noch keine Möglichkeit zur Vermeidung mehrerer Variablentypen (real/integer), des Formatstatements und der wenig eleganten zweiseitig bedingten Sprunganweisung. Bei ALGOL konnten die aus der Struktur der Laufanweisung sich ergebenden komplexen Statementformen nicht umgangen werden.[3]

Angeregt durch die benutzerfreundlichen Programmiersprachen JOSS und CORC[4] wurde bei der Ursprungsversion von BASIC die Zahl der Statements gering und deren Struktur möglichst einfach gehalten. Dadurch sollte die leichte Erlern- und Benutzbarkeit von BASIC sichergestellt werden.

Die Grundversion von BASIC umfasste nur die aus Abbildung 1 ersichtlichen Statements. Bei der Statementstrukturierung wurde Wert darauf gelegt, starke Unterschiede gegenüber FORTRAN und ALGOL zu vermeiden, um den BASIC-Benutzern den Zugang zu den erwähnten höheren Programmiersprachen zu erleichtern. Mit BASIC wurde damit nicht nur eine leicht erlernbare, sondern auch eine leicht übersetzbare Programmiersprache geschaffen. „BASIC was meant to be a very simple language to learn and also one that would be easy to translate."[5]

1 Vgl. Karl Weber und Carl Wolfram Türschmann. BASIC: Lehr- und Handbuch der Programmiersprache BASIC mit wirtschaftswissenschaftlichen Anwendungsbeispielen. UTB 588 und 589, Bern und Stuttgart 1977.
2 John G. Kemeny and Thomas E. Kurtz. BASIC. User's Manual. [1st ed.,] Hanover 1964; John G. Kemeny and Thomas E. Kurtz. BASIC Programming. 2nd ed., New York 1971.
3 Stephan J. Garland. Dartmouth BASIC: A Specification. Hanover 1973.
4 Zur Entwicklung der Programmiersprachen vgl. Jean E. Sammet. Programming Languages: History and Future. Communications of the ACM 15 (1972) 7, July, p. 601–610.
5 Jean E. Sammet. Programming Languages: History and Fundamentals. Englewood Cliffs 1969, p. 229.

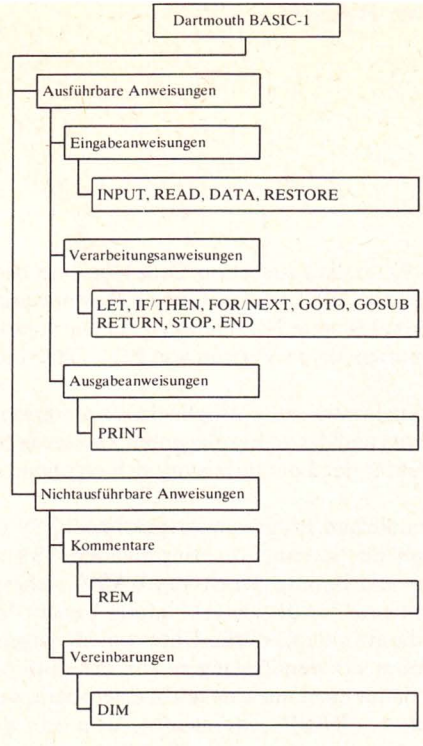


Abbildung 1. Dartmouth BASIC-1

Gegenüber FORTRAN und ALGOL weist BASIC – hier speziell Dartmouth BASIC-1 – folgende Besonderheiten auf:

- INPUT-Statement: Es lässt eine formatfreie Dateneingabe (über periphere Eingabestationen) zu.
- FOR/NEXT-Statement: Das FOR-Statement ist ebenso klar aufgebaut, wie das entsprechende ALGOL-Statement. Im Gegensatz zu ALGOL braucht die Schrittweite von + 1 nicht explizit angegeben zu werden. Im Unterschied zu FORTRAN ist die Laufvariable nicht auf Ganzzahlen (integers) beschränkt; negative Schrittweiten sind zulässig.
- PRINT-Statement: Mit seiner Hilfe erfolgt die Datenausgabe in feststehender Form, über 5 Druckbereiche mit je 15 Druckstellen.

- DIM-Statement: Es dient zur expliziten Dimensionierung von Vektoren und Matrizen und ist nur dann zu verwenden, wenn die implizite Dimensionierung (0:10) resp. (0:10, 0:10) nicht genügt.
- Der benutzerfreundliche Charakter von BASIC kommt auch in der Definition der Variablennamen deutlich zum Ausdruck. Diese dürfen generell nur einen Buchstaben oder einen Buchstaben mit nachfolgender Ziffer umfassen. Die Benutzer müssen mithin nicht mit syntaktischen Regeln belastet werden und brauchen keine „reservierten Worte" zu beachten.

Struktur und Verwendung einiger in Dartmouth BASIC-1 verfügbarer Statements geht aus dem folgenden Programm hervor; zu beachten ist, dass jedes Statement eine Nummer aufweisen muss.

```
 10 REM SORTIERPROGRAMM
 20 DIM X(100)
 30 READ N
 40 FOR I=1 TO N STEP 1
 50 READ X(I)
 60 NEXT I
 70 FOR I = 1 TO N-1
 80 FOR J = I+1 TO N
 90 IF X(I) <= X(J) THEN 130
100 LET T = X(I)
110 LET X(I) = X(J)
120 LET X(J) = T
130 NEXT J
140 NEXT I
150 FOR I = 1 TO N
160 PRINT X(I)
170 NEXT I
180 DATA 12
190 DATA 10,-5.5,9,8,-7,20,3.5,8,9,-99.99,20,12
210 END
```

12 BASIC-Entwicklung

BASIC wurde am Dartmouth College im Rahmen eines hochschuleigenen Timesharing-Systems (GE-265) verwendet.[6] Die mit diesem System erzielten Erfolge veranlassten die General Electric Company 1965 zur Entwicklung eines eigenen Timesharing Service-Netzes, unter bevorzugter Verwendung von BASIC als Programmiersprache.[7]

Übernommen und weiterentwickelt wurde BASIC auch von IBM. IBM BASIC umfasst eine grössere Zahl eingebauter Funktionen, vordefinierte Konstante (internal constants) und bietet insbesondere auch die Möglichkeit zum Rechnen mit einfacher oder doppelter Genauigkeit, zur Drucksteuerung mit Formatangaben und zur externen Datenspeicherung.

Weiterhin wurde BASIC als Dialogsprache für Computersysteme der Firma Philips-Electrologica B. V., National Cash Register Company u. a. adaptiert.

6 Vgl. J. Daniel Couger. Computers and the Schools of Business. Boulder 1967, p. 68–75.

7 Zur BASIC-Entwicklung vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 11, 12, 13 und Bd. 2, Abs. 12.

Ausgehend von Dartmouth BASIC – und mit Unterstützung der National Aeronautics and Space Administration – wurde an der University of Maryland 1968 der UNIVAC BASIC Processor (UOM BASIC) entwickelt. Er war primär für Dialogbetrieb (conversational use) konzipiert, aber auch im Stapelbetrieb (batch processing) verwendbar und zeichnete sich durch verschiedene Weiterentwicklungen (mehrzeilige Benutzerfunktionen, Alternativformen für FOR/NEXT-Statement usw.) aus.
In der Folge übernahm die Sperry Rand Corporation den UOM BASIC-Compiler für das Computersystem UNIVAC 1100 und stellt diesen nunmehr – unter der Bezeichnung UBASIC – einem weiten Benutzerkreis zur Verfügung.
Für Alternativbetrieb (Dialog/Batch) sind die BASIC-Versionen der Burroughs Corporation, Control Data Corporation, Siemens Aktiengesellschaft u. a. geeignet; verwiesen sei insbesondere auch auf die BASIC-Version von Xerox Data System, die auch der Compagnie Internationale pour l'Informatique als Grundlage für eigene Weiterentwicklungen (SIRIS 7/ SIRIS 8) diente.
Speziell auf den Batchbetrieb ausgerichtet sind die auf FORTRAN-Basis entwickelten UWBIC- und FOSBIC-Compiler.[8]
Besonders wichtig ist, dass BASIC (neben FORTRAN) in zunehmendem Masse für Minicomputer als Programmiersprache benutzt wird, so dass die Verwendung von Maschinen- oder Assemblersprachen bei Kleincomputern nicht mehr unbedingt erforderlich ist.[9]
Die BASIC-Weiterentwicklung liessen auch Dartmouth BASIC nicht unberührt. 1971 wurde die sechste BASIC-Version freigegeben. Sie ist nicht mehr ausschliesslich für Anfänger bestimmt, sondern weist eine Reihe von Sonderanweisungen – insbesondere Statements zur Matrizen- und Fileverarbeitung – auf.[10]
Vgl. Abbildung 2.
Weitere Entwicklungstendenzen gehen dahin, den BASIC-Sprachumfang durch spezielle PLOT-Befehle auszuweiten.[11]

13 BASIC-Verwendung

Am Dartmouth College vermochte sich BASIC – im Rahmen des dortigen Timesharing Systems – sehr rasch auszubreiten. Ähnlich verlief die Entwicklung an anderen amerikanischen Hochschulen.
Besonder häufig wird BASIC im Rahmen des wirtschaftswissenschaftlichen Studiums verwendet. An zahlreichen Business Schools wird der Nachweis von BASIC-Programmierkenntnissen als Voraussetzung für

8 Vgl. Abs. 21 und 22.
9 Vgl. L. C. Hobbs and Richard A. McLaughlin. Minicomputer Survey. Datamation 20 (1967) 7, July, p. 50–61; Datapro Research Corporation. All About Minicomputers. Delran 1976.
10 Stephan V. Waite and Diane G. Mather. BASIC. 6th ed. Hanover 1971.
11 Arthur Luehrmann. An Elaboration of Some Thoughts on a Graphical Syntax in BASIC. Hanover 1974.

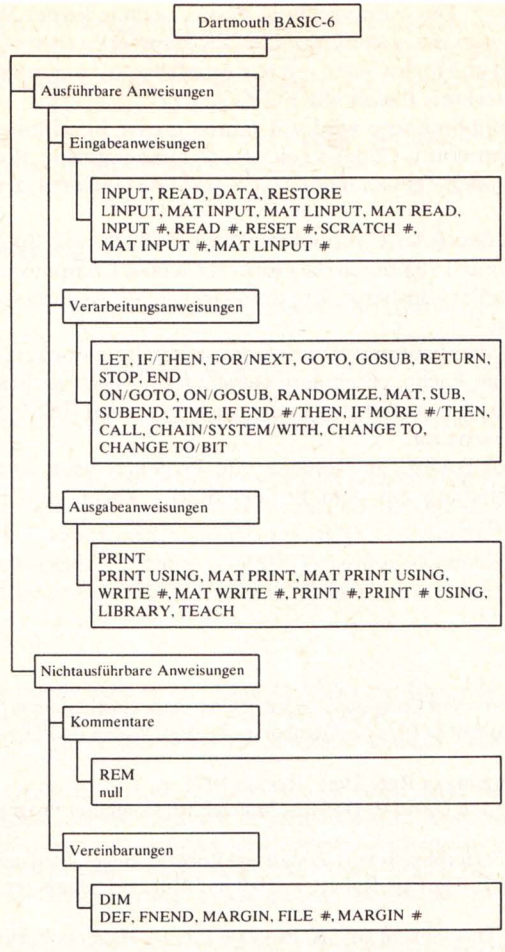


Abbildung 2. Dartmouth BASIC-6

den Studienabschluss verlangt.[12] Diese Entwicklung scheint noch in keiner Weise abgeschlossen zu sein; nachdem sich BASIC bereits gegenüber allen anderen Dialogsprachen (insbesondere APL) durchzusetzen vermochte, dürfte – jedenfalls an Hochschulen mit leistungsfähigen Programmbibliotheken – BASIC bald einen ebenso grossen Benutzerkreis finden wie FORTRAN.[13]

Dem Aufbau von Programmbibliotheken wird seit Jahren grosse Beachtung geschenkt; Pionierarbeiten wurden diesbezüglich am Dartmouth College geleistet.[14] Umfangreiche Programmbibliotheken stehen aber auch an anderen Hochschulen – genannt seien Georgia State University und Michigan State University – zur Verfügung.

Besonders hervorzuheben ist, dass BASIC vermehrt in der Lehrbuch- und Spezialliteratur verwendet wird. Diese Entwicklung wurde bereits 1965 durch die General Electric Company eingeleitet[15] und in der Folge durch weitere Computerhersteller – insbesondere durch freizügige Abgabe von BASIC-Programmsammlungen – wesentlich gefördert.[16]

Der Einsatz von BASIC blieb aber keineswegs auf Universitäten beschränkt; BASIC wird auch an anderen Lehranstalten, insbesondere an Fachhochschulen, verwendet.[17] Auch an Management-Ausbildungszentren – genannt sei das Europäische Institut für Unternehmensführung (INSEAD) – wird vorzugsweise mit der Dialogsprache BASIC gearbeitet.

Als Programmiersprache wird BASIC im Rahmen von Projekten zum computergestützten Unterricht eingesetzt, wobei – in Verbindung mit dem Projekt Solo – sogar eine besondere BASIC-Variante (NEWBASIC) entwickelt wurde.[18]

12 Report on the 3rd Triennial Survey of Computer Uses and Computer Curriculum in Schools of Business. Computing Newsletter for Schools of Business 8 (1974) 2, October, p. 2.– Vgl. Weber und Türschmann, BASIC, Bd. 2, Tabelle 3.

13 Don Cassel. BASIC Programming in Real Time. Reston 1975, p. V.

14 Vgl. etwa: J. Peter Williamson and David H. Downes. Manuals for Computer Programs in Finance and Investments. 3rd ed., Hanover 1973.

15 Clifford H. Springer, Robert E. Herlihy, Robert T. Mall and Robert I. Beggs. Statistical Inference. Homewood 1966; Clifford H. Springer, Robert E. Herlihy, Robert T. Mall and Robert I. Beggs. Probabilistic Models. Homewood 1968.

16 Vgl. etwa: Hewlett-Packard. Time Shared BASIC Program Library Handbook. November 1971 Supplement. Cupertino 1971.

17 Vgl. Hans Rehbein. BASIC – Leicht gemacht. Eine BASIC-Einführung und 50 vollständige Übungsaufgaben mit Lösungen. 2. Aufl., Düsseldorf 1974.

18 Com-Share Inc. NEWBASIC Reference Manual. Pittsburgh 1971. Thomas A. Dwyer. Teacher/Student Authored CAI Using NEWBASIC System. Communications of the ACM 15 (1972) 1, January, p. 21–28.

An amerikanischen Hochschulen wird die Programmiersprache BASIC auch in der Forschung verwendet. Dies gilt nicht zuletzt für das Gebiet des Operations Research,[19] wobei selbst „the cumbersome DYNAMO of Forrester into the computer language BASIC" umgesetzt und zur Simulation von Weltmodellen verwendet wurde.[20]
Über den Einsatz von BASIC in der Praxis liegen nur wenige Berichte vor;[21] offensichtlich vermag sich BASIC auch hier in zunehmendem Masse durchzusetzen.[22]

14 BASIC-Standardisierung

Seit einiger Zeit wird von amerikanischen Hochschulen ein interuniversitärer BASIC-Programmaustausch angestrebt. Diesbezüglich ergaben sich zunächst insofern gewisse Schwierigkeiten, als mit verschiedenen BASIC-Versionen gearbeitet wurde und die freigegebenen Programme nicht voll austauschfähig waren.
Diese Situation führte 1972 – unter Verzicht auf Vollausschöpfung mancher BASIC-Dialekte – zur Konzeption von ACT Transport BASIC, eines bewusst auf „interdialect translatability" ausgerichteten BASIC-Standards.[23]
ACT Transport BASIC wurde in der Folge – beispielsweise im Rahmen des Projektes CONDUIT[24] – weitgehend beachtet und regte zu weiteren Standardisierungsbemühungen an.[25]
Diese Bestrebungen führten 1974 zur Konstituierung einer BASIC-Standardisierungskommission im Rahmen des American National Standards Institute (ANSI). Von dieser Kommission wurden bisher – für Elementar-BASIC – folgende Richtlinien *(ANSI-Minimal BASIC)* erarbeitet:[26]

19 A. O. Converse. Optimization. New York 1970; Claude McMillan and Richard F. Gonzales. Systems Analysis. A Computer Approach to Decision Models. 3rd ed., Homewood 1973.
20 John Wilkinson. Resources in a Stable Society *Or* The Trashman Cometh. Technological Forecasting and Social Change 7 (1975) 1, p. 14.
21 Vgl. Bull General Electric. Der B-GE Time-Sharing Service im Urteil seiner Benutzer. Köln 1970.
22 Vgl. Bennet P. Lientz. A Comparative Evaluation of Versions of BASIC. Communications of the ACM 19 (1976) 4, April, p. 175: „A survey of 101 organizations revealed that while Fortran was used by 65 percent of the respondents, BASIC was second with 49 percent." – Über eine frühere Untersuchung (mit geringerer Verbreitung von BASIC) vgl. Andreas S. Philippakis. Programming Language Usage. Datamation 19 (1973) 10, October, p. 109–110, 114.
23 Gerald L. Isaacs. Interdialect Translatability of the BASIC Programming Language. Iowa City 1972.
24 Trinka Dunnagan. CONDUIT Technical Transport Guidelines. Iowa City 1973.
25 G. M. Bull, W. Freeman and S. J. Garland. Specification for Standard BASIC. Manchester 1973.
26 ANSI X3J2 BASIC Standard Committee. Proposed American National Standard for Minimal BASIC. Hanover 1976, unter Mitberücksichtigung der Änderungen bis 31. Dezember 1976.
– Vgl. auch: Weber und Türschmann, BASIC, Bd. 1 und 2.

- BASIC ist als eine zeilenorientierte Programmiersprache zu konzipieren, die mindestens die folgenden Kennworte umfasst: DATA, DEF, DIM, END, FOR, GOSUB, GOTO, IF/THEN, INPUT, LET, NEXT, ON/GOTO, PRINT, RANDOMIZE, READ, REMARK, RESTORE, RETURN, STOP.
- Mit Hilfe von BASIC-Statements sollen numerische und alphanumerische Konstante und Variable verarbeitet werden können, wobei numerische Werte in den Grenzen 1E − 38 bis 1E + 38 und Strings mit 0 bis 18 Zeichen darstellbar sein sollen.
- Als Operationszeichen sind + (Addition), − (Subtraktion), * (Multiplikation), / (Division) und ^ (Exponentiation, Involution) vorzusehen.
- Als Standardfunktionen (implementation-supplied functions) gelten: ABS, ATN, COS, EXP, INT, LOG, RND, SGN, SIN, SQR, TAN, wobei RND ohne Argumentangabe erscheint.

Bezüglich der einzelnen Minimal BASIC-Statements wurde vor allem folgendes festgelegt:

- INPUT-Statement: Dateneingabe soll formatfrei erfolgen können.
- DATA-Statement: Es soll nur eine Gesamtdatei geschaffen werden, ohne Differenzierung zwischen numerischen und alphanumerischen Daten.
- LET-Statement: Es soll auch das implizite let-Statement zulässig sein; mehrfache Wertzuweisung über ein LET-Statement sind in Minimal BASIC nicht vorgesehen.
- IF-Statement: Als Vergleichsoperatoren sind vorzusehen: < (kleiner), <= (kleiner/gleich), = (gleich), > (grösser), >= (grösser/gleich), <>(nicht gleich).
  Alphanumerische Vergleiche sind auf dem ASCII-Code basierend durchzuführen.
- FOR/NEXT-Statement: Schrittweiten können positiv oder negativ sein; bei Schrittweite + 1 ist STEP-Angabe nicht erforderlich.
- ON/GOTO-Statement: Die zur Spezifikation der Sprungstelle verwendeten Werte arithmetischer Ausdrücke werden gerundet; falls keine der angegebenen Zeilennummern anzuspringen ist, erfolgt Programmabbruch mit Fehlermeldung.
- PRINT-Statement: Zur Drucksteuerung sollen Komma, Strichpunkt und Tabulator verfügbar sein; normalerweise wird die gesamte Druckbreite in fünf gleich breite Druckbereiche eingeteilt.
- DIM-Statement: Unter- und Obergrenze der Indizes sollen angegeben werden können; falls keine Untergrenze angegeben ist, wird diese mit Null angenommen (default lower bound set to zero).
- DEF-Statement: Mehrzeilige Benutzerfunktionen sind nicht vorgesehen.
- RANDOMIZE-Statement: Wahl zufälliger Startbasen für Zufallszahlengeneratoren; Argumentangabe ist nicht notwendig.
- END-Statement: Zur Kennzeichnung des Programmendes unbedingt erforderlich.

Weiterhin wurde festgelegt, dass die Standards für Minimal BASIC nicht nur für den Timesharing-Betrieb (interactive mode), sondern auch für Stapelbetrieb (batch mode) gelten sollten.

Das zweite Projekt der ANSI BASIC-Kommission betrifft die Erarbeitung von Standards für ausgeweitetes BASIC (Enhancement BASIC), wobei String-, File- und Matrizenverarbeitung sowie Formatierung im Vordergrund des Interesses stehen. Vgl. Abbildung 3.

Mit dem Problem der BASIC-Standardisierung befasst sich seit 1974 auch die European Computer Manu-

facturers Association (ECMA).[27] Die Arbeiten des BASIC Komitees TC 21 bestanden bisher im wesentlichen in einer Überarbeitung von Arbeitspapieren der ANSI BASIC Kommission X3J2; weiterhin wurden entsprechende Arbeiten zu den File-, Matrizen- usw. Anweisungen aufgenommen.
Aufgrund der engen Zusammenarbeit zwischen ANSI und ECMA BASIC-Kommissionen kann auf längere Sicht mit einer internationalen Standardisierung von Minimal BASIC gerechnet werden.
Eine ähnliche Standardisierung wird auch vom Niederländischen Normeninstitut angestrebt. Die von dieser Organisation in den Jahren 1973–75 erarbeiteten Richtlinien zu Elementar-BASIC (Nederlandse praktijkrichtlijn voor elementair BASIC[28]) werden von den ANSI/ECMA BASIC-Kommissionen als wertvoller Beitrag zur BASIC-Standardisierung anerkannt.

27 European Computer Manufacturers Association (ECMA). Memento 1975. Geneva 1975. p. 27.
28 Nederlands Normalisatie-instituut. Automatische gegevensverwerking. Programmeertaal Elementair Basic. Rijswijk 1975; Nederlands Normalisatie-instituut. Proposal for ISO-Standard on Programming Language Elementary BASIC. Rijswijk 1976.

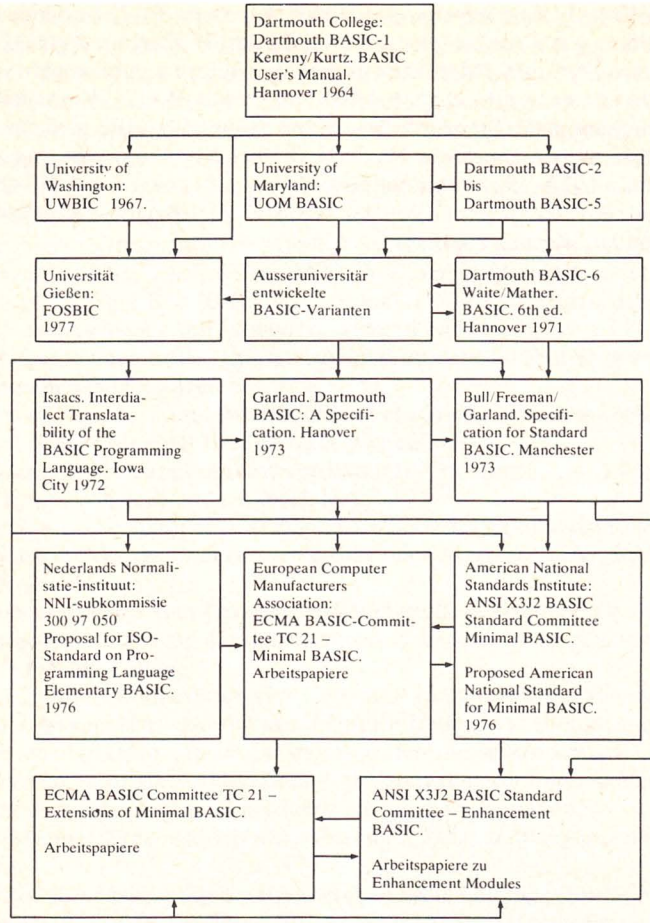


Abbildung 3. BASIC-Diversifikation und -Standardisierung

2 BASIC-Compiler auf FORTRAN-Basis

21 UWBIC (University of Washington BASIC Interpretive Compiler)

211 Ursprung

UWBIC wurde an der Graduate School of Business Administration der University of Washington unter der Leitung von William F. Sharpe entwickelt und 1967 auf der dortigen Datenverarbeitungsanlage (IBM 7094) implementiert.[1] UWBIC war vollständig in FORTRAN IV programmiert und konnte damit leicht auf mittlere und grössere Datenverarbeitungsanlagen beliebigen Typs übernommen werden. UWBIC fand denn auch an amerikanischen Hochschulen relativ rasch Verbreitung.[2]

212 Sprachumfang

Der Sprachumfang von UWBIC entspricht ANSI Minimal BASIC nicht vollständig. Insgesamt sind 16 Elementaranweisungen
REM, DIM, DATA, data, READ, PRINT, PAGE, LET, GOTO,
IF/THEN, IF/GOTO, FOR/NEXT, STOP, END, GOSUB, RETURN
und 2 Sonderanweisungen
PRINT ALL, DUMP
verfügbar.
Die Struktur der vorerwähnten BASIC-Statements wurden an anderer Stelle ausführlich behandelt.[3] Erwähnt sei, dass DATA- und data-Statements nur numerische Konstante in I- und F-Format enthalten und dementsprechend auch READ-Statements nur numerische Variablenlisten umfassen können. Die Textverarbeitung ist auf alphanumerische Konstante innerhalb des PRINT-Statements beschränkt.

1 William F. Sharpe. University of Washington BASIC Interpretive Compiler UWBIC. Seattle 1967.
2 Richard L. Nolan. Introduction to Computing Through the BASIC Language. 2nd ed., New York 1974, p. 271: „The UWBIC system is currently operating at a number of computer installations using hardware such as IBM System 360 and 370 computers, Univac 1108 and Control Data 3600 and 6600."
3 Weber und Türschmann, BASIC, Bd. 1 (Elementaranweisungen) und Bd. 2 (Sonderanweisungen).

Von den nach ANSI Minimal BASIC vorgesehenen Standardfunktionen umfasst UWBIC
ABS, ATN, COS, EXP, INT, LOG, RND, SIN, SQR, TAN.
Nicht verfügbar ist die Funktion SGN und das RANDOMIZE-Statement.
Als sprachliche Besonderheit ist zu erwähnen, dass UWBIC nur die Verwendung der transliterierten Vergleichsoperatoren
GT, GE, LT, LE, NE
zulässt. Als Operationszeichen für die Exponentiation wird der Doppelstern (**) benutzt; Drucksteuerung mit Strichpunkt (;) ist nicht möglich.[4]

213 Compilerstruktur

Das UWBIC-Compilersystem besteht aus acht Programmen, von denen je vier zum Compilierungs- resp. Ausführungssystem gehören. Dementsprechend werden die Programme mit C1 bis C4 (compilation phase) resp. E1 bis E4 (execution phase) bezeichnet.
C1 (ZBASIC) übernimmt im UWBIC-System die Funktion eines Hauptprogramms und dient zum Laden und Aufruf der Compilierungsroutinen.
Der Aufruf von C1 erfolgt normalerweise direkt über Systemsteuerkarten aus der Programmbibliothek. Dieser Aufruf hat für jeden UWBIC-Job, der seinerseits aus mehreren BASIC-Jobs bestehen kann, zu erfolgen. Jeder BASIC-Job stellt ein vollständiges BASIC-Programm dar, das von einer Sternkarte (* in Lochspalte 1) eingeleitet und abgeschlossen wird.
ZBASIC ruft Subroutine C2 zur Initialisierung des UWBIC-Systems auf und leitet anschliessend das Lesen der BASIC-Programmkarte ein, wobei jede Lochkarte (Format 80A1) ein BASIC-Statement beinhaltet. Jedes BASIC-Statement wird separat verarbeitet. Nach dem Einlesen erfolgt die Prüfung auf Existenz einer Statementnummer, die kennwortbezogene Zuordnung zu einem bestimmten Statementtyp, die syntaktische Überprüfung und – falls fehlerfrei – die Codierung in einem UWBIC-eigenen Code.
Beim Auftreten arithmetischer Ausdrücke bzw. numerischer Konstanter werden die Subroutinen C3 und C4 aufgerufen.
Durch ZBASIC wird auch eine fortlaufende Zählung der festgestellten syntaktischen Fehler und eine interne Programmüberwachung vorgenommen. Die Programmüberwachung bezieht sich auf die Korrespondenz der Laufvariablen bei Schleifenverarbeitung mit FOR/NEXT-Statements und die Speicherplatzausnutzung bei numerischen Konstanten.
Syntaktische Fehler werden unmittelbar im Anschluss an das betreffende BASIC-Statement ausgewiesen.[5]
Bei Erreichung eines END-Statements erfolgt Test auf erfolgreiche Compilation und gegebenenfalls – für

4 Vgl. Weber und Türschmann, BASIC, Bd. 1, Tabelle 4 und 15.
5 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 131, Tabelle 2.

sämtliche Variablen, Konstanten und Hilfswerte – die endgültige Speicherplatzzuweisung im Vektor DATA. Wurden syntaktische Fehler festgestellt, so wird deren Gesamtzahl ausgewiesen und die Programmbearbeitung abgebrochen.
Bei fehlerfreier Kompilation wird die Subroutine E1 (ZEXEC) aufgerufen und die Ausführungsphase eingeleitet.
Die Subroutine C2 (ZINITL) führt zur Systeminitialisierung, mit Ladung der Ziffern, Buchstaben und Sonderzeichen in den Commonbereich und Initialisierung der Zählwerke, etc.
Die Subroutine C3 (ZTRANX) dient zur Diagnose und Übersetzung arithmetischer Ausdrücke. Syntaktische Fehler werden durch entsprechende Hinweise kenntlich gemacht; fehlerfreie Ausdrücke werden nach der inversen polnischen Notation umgesetzt und in den Vektor IPROG übergeführt.
Subroutine C4 (ZCONVN) dient zur Umsetzung der in den BASIC-Statements für numerische Konstante enthaltenden BCD-Zeichenfolgen in entsprechende Realzahlen.
Die Subroutine E1 (ZEXEC) simuliert einen Interpreter, der zwanzig allgemeine Operationsbefehle zu verarbeiten vermag. Die Operationscodes werden durch negative Zahlen zwischen -1 und -20 identifiziert.[6]
In Verbindung mit bestimmten OP-Codes werden von ZEXEC auch die Subroutinen E2, E3 und E4 aufgerufen.
Subroutine E2 (ZEVAL) wird zur Verarbeitung arithmetischer Ausdrücke benötigt. Ausdrücke werden durch Folgen von positiven und negativen Zahlen im Vektor IPROG dargestellt. Die positiven Zahlen beziehen sich auf Adressen im Vektor DATA, die negativen Zahlen (-1 bis -19) auf 19 arithmetische Operatoren.[7] Der errechnete Wert des bearbeiteten arithmetischen Ausdrucks wird entweder in ZEVAL zwischengespeichert oder zu einem simulierten Akkumulator (ACC) übertragen. ZEVAL berechnet auch die Adressen indizierter Variabler und speichert den zugehörigen Wertinhalt ab.
Durch die Subroutine E3 (ZINSNO) wird eine Standarddruckzeile, bestehend aus 5 Druckzonen zu je 15 Druckspalten, aufgebaut.
Subroutine E4 (ZHOPPR) dient zur Übernahme von Daten aus der in der Compilierungsphase mit E4 aus den DATA-Statements aufgebauten Datei DATAN oder aus den dem END-Statement folgenden data-Statements. Dabei wird jedes data-Statement vollumfänglich in den Vektor BUFFER übertragen und von dort zur weiteren Verarbeitung bereitgestellt. BUFFER wird sukzessive wieder mit dem Inhalt des nächstfolgenden data-Statements aufgefüllt. Wird die am Programmende stehende Sternkarte erreicht, so erfolgt Programmabschluss via C1 und Reinitialisierung des UWBIC-Systems für den nächstfolgenden BASIC-Job.

6 Sharpe, University of Washington BASIC Interpretive Compiler, p. 24–26.
7 Sharpe, University of Washington BASIC Interpretive Compiler, p. 26–27.

22 FOSBIC (FORTRAN Simulated BASIC Interpretive Compiler)

221 Ursprung

UWBIC wurde 1970 von der Professur für Betriebswirtschaftslehre V der Justus-Liebig-Universität übernommen und seither wesentlich weiter entwickelt. Ergebnis dieser – von Carl Wolfram Türschmann durchgeführten – Entwicklungsarbeiten ist der im Anhang vollumfänglich abgedruckte FOSBIC-Compiler. Die Freigabe dieses – auf UWBIC aufbauenden – BASIC-Compilers erfolgt mit freundlicher Genehmigung von Professor William F. Sharpe, nunmehr Stanford University.[8]

222 Sprachumfang

Das FOSBIC-Compilersystem lässt die Verarbeitung von Elementar-, Matrizen- und Fileanweisungen zu. Die Struktur dieser Anweisungen wurde an anderer Stelle ausführlich dargestellt.[9]
FOSBIC BASIC entspricht mit wenigen Ausnahmen – DEF-Statement, SGN- und RANDOMIZE-Funktion – vollumfänglich den Anforderungen von ANSI Minimal BASIC.

223 Compilerstruktur

Die FOSBIC-Compilerstruktur wird in Kapitel 3 genauer beschrieben. Besonders hervorzuheben ist der modulare Aufbau des Compilers, mit separatem
- Steuerungssystem
- Compilierungssystem und
- Ausführungssystem.

Der modularen Aufbaustruktur entsprechend kann der FOSBIC-Compiler in mehreren Varianten implementiert werden, wobei neben der vollständigen Übernahme mit
- Elementar-, Matrizen- und Fileanweisungen
  auch Teilrealisationen möglich sind. Insbesondere kann das FOSBIC-System partiell für
- Elementar- und Matrizenanweisungen

8 Brief vom 30. Januar 1974: „I would be happy to have you mention my work on the UWBIC system in your books. I have long since left both the University of Washington and development of such programming systems but retain an interest in the sort of work you have done. You have made impressive contributions and produced a system of the most modern type. I congratulate you and your students on these advances."

9 Weber und Türschmann, BASIC, Bd. 1 und 2.

– Elementar- und Fileanweisungen
– Elementaranweisungen
implementiert werden.
Der FOSBIC-Compiler ist infolge seines modularen Aufbaus – bei Verwendung von FORTRAN IV als "host language" – in besonderem Masse als Instruktionsmaterial für Lehrveranstaltungen zum Compilerbau und zur Compilerforschung geeignet.

23 VIEWIT (View it)

231 Ursprung

Der VIEWIT-Compiler wurde am Worcester Polytechnic Institut ausschliesslich zur Verwendung in Lehrveranstaltungen zum Compilerbau entwickelt.[10] Mit VIEWIT liegt der erste BASIC Untersuchungscompiler (take-apart compiler) vor.[11]

232 Sprachumfang

Vom VIEWIT-Compiler können 13 BASIC-Elementaranweisungen
REM, DIM, DEF, DATA, READ, PRINT, LET, GOTO,
IF, FOR/NEXT, END, GOSUB, RETURN
in ihre konstitutiven Elemente zerlegt werden. Entsprechendes gilt für 10 numerische Standardfunktionen
ABS, ATN, COS, EXP, INT, LOG, SGN, SIN, SQR, TAN
und 2 alphanumerische Funktionen
SUB, LEN.
In Abweichung von allen bekannten BASIC-Dialekten werden von VIEWIT nur arithmetische Vergleichsoperatoren vom Typ
.LT. , .LE. , .EQ. , .NE. , .GT. , .GE.

10 Elaine M. Henshon. VIEWIT: A Take-Apart Compiler. M. S. thesis. Worcester 1973. – „The acronym VIEWIT literally means view it." Mitteilung von Professor Norman E. Sondak, Computer Science Departement, Worcester Polytechnic Institute (14. Juni 1976).

11 Henshon, VIEWIT, p. II: „The goal of a take-apart compiler is to allow the student to watch the compiler as it unravels, verifies, diagnoses and translates his input. The language translating process has this become illuminated and understandable. In addition, the student can modify in a simple and direct manner any of the phases to examine the effects of compiler and language desing changes."

und alphanumerische Vergleichsoperatoren vom Typ
.EQ. , .NE.
verarbeitet.
Als weitere Besonderheit von VIEWIT BASIC sei der Stringoperator + + erwähnt;[12] ferner können einfache numerische und einfache alphanumerische Variable auch zwei Buchstaben umfassen, so dass VIEWIT beispielsweise auch AZ, AB1 oder BC als korrekt strukturierte BASIC Variable erkennt. Indizierte Variable können durch einen Buchstaben mit oder ohne nachfolgende Ziffer definiert werden.

233 Compilerstruktur

VIEWIT wurde auf FORTRAN-Basis entwickelt. Als „lexical analyzer" bewirkt VIEWIT eine Zerlegung der zur Verarbeitung gelangenden BASIC-Statements in deren konstitutive Elemente. Diese Statements werden in der Folge durch besondere Prozessoren weiter analysiert, so dass VIEWIT auch als „syntax/semantics analyzer" wirkt.[13]
Die Struktur der einzelnen Unterprogramme wird von Elain Henshon in ihrer Arbeit in Form von Flussdiagrammen dargestellt. Weiterhin dienen zahlreiche in den Text eingestreute Beispiele zur Verdeutlichung der mit VIEWIT erreichbaren BASIC-Statementanalyse.
Seiner besonderen Struktur entsprechend ist der VIEWIT-Compiler nur für Informatiker von Interesse.

12 Henshon, p. 9: „The concatenation operation has the form: Operand + + operand. The string produced by this operation is equal to the first string adjacent to the second string. The length of the string result is the sum of the lenghts of the two string operands."

13 Vgl. auch Franklin L. DeRemer. Lexical Analysis. In: F. L. Bauer and J. Eickel. Compiler Construction. An Advanced Course. Berlin 1974, p. 109–120.

3 FOSBIC-Compilersystem

Die vom Dartmouth College ausgehende BASIC-Entwicklung bezog sich zunächst nur auf Dialogsystems[1]. Um BASIC auch im Batchbetrieb einsetzbar zu machen, wurde von William F. Sharpe ein BASIC-Compiler auf FORTRAN-Basis konzipiert (UWBIC) und durch Veröffentlichung allgemein zugänglich gemacht[2].

31 FOSBIC-Entwicklung

UWBIC BASIC wurde 1970 auf der Anlage CD 3300 des Hochschul-Rechenzentrums der Justus-Liebig-Universität Giessen implementiert[3]. Davon ausgehend wurde in der Folge – über sechs miteinander kompatible Versionen – das im Anhang veröffentlichte FOSBIC-Compilersystem entwickelt.
Den Arbeiten am FOSBIC-Compilersystem lagen bisher folgende Ziele zugrunde[4]:
- Compilerkompatibilität
- Austauschfähigkeit von BASIC-Programmen
- Benutzerfreundlichkeit von BASIC
- Weiterentwicklung und Standardisierung von BASIC
- Realisation eines modular aufgebauten BASIC-Compiler-Compilers.

Zukünftige FOSBIC-Weiterentwicklungen sollen insbesondere zum Aufbau einer Modellbank[5] und eines selbständig lehrenden BASIC-Systems führen[6].

1 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 111.
2 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 131.
3 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 132.2.
4 Vgl. auch: James J. Horning. Structuring Compiler Development. In: F. L. Bauer und J. Eickel (ed.). Compiler Construction, An Advanced Course. Berlin 1974, p. 498–504.
5 C. Wolfram Türschmann. OR-Modellbank in BASIC. In: J. Kohlas u. a. (Hrsg.). Proceedings in Operations Research 5. Würzburg 1976, S. 504–505. – Vgl. Abs. 352.
6 Vgl. Abs. 353.

311 Compilerkompatibilität

Die Kompatibilität[7] des FOSBIC-Compilersystems wird gewährleistet durch
- Verwendung von USASI FORTRAN IV als Basissprache und
- Einstellbarkeit auf maschinenorientierte Systemkonstante.

311.1 Basissprache FORTRAN IV

Der FOSBIC-Compiler enthält – mit wenigen Ausnahmen – nur Anweisungen in USASI FORTRAN IV. Die Ausnahmen sind bedingt durch Verwendung des FORTRAN IV-Compilers der CD 3300 und betreffen
- Ablauftechnik des FOSBIC-Compilersystems (OVERLAY-Struktur)
- Abfrageanweisungen auf ein Fileende
- Fileanweisungen in FORTRAN IV und
- Zeichencode des FOSBIC-Programms.

Durch die in den Programmlisten angegebenen Hinweise können die bei der FOSBIC-Implementation auf anderen Anlagen erforderliche Anpassungen an den verwendeten FORTRAN-Compiler leicht realisiert werden.

311.2 Einstellbarkeit auf Systemkonstante

Computersysteme verschiedener Hersteller, unterschiedlicher Entwicklungs- oder Ausbaustufen werden durch spezifische Systemkonstante charakterisiert. Besonders bedeutungsvoll sind die sich auf Zentraleinheit und Peripherie beziehenden Kenngrössen. Bezüglich der Zentraleinheit handelt es sich um
- Kernspeichergrösse
- Wortgrösse
- Wortstruktur
- Zeichencode

und bezüglich der Peripherie um
- Kanalnummer des Lochkartenlesers
- Kanalnummer des Schnelldruckers

7 Unter Compilerkompatibilität wird hier die „Portability and Adaptability" eines Compilers verstanden; vergl. hierzu: Peter C. Poole. Portable and Adaptable Compilers. In: Bauer and Eickel, Compiler Construction, p. 427 und p. 429: „...we can say that a [portable and adaptable compiler] is one that can easily be moved to a new machine and modified to interface with its operating system".

– Druckbreite des Schnelldruckers
– Anzahl und Art der anlegbaren Files.

Der FOSBIC-Compiler kann bei seiner Implementation mittels eines speziellen Unterprogramms ZINITL auf alle wichtigen Systemkonstanten eingestellt werden.[8]

312 Austauschfähigkeit von BASIC-Programmen

Unter Austauschfähigkeit wird generell die Möglichkeit verstanden, Programme von einem System auf ein anderes weitgehend unverändert zu übertragen.[9]

In BASIC-Programmen können bei Verwendung der nicht standardisierten Sonderanweisungen[10] und des INPUT-Statements Austauschschwierigkeiten auftreten, letztere insbesondere beim Übergang vom Timesharing- zum Batchbetrieb.

INPUT-Statements werden vom FOSBIC-Compiler auch im Monologbetrieb verarbeitet, so dass diesbezüglich keine Änderungen des BASIC-Programms erforderlich sind.

Die Austauschfähigkeit von BASIC-Programmen wird vom FOSBIC-Compiler auch durch Verarbeitung syntaktisch unterschiedlich aufgebauter BASIC-Statements gleicher semantischer Bedeutung in hohem Masse gewährleistet. Verwiesen sei insbesondere auf die

– Elementaranweisungen: LET, let, IF, bedingtes GOTO, bedingtes GOSUB[11] und die
– Matrizenanweisungen.[12]

Weiterhin wird durch alternative Angabe der Vergleichsoperatoren als Grundsymbole oder als transliterierte Zeichenfolge[13] die Programmübernahme erleichtert. Ebenso kann als Exponentiationszeichen alternativ der Doppelstern (**) oder ein spezielles Grundsymbol (^) verwendet werden.

Bezüglich der BASIC-Fileanweisungen konnte – infolge fehlender Standardisierung und grosser Anweisungsvielfalt[14] – eine allgemeine Transferabilität nicht verwirklicht werden. Vermerkt sei aber, dass sich aus der Grundstruktur des FOSBIC-Compilers die Möglichkeit ergibt, die Zahl der derzeit zulässigen Filekennworte beliebig auszuweiten. Begrenzt werden diese Ausbaumöglichkeiten nur durch den Program-

8 Vgl. Abs. 332.22
9 Vgl. auch: Poole, Portable and Adaptable Compilers, p. 437–438.
10 Vgl. Gerald L. Isaacs. Interdialect Translatability of the BASIC Programming Language. Iowa City 1972, p. 15–17.
11 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 332.3, 332.41, 332.412 und 332.462.
12 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 331.
13 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 321.1 und Tabelle 4.
14 Vgl. auch Isaacs, Interdialect Translatability of the BASIC Programming Language, p. 14: „The least translatable of all the statements are the file handling statements. Different dialects have different methods for handling files. In some dialects the user allocates a file name with a FILE statement, a FILES statement or an ASSIGN statement depending upon the dialect. Other dialects implicitly do this in the OPEN or first access. Backspacing and rewinding of files are allowed in a few dialects. Some dialects read from files with an INPUT statement while others use a READ

mierumfang und den verfügbaren Kernspeicherplatz. Die Schnittstellen, an denen entsprechende Programmiereingriffe erfolgen können, sind für das ganze FOSBIC-System eindeutig definiert.

313 Benutzerfreundlichkeit von BASIC

Der lernorientierte Charakter von BASIC kommt in der Einfachheit der Sprachstruktur deutlich zum Ausdruck. Dementsprechend wird auch benutzerfreundlichen Fehlermeldungen grosse Bedeutung beigemessen.[15]
Der FOSBIC-Compiler erkennt
– syntaktische Fehler, die während der Compilierung auftreten und
– semantische Fehler, die zu einem Programmabbruch im Object-Time-Program führen
und erläutert sie im Vergleich zu anderen BASIC-Compilern besonders klar und ausführlich.
Ein Ausbau der Fehlerdiagnose ist sowohl für die Compilierungs- als auch für die Ausführungsphase des FOSBIC-Compilersystems praktisch unbegrenzt und in einfacher Weise möglich. Derartige Ausweitungen sind aber so zu gestalten, dass das im FOSBIC-Compilersystem konsequent realisierte Prinzip der internen Fehlererkennung und Fehlerverarbeitung beibehalten bleibt, so dass ein compilerexterner Programmabbruch nur bei Maschinenfehlern auftreten kann.
Das vorerwähnte Postulat wurde beim UWBIC-Compiler noch nicht vollständig eingehalten. Beispielsweise wurde eine Division durch Null nicht abgefangen, so dass es zu einem Programmabbruch ausserhalb des Compiler-Compilers durch das Maschinensystem kam, wodurch alle eventuell nachfolgende BASIC-Jobs innerhalb eines Gesamtjobs nicht mehr bearbeitet wurden.

314 Weiterentwicklung und Standardisierung von BASIC

Bei der Darstellung der Fileanweisungen[16] ergaben sich erhebliche Probleme, Gemeinsamkeiten bei den einzelnen BASIC-Compilern zu finden. Die nun vorliegenden FOSBIC-spezifischen Fileanweisungen können als Anregung einer entsprechenden BASIC-Standardisierung gelten.
Das FOSBIC-Compilersystem kann auf Grund seiner Struktur dem internationalen BASIC-Standard fort-

statement. Also, PRINT and WRITE statements are used for writing into files in different dialects. Some dialects sense for end of file with an IF END statement; others use a NODATA statement, while others use an ENDFILE statement. File namens are determined from dialect to dialect and even from installation to installation within a dialect. Therefore, file handling is not directly translatable."

15 Vgl. auch: James J. Horning. What the Compiler Should Tell the User. In: Bauer and Eickel, Compiler Construction, p. 525–548.

16 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 332.

laufend angepasst werden und durch Neuentwicklung von BASIC-Anweisungen oder BASIC-Modulen zum Aufbau dieser Sprache beitragen[17]. Dies trifft vor allem zu für die Integration
- weiterer eingebauter Funktionen (intrinsic funktion)[18],
- zusätzlicher MAT-Funktionen[19] und
- verarbeitungsschnellerer Fileanweisungen[20].

315 Realisation eines modularen Compiler-Compiler Aufbaus

Ein Compiler-Compiler[21] kann als ein in einer Basissprache geschriebenes Programm, das wiederum einen Compiler für eine Programmiersprache erzeugt, definiert werden.
Bei den BASIC-Compilern FOSBIC, UWBIC und VIEWIT[22] wurde FORTRAN als Basissprache verwendet. Alle drei Compiler-Compiler sind modular aufgebaut[23], wobei im UWBIC-Compiler die Modul-

17 Vgl. Abs. 352.
18 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 325.
19 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 331.221.
20 Vgl. Abs. 351.22.
21 Über Compiler-Compiler vgl.: F. R. A. Hopgood. Compiler: Die Übersetzung von Programmiersprachen. München 1970, S. 149–152; W. D. Maurer. Generalized Interpretation and Compilation. In: Julius T. Tou (ed.). Software Engineering, Vol. 1. New York 1970, p. 139–152; David Gries. Compiler Construction for Digital Computers. New York 1971, p. 435–439; R. Brooker, I. MacCallum, D. Morris and J. S. Rohl. The Compiler-Compiler. Annual Review in Automatic Programming 3 (1963), p. 229; John A. Lee. The Anatomy of a Compiler. New York 1974, p. 38–41; R. A. Brooker, D. Morris and J. S. Rohl. Experience with the Compiler-Compiler. Computer Journal 9 (1967) 4, p. 345; J. S. Rohl. An Introduction to Compiler Writing. London 1975, p. 281–288; Hans J. Schneider. Compiler: Aufbau und Arbeitsweise. Berlin 1975, S. 25; John E. Nicholls. The Structure and Design of Programming Languages. London 1975, p. 50–52; Barry W. Pollack. Compiler Techniques. New York 1972 (insbesondere die Beiträge von Jerome A. Feldman und Saul Rosen); Friedrich L. Bauer and J. Eickel (ed.). Compiler Construction: An Advanced Course. Berlin 1974 (insbesondere die Beiträge von W. M. McKeeman, M. Griffiths, James J. Horning und Peter C. Poole). – Eine umfangreiche Bibliographie enthält: Peter Naur. Concise Survey of Computer Methods. New York 1974, p. 344–345.
22 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 132.1.
23 Zum modularen Programmaufbau vgl.: Sherman C. Blumenthal. Management Information System: A Framework for Planning and Development. Englewood Cliffs 1969; John G. Burch and Felix R. Strater. Information Systems: Theory and Practice. Santa Barbara 1974, p. 264–265 und 285; Rüdiger und Anna Gritsch. Das Programmieren von Computern. Ein Lehr- und Lernbuch unter Verwendung von FORTRAN. München 1972, S. 281–286; W. M. McKeeman. Compiler Construction. In: Bauer and Eickel, Compiler Construction, p. 15; Gerhard Goos. Systemprogrammiersprachen und strukturiertes Programmieren. In: Clemens E. Hackl (ed.). Programming Methodology. Berlin 1975, p. 203–224; Glenford J. Myers. Reliable Software Through Composite Design. New York 1975, p. 8, 14–15, 19–54 und 110–112.

struktur nur für arithmetische Ausdrücke durch das Unterprogramm ZTRANX verwirklicht wurde. Im Gegensatz dazu ist beim FOSBIC-Compiler die Modulstruktur auf die gesamte Programmiersprache BASIC – Elementar-, Matrizen- und Fileanweisungen umfassend – ausgedehnt.
Unter der Anwendung des Schemas von W. D. Maurer[24] stellt sich das Übersetzungsschema des FOSBIC-Compilers in drei Stufen dar.

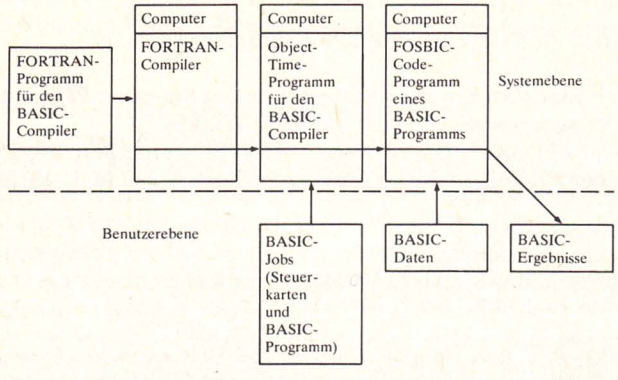


Abbildung 4. Übersetzungsschema des FOSBIC-Compilersystems

32 FOSBIC-Struktur

Voraussetzung zur Implementation[25], Variation[26] und Extension[27] des modularen FOSBIC-Compilersystems ist die Kenntnis der
- Aufbaustruktur und
- Ablaufstruktur.

24 Maurer, Generalized Interpretation and Compilation, in: Julius T. Tou (ed.), Software Engineering, Vol. 1, p. 147.
25 Vgl. Abs. 33
26 Vgl. Abs. 34
27 Vgl. Abs. 35

321 Gesamtstruktur

321.1 Aufbaustruktur

Das FOSBIC-Compilersystem kann in
- Gesamtjob-Struktur oder
- OVERLAY-Struktur[28]

aufgebaut werden, wobei sich das Gesamtsystem in drei Subsysteme
- Steuersystem,
- Compilierungssystem und
- Ausführungssystem

aufteilen kann.
Bei Gesamtjob-Struktur stehen Compilierungs- und Ausführungssystem ständig im Kernspeicher. Ein spezielles Steuerungssystem entfällt.
Bei OVERLAY-Struktur steht das Steuersystem mit dem Steuerprogramm OVERT ständig im Kernspeicher, während Compilierungs- oder Ausführungssystem nach Bedarf von einer Magnetplatte hinzugeladen werden. Der Vorteil dieser Technik besteht in einer erheblichen Einsparung an Kernspeicherplatz – ca. 10 K Worte (1 Wort = 48 Bits) bei der CD 3300 – gegenüber der Gesamtjob-Struktur. Nachteilig ist eine gewisse Systeminkompatibilität, die allerdings leicht behoben werden kann.
Das Gesamtsystem besteht aus 27 Programmen und Unterprogrammen, wovon 6 Programme auf das Steuersystem, 9 Programme auf das Compilierungssystem und 12 auf das Ausführungssystem entfallen. Fünf Programme des Steuersystems werden sowohl bei der Compilierung als auch bei der Ausführung des FOSBIC-Code Programms benötigt.

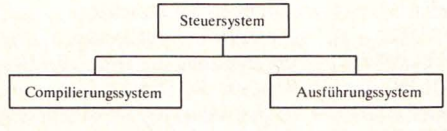


Abbildung 5. Aufbau des FOSBIC-Compilersystems in OVERLAY-Struktur

Tabelle 1 zeigt eine Übersicht über die einzelnen Programme und deren Aufgabengebiet.

28 Zur OVERLAY-Technik der CD 3300 siehe: W. Lucklum, G. Teichmann und D. Weiß. FORTRAN-Programmierung für die EDV-Anlage CD 3300. Giessen (Rechenzentrum der Justus-Liebig-Universität Giessen) 1974, S. 73–75.

Tabelle 1. Programme des FOSBIC-Compilersystems

Programmart	Name	Aufgabengebiet
Hauptprogramm, intermediäre Unterprogramme	OVERT	Systemsteuerung
	ZHOPPR	Datenaufnahme und Verwaltung
	ZDIGIT	Ziffernerkennung
	STRING	Textverarbeitung
	ZALPH	Zeichenerkennung
	FINDFI	Dateinamenerkennung
Programm und Unterprogramme zur Compilierung der BASIC-Statements	MAIN	Compilierung der Elementaranweisungen (Steuerung des Systems bei Gesamtjobstruktur)
	COMERR	Compilierungsdiagnostik
	ZINITL	Systeminitialisierung und Verarbeitung der BASIC-Steuerkarten
	ZTRANX	Verarbeitung arithmetischer Ausdrücke
	ZCONVN	Aufnahme numerischer Konstanten
	MATTRA	Compilierung der Matrizenanweisungen
	ZKLAM	Redimensionierung
	ZLISTE	Compilierung von Ein- und Ausgabelisten
	ZFILE	Compilierung der Fileanweisungen
Programm und Unterprogramme zur Verarbeitung des FOSBIC-Code-Programms	ZEXEC	Verarbeitung der Elementaranweisung
	ZEVAL	Verarbeitung arithmetischer Ausdrücke
	ZINSNO	Standarddruckausgabe
	ZEXFIL	Verarbeitung von File-Codes
	ZEXMAT	Verarbeitung von MAT-Codes
	EXERR	Object-Time-Diagnostic
	CLEAR	Löschung der Druckzeile
	ZNUMB	Ganzzahl in Druckposition bringen
	ZIMAGE	Verarbeitung der IMAGE-Maske
	PRILIN	Ausgabe der Druckzeile
	CHECK	Finden einer Standardeinteilung der Druckzeile
	SUBINV	Inversion einer Matrize

Die für das FOSBIC-Compilersystem bedeutsamen Variablen und Parameter sind für alle Programme, mit Ausnahme des Programms SUBINV, in identischen COMMON-Blöcken zusammengefasst. Tabelle 2 erläutert kurz, in der Reihenfolge ihres Auftretens in den COMMON-Blöcken, die Bedeutung und Verwendung der einzelnen Grössen.[29]

29 Vgl. auch FOSBIC-Programmliste im Anhang

Tabelle 2. Bedeutung der Variablen in den COMMON-Blöcken

FORTRAN-Name	Bedeutung und Verwendung
ACC	Systemakkumulator
INREG	Zählregister des Programmablaufs
LNGCRP	Ende eines BASIC-Statements
NERRS	Zählregister der ERROR-Meldungen
NEXTDT	Zählregister der Daten aus den DATA-Statements
NIFOR	Zählregister der ineinandergeschachtelten FOR/NEXT-Schleifen
NIRET	Zählregister der ineinandergeschachtelten Unterprogramme
NSTLST	Zählregister für die numerierten Statements
INEXT	Oberste Adresse des benötigten Adressbereichs
NUMBUF	Zählregister für Daten innerhalb eines DATA-Statements
IBEGST	Anfang eines BASIC-Statements
IWRIT	Breite der Standarddruckzeile IWRIT = IZONE * IPEND + 1
NPRUS	Zählregister der Ausgabezeile für PRINT USING-Statements
NCARD	Steuerparameter des PRINT USING-Statements
NPRI	Steuerparameter zur gepackten und ungepackten Ausgabe in der Standarddruckzeile
NUMFIL	Dateizählregister
NZIM	IMAGE-Statementzählregister
NSTZEI	Steuerparameter zur Eingabe der BASIC-Grundsymbole
IMIRC	Merkspeicher für den Standardeingabekanal
INTZEI	Anzahl der codierbaren BASIC-Grundsymbole
CARDT	Compilierungsspeicher für MAT-Statements
MERKER	Feldgrenzenspeicher
CARP	Bufferspeicher für die Ausgabe
ALPH	Speicher der codierbaren BASIC-Grundsymbole
BUFFER	Speicher zur Datenaufnahme aus einem DATA- oder data-Statement
CARD	Compilierungsspeicher zur Aufnahme von BASIC-Statements
CARDP	Compilierungsspeicher mit verdichtetem BASIC-Statement
DIGIT	Ziffernspeicher
IFOR	Adressspeicher für die Laufvariablen von FOR/NEXT-Schleifen
IRET	Adressspeicher zur Unterprogrammtechnik
XXX	Hilfsspeicher für Vergleiche und zur Redimensionierung von Feldgrenzen
NFILE	Speicher der Dateinamen, Satzanzahl und des Anfang-DSI's.
ISTLST	Speicher der internen Programmadressen
LISTST	Speicher der Statementnummern
DATAN	Speicher der Daten aus den DATA-Statements
DATA	Adress-, Konstanten- und Felderspeicher
IPROG	FOSBIC-Code Programmspeicher

Die Bedeutung der nicht aufgeführten FORTRAN-Namen kann den Tabellen 7 und 8 Abs. 332.21 und Tabelle 9 Abs. 332.22 entnommen werden.

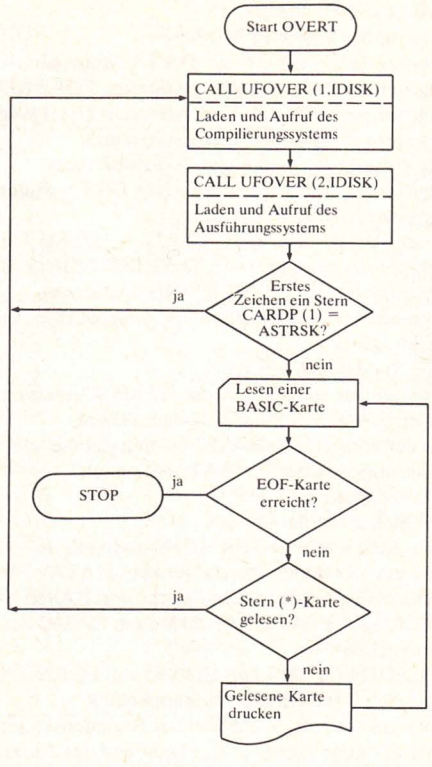


Abbildung 6. Ablaufstruktur des FOSBIC-Compilersystems in OVERLAY-Struktur

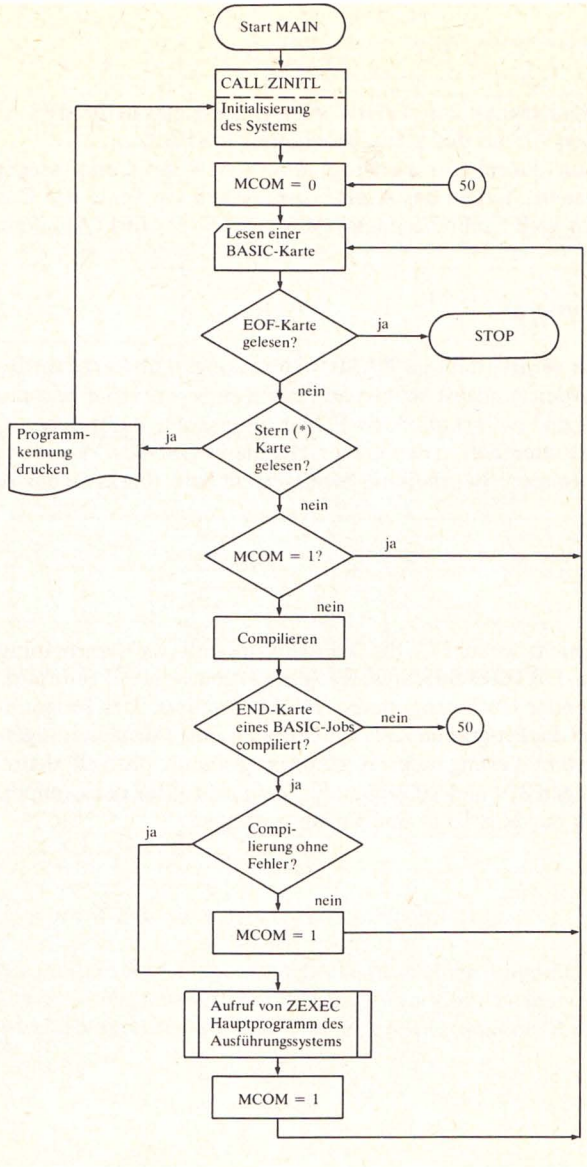


Abbildung 7. Ablaufstruktur des FOSBIC-Compilersystems in Gesamtjobstruktur

321.2 Ablaufstruktur

Abbildung 6 zeigt die Ablaufstruktur des FOSBIC-Compilersystems in OVERLAY-Struktur. Gleichzeitig beinhaltet es den logischen Ablauf des Steuerprogramms OVERT.
In der Gesamtjob-Struktur übernimmt das Programm MAIN des Compilierungssystems zusätzlich die Steuerung des Gesamtsystems, indem das Ausführungssystem am Ende der Compilierung eines fehlerfreien BASIC-Programms vom Compilierungssystem aufgerufen wird (Abbildung 7).

322 Compilierungssystem

Das Compilierungssystem verarbeitet alle BASIC-Anweisungen und erkennt BASIC-Steuerkarten. Die BASIC-Anweisungen werden zunächst auf Grund ihrer Kennworte einer bestimmten Anweisungsgruppe zugeordnet und anschliessend auf syntaktische Fehler untersucht. Auftretende Fehler werden analysiert und gemeldet, fehlerfreie Statements in den entsprechenden FOSBIC-Code umgewandelt und fehlerfreie BASIC-Programme zur weiteren Bearbeitung bereitgestellt bzw. direkt an das Ausführungssystem übergeben.

322.1 Aufbaustruktur

Die Aufbaustruktur entspricht der für FOSBIC typischen modularen Verarbeitung von Elementar-, Matrizen- und Fileanweisungen. Die Elementaranweisungen werden von dem zentralen Programm MAIN unter Zuhilfenahme entsprechender Unterprogramme selbst compiliert. Das Programm MATTRA bearbeitet Matrizenanweisungen und das Programm ZFILE übernimmt alle Fileanweisungen. MATTRA und ZFILE benötigen zur Statementcompilierung mehrere Unterprogramme, die Teilinhalte der BASIC-Statements bearbeiten. Die Abbildungen 8, 9 und 10 zeigen die Aufbaustruktur des Compilierungssystems sowie der Module zur Verarbeitung der Matrizen- und Fileanweisungen.

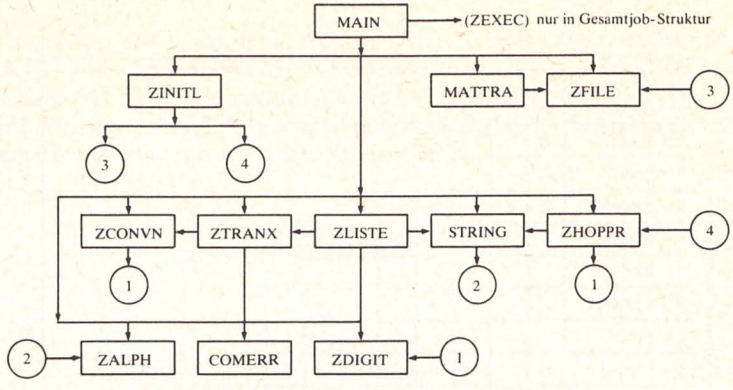


Abbildung 8. Aufbaustruktur des Compilierungssystems

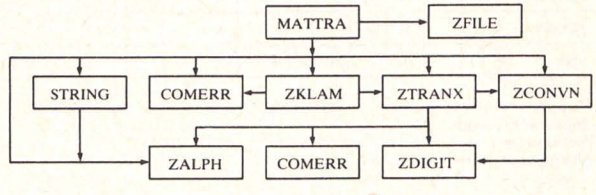


Abbildung 9. Aufbaustruktur des Moduls zur Verarbeitung von Matrizenanweisungen

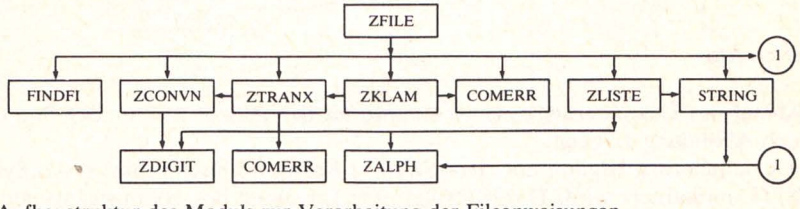


Abbildung 10. Aufbaustruktur des Moduls zur Verarbeitung der Fileanweisungen

Die Zusammenfassung der Module des Compilierungssystems in einer Programmatrix zeigt Abbildung 11. Beim Entfernen eines Moduls – beispielsweise der Fileanweisung – ist zu beachten, von welchen Programmen das ausscheidende Modul aufgerufen wird, da dort die entsprechenden Modulaufrufe unwirksam gemacht werden müssen.[30]

30 Vgl. Abs. 332.1

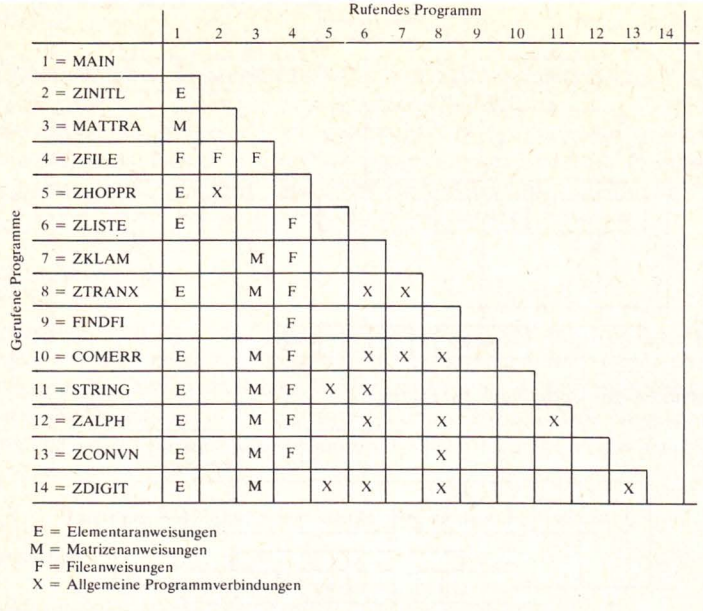


Abbildung 11. Programmtrix des FOSBIC-Compilierungssystems

322.2 Ablaufstruktur

Der logische Ablauf des Compilierungssystems stimmt mit der Ablaufstruktur des Programms MAIN überein, wie auch Abbildung 12 zeigt.
Der Ablauf der Compilierung beginnt mit einer Stern(*)-Karte, wodurch zunächst das System mit dem Programm ZINITL initialisiert wird. Dabei erfolgt eine Löschung des vorangegangenen BASIC-Programms, eine entsprechende Vorbereitung zur Übernahme des neuen BASIC-Programms und eine Einstellung der BASIC-Hardwareparameter entsprechend den Angaben auf der BASIC-Steuerkarte. Nach dieser Systeminitialisierung erfolgt die zeilenweise Aufnahme des neuen BASIC-Programms, dessen Codierung in den entsprechenden FOSBIC-Code und bei einer fehlerfreien Compilierung eine Überarbeitung des FOSBIC-Code Programms. Bei Gesamtjob-Struktur wird das Compilierungssystem beendet durch die Übergabe des Systemablaufs an das Ausführungssystem unter der Leitung des Programms ZEXEC; bei OVERLAY-Struktur erfolgt die Rückgabe der Systemkontrolle an das Programm OVERT.

Alle Matrizenanweisungen werden von dem Programm MAIN an ihrem Kennwort MAT erkannt und an das Programm MATTRA übergeben. Bei MAT-Fileanweisungen übergibt MATTRA die endgültige Bearbeitung an ZFILE, wobei für die Fileanweisungen auf Grund des Kennwortes bereits eine Codezuordnung getroffen wird. Dies gilt auch für die übrigen Fileanweisungen, deren Kennwort zunächst von MAIN analysiert wird, bevor die Codierung durch ZFILE erfolgt.

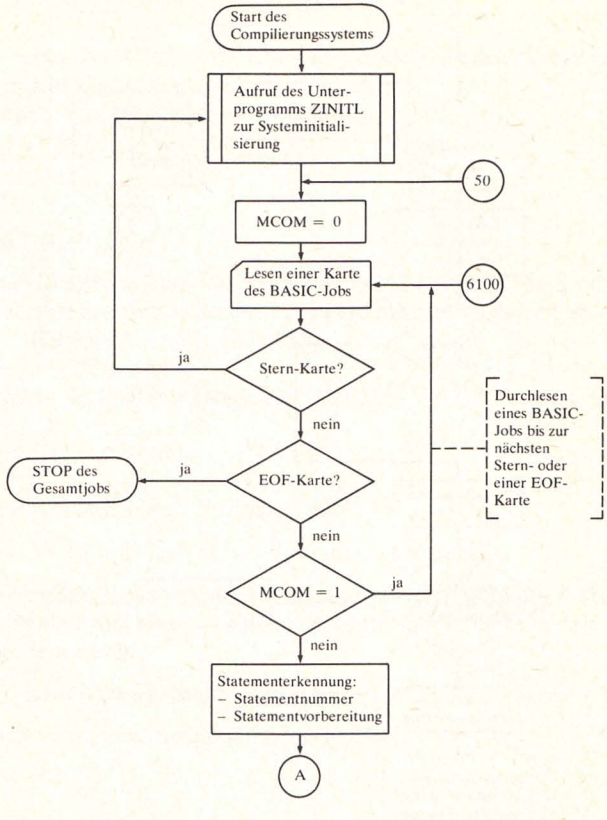
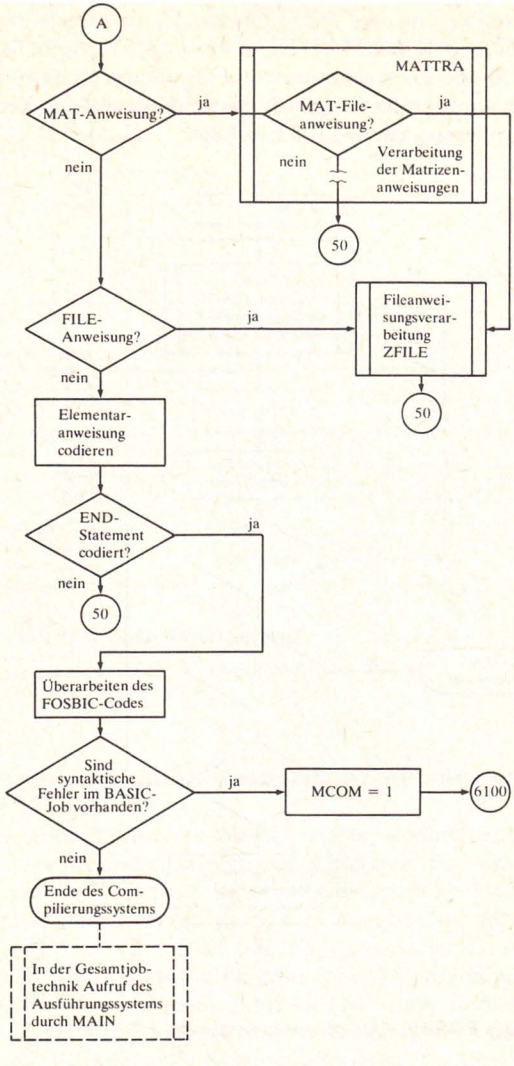


Abbildung 12. Ablaufstruktur des FOSBIC-Compilierungssystems



Fortsetzung Abbildung 12.

322.3 Adressierung

Unter Adressierung wird die vom Compilierungssystem vorgenommene Zuordnung von voradressierten Speicherplätzen eines vorgegebenen Speicherbereichs an mnemotechnisch benannte Variable und Konstante eines Programms verstanden.

322.31 Adressierung einfacher Variabler

Die Speicherplätze aller in einem BASIC-Programm möglichen einfachen Variablen liegen implizit im unteren Teil des Adressbereichs des FOSBIC-Code Programms[31].
Die Adressen der einfachen Variablen nach der Definition[32]

⟨Einfache Variable⟩ := ⟨Buchstabe⟩ ⟨Ziffer⟩

werden nach der Formel

Adresse = K + 26 * (L − 1) + 53

berechnet, wobei K die Codezahl des Buchstaben im Vektor ALPH und L die Codezahl im Vektor DIGIT darstellen[33]. Einfache Variable des vorerwähnten Typs werden somit im Adressbereich 54 bis 313 des Vektors DATA gespeichert. Einfache Variable vom Typ

⟨Einfache Variable⟩ := ⟨Buchstabe⟩

erhalten ihren Speicherplatz als Feld der Dimension (0,0).

322.32 Adressierung indizierter Variabler

Für die Dimensionierung von indizierten Variablen gilt[34]:

⟨Dimensionierung⟩ := ⟨Spaltenzahl⟩| ⟨Zeilenzahl⟩, ⟨Spaltenzahl⟩

Der Feldanfang einer indizierten Variablen wird in den Adressen 1 bis 26 des Adressbereichs des Vektors DATA gespeichert. Die Spaltenzahl eines zweidimensionalen Feldes steht in den Adressen 28 bis 53 von DATA; das Feldende in dem durch

DATA (DATA(Codezahl des Feldnamens in ALPH))

festgelegten ersten Speicherplatz des entsprechenden Feldes.

31 Vgl. Abs. 323.1
32 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 323.11.
33 Vgl. Tabelle 7 und 8.
34 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 331.211.

Die Speicherplatzreservierung erfolgt in mehreren Schritten.
Zunächst wird auf Grund eines DIM-Statements die Elementzahl des Feldes nach folgenden Beziehungen berechnet[35]:

- für eindimensionale Felder
  ILNG = SIZE + 2
- für zweidimensionale Felder
  ILNG = (ROWS + 1) * (COLS + 1) + 1.

Die Werte von SIZE = Zeilenzahl, ROWS = Zeilenzahl und COLS = Spaltenzahl werden mittels des Programms ZCONVN aus dem DIM-Statement bestimmt. Der Wert von ILNG wird im Adressteil in den Adressen von 1 bis 26 vermerkt, wobei die Codezahl des mnemotechnischen Feldnamens bestimmt, welche Adresse verwendet wird. Die Spaltenzahl eines zweidimensionalen Feldes wird in DATA(I+27) gespeichert.
Nach fehlerfreier Compilierung werden vom Compilierungssystem auf Grund der festgehaltenen Feldgrenzen die aktuellen Feldgrenzen der explizit oder implizit[36] dimensionierten Variablen für das FOSBIC-Code Programm errechnet, wobei die vorerwähnten einfachen Variablen der Dimension (0,0) mit berücksichtigt werden. Hierbei wird, ausgehend von der obersten Adresse INEXT des Adressbereichs, der zum Speichern von numerischen Konstanten und Hilfsspeichern belegt wurde, in den Adressen 1 bis 26 die Feldanfangsadresse jedes Feldes gespeichert. Zu dieser Feldanfangsadresse wird die berechnete Feldgrösse hinzuaddiert und der sich ergebende Wert im ersten Speicherplatz des entsprechenden Feldes vermerkt und als Feldanfangsadresse für die nächste indizierte Variable verwendet.
Beispielsweise sei nach erfolgreicher Compilierung eines BASIC-Programms der Adressbereich von DATA durch numerische Konstanten und Hilfsspeicher bis zur Adresse 400 aufgefüllt, wobei als einziges DIM-Statement

```
10 DIM A(10,10),C(5),D(4,3)
```

aufgetreten war. Im Adressbereich von DATA werden dann folgende aktuellen Feldgrenzen festgelegt:

DATA (1)	= 400	
DATA (400)	= 522	= 400 + (10 + 1) (10 + 1) + 1, für A (10, 10)
DATA (2)	= 522	
DATA (522)	= 524	= 522 + (0 + 1) (0 + 1) + 1, für B (0, 0)
DATA (523)	enthält den Wert der einfachen Variablen B, da B nicht dimensioniert wurde	
DATA (3)	= 524	
DATA (524)	= 531	= 524 + 5 + 2
DATA (4)	= 531	

35 Vgl. FOSBIC-Programmliste, Programm MAIN, Statement 893 bis 964.
36 Vgl. zur impliziten Dimensionierung: Weber und Türschmann, BASIC, Bd. 1, Abs. 331.211.

DATA (531) = 552 = 531 + (4 + 1) (3 + 1) + 1
DATA (5) = 552
DATA (552) = 554 = 552 + (0 + 1) (0 + 1) + 1, da E nicht dimensioniert wurde
⋮

In den Speicherplätzen für die Spaltenzahl der Felder DATA (I + 27) wird folgendes vermerkt:
DATA (28) = 11 = 10 + 1
DATA (29) = 11 implizit durch ZINITL festgelegt
DATA (30) = 11 implizit durch ZINITL festgelegt
DATA (31) = 4 = 3 + 1
DATA (32) = 11 implizit durch ZINITL festgelegt
⋮

Die Adresse eines beliebigen Feldelementes errechnet sich nach folgenden Formeln:
– eindimensionales Feld

Adresse = DATA(I) + Zeilenindex + 1

– zweidimensionales Feld

Adresse = DATA(I) + DATA(I+27)*Zeilenindex + Spaltenindex + 1

wobei I die Codenummer des Feldnamens darstellt.
Auf vorausgehendes Beispiel bezogen steht der Wert des Elementes $D_{3,2}$ in

Adresse (D(3,2)) = DATA (4) + DATA (4 + 27) 3 + 2 + 1
531 + 4 * 3 + 2 + 1 = 546.

I hat hier den Wert 4, da D der vierte Wert in der Codetabelle für Variablen- bzw. Feldnamen ist.[37]
Mit dem PRINT ALL-Statement[38] kann sich der Benutzer alle Werte eines BASIC-Programms grösser als SMALL und deren Feldadressen im Vektor DATA ausgeben lassen. SMALL ist ein im Programm ZINITL festlegbarer Steuerparameter[39].
Alle durch DIM-Statements oder implizite Dimensionierung ursprünglich festgelegte Feldgrenzen werden in der Matrix MERKER festgehalten. In MERKER(I,1) steht die Zeilenzahl und in MERKER(I,2) die Spaltenzahl. Für einen Zeilenvektor wird die Elementzahl in MERKER(I,1) vermerkt und MERKER(I,2) auf Null gesetzt. I stellt wiederum die Codenummer des Feldnamens dar.

37 Vgl. Tabelle 7 und 8.
38 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 333.12.
39 Vgl. Abs. 332.22 und Tabelle 9.

Das folgende BASIC-Programm zeigt die verschiedenen Möglichkeiten zur Festlegung und Redimensionierung von Feldgrenzen. Während durch das DIM-begrenzte MAT READ-Statement die Feldgrenzen nur aktuell auf die im Statement angegebenen Feldgrenzenwerte redimensioniert werden, bleibt die Redimensionierung für das Feld B nach dem Statement 70 erhalten.

```
                    10 DIM A(10,1),B(1,20),F(4,3)      ! EXPLIZITE DIMENSIONIERUNG
                    20 MAT LET G=ZER(3,2)              ! EXPLIZITE DIMENSIONIERUNG
                    30 LET R(2,3)=4                    ! IMPLIZITE DIMENSIONIERUNG
                    40 MAT READ A(5,1)                 ! DIM-BEGRENZTES MAT READ-STATEMENT
                    50 DATA 1,2,3,4,5
                    60 PRINT ALL
                    70 MAT LET B=CON(3,2)              ! REDIMENSIONIERUNG VON FELDGRENZEN
                    80 PRINT ALL
                    90 END

LOCS=   324  LOCU=   347  ROWS=    11  COLUMNS=    2.  VARIABLE A
                     A(    3) OR A(   1,    1) =        1          DATA(  328)
                     A(    5) OR A(   2,    1) =        2          DATA(  330)
                     A(    7) OR A(   3,    1) =        3          DATA(  332)
                     A(    9) OR A(   4,    1) =        4          DATA(  334)
                     A(   11) OR A(   5,    1) =        5          DATA(  336)
LOCS=   347  LOCU=   390  ROWS=     2  COLUMNS=   21.  VARIABLE B
LOCS=   390  LOCU=   392  ROWS=     0  COLUMNS=   11.  VARIABLE C
LOCS=   392  LOCU=   394  ROWS=     0  COLUMNS=   11.  VARIABLE D
LOCS=   394  LOCU=   396  ROWS=     0  COLUMNS=   11.  VARIABLE E
LOCS=   396  LOCU=   417  ROWS=     5  COLUMNS=    4.  VARIABLE F
LOCS=   417  LOCU=   430  ROWS=     4  COLUMNS=    3.  VARIABLE G
LOCS=   430  LOCU=   432  ROWS=     0  COLUMNS=   11.  VARIABLE H
LOCS=   432  LOCU=   434  ROWS=     0  COLUMNS=   11.  VARIABLE I
LOCS=   434  LOCU=   436  ROWS=     0  COLUMNS=   11.  VARIABLE J
LOCS=   436  LOCU=   438  ROWS=     0  COLUMNS=   11.  VARIABLE K
LOCS=   438  LOCU=   440  ROWS=     0  COLUMNS=   11.  VARIABLE L
LOCS=   440  LOCU=   442  ROWS=     0  COLUMNS=   11.  VARIABLE M
LOCS=   442  LOCU=   444  ROWS=     0  COLUMNS=   11.  VARIABLE N
LOCS=   444  LOCU=   446  ROWS=     0  COLUMNS=   11.  VARIABLE O
LOCS=   446  LOCU=   448  ROWS=     0  COLUMNS=   11.  VARIABLE P
LOCS=   448  LOCU=   450  ROWS=     0  COLUMNS=   11.  VARIABLE Q
LOCS=   450  LOCU=   572  ROWS=    11  COLUMNS=   11.  VARIABLE R
                     R(   25) OR R(   2,    3) =        4          DATA(  476)
LOCS=   572  LOCU=   574  ROWS=     0  COLUMNS=   11.  VARIABLE S
LOCS=   574  LOCU=   576  ROWS=     0  COLUMNS=   11.  VARIABLE T
LOCS=   576  LOCU=   578  ROWS=     0  COLUMNS=   11.  VARIABLE U
LOCS=   578  LOCU=   580  ROWS=     0  COLUMNS=   11.  VARIABLE V
LOCS=   580  LOCU=   582  ROWS=     0  COLUMNS=   11.  VARIABLE W
LOCS=   582  LOCU=   584  ROWS=     0  COLUMNS=   11.  VARIABLE X
LOCS=   584  LOCU=   586  ROWS=     0  COLUMNS=   11.  VARIABLE Y
LOCS=   586  LOCU=   588  ROWS=     0  COLUMNS=   11.  VARIABLE Z
```

Druckliste aufgrund des zweiten PRINT ALL-Statements.

```
LOCS=   324  LOCU=   347  ROWS=    11  COLUMNS=    2.  VARIABLE A
                     A(    3) OR A(   1,    1) =        1          DATA(  328)
                     A(    5) OR A(   2,    1) =        2          DATA(  330)
                     A(    7) OR A(   3,    1) =        3          DATA(  332)
                     A(    9) OR A(   4,    1) =        4          DATA(  334)
                     A(   11) OR A(   5,    1) =        5          DATA(  336)
LOCS=   347  LOCU=   360  ROWS=     4  COLUMNS=    3.  VARIABLE B
                     B(    4) OR B(   1,    1) =        1          DATA(  352)
                     B(    5) OR B(   1,    2) =        1          DATA(  353)
                     B(    7) OR B(   2,    1) =        1          DATA(  355)
                     B(    8) OR B(   2,    2) =        1          DATA(  356)
                     B(   10) OR B(   3,    1) =        1          DATA(  358)
                     B(   11) OR B(   3,    2) =        1          DATA(  359)
```

```
LOCS=   390  LOCU=   392  ROWS=     0  COLUMNS=   11.  VARIABLE C
LOCS=   392  LOCU=   394  ROWS=     0  COLUMNS=   11.  VARIABLE D
LOCS=   394  LOCU=   396  ROWS=     0  COLUMNS=   11.  VARIABLE E
LOCS=   396  LOCU=   417  ROWS=     5  COLUMNS=    4.  VARIABLE F
LOCS=   417  LOCU=   430  ROWS=     4  COLUMNS=    3.  VARIABLE G
LOCS=   430  LOCU=   432  ROWS=     0  COLUMNS=   11.  VARIABLE H
LOCS=   432  LOCU=   434  ROWS=     0  COLUMNS=   11.  VARIABLE I
LOCS=   434  LOCU=   436  ROWS=     0  COLUMNS=   11.  VARIABLE J
LOCS=   436  LOCU=   438  ROWS=     0  COLUMNS=   11.  VARIABLE K
LOCS=   438  LOCU=   440  ROWS=     0  COLUMNS=   11.  VARIABLE L
LOCS=   440  LOCU=   442  ROWS=     0  COLUMNS=   11.  VARIABLE M
LOCS=   442  LOCU=   444  ROWS=     0  COLUMNS=   11.  VARIABLE N
LOCS=   444  LOCU=   446  ROWS=     0  COLUMNS=   11.  VARIABLE O
LOCS=   446  LOCU=   448  ROWS=     0  COLUMNS=   11.  VARIABLE P
LOCS=   448  LOCU=   450  ROWS=     0  COLUMNS=   11.  VARIABLE Q
LOCS=   450  LOCU=   572  ROWS=    11  COLUMNS=   11.  VARIABLE R
                    R(  25) OR R(  2,   3) =         4          DATA(  476)
LOCS=   572  LOCU=   574  ROWS=     0  COLUMNS=   11.  VARIABLE S
LOCS=   574  LOCU=   576  ROWS=     0  COLUMNS=   11.  VARIABLE T
LOCS=   576  LOCU=   578  ROWS=     0  COLUMNS=   11.  VARIABLE U
LOCS=   578  LOCU=   580  ROWS=     0  COLUMNS=   11.  VARIABLE V
LOCS=   580  LOCU=   582  ROWS=     0  COLUMNS=   11.  VARIABLE W
LOCS=   582  LOCU=   584  ROWS=     0  COLUMNS=   11.  VARIABLE X
LOCS=   584  LOCU=   586  ROWS=     0  COLUMNS=   11.  VARIABLE Y
LOCS=   586  LOCU=   588  ROWS=     0  COLUMNS=   11.  VARIABLE Z
```

322.33 Adressierung numerischer Konstanter

Das Compilierungssystem reserviert Speicherplätze für explizit angegebene numerische Konstante und Hilfsspeicherbereiche für FOR/NEXT-Schleifen.
Der Bereich dieser Speicherplätze schliesst sich direkt an den für jedes BASIC-Programm notwendigen Adressteil mit der untersten Adresse DATA(324) an. Das Compilierungssystem erweitert den Adressbereich um eine Adresse, sobald für eine abzuspeichernde Konstante noch keine Speicherplatzadresse vergeben wurde. Jede FOR/NEXT-Schleife führt ebenfalls zu einer Erweiterung des Adressbereichs, da Laufbereichsanfang, Laufbereichsende und eine explizit angebene Schrittweite sich aus einem arithmetischen Ausdruck ergeben können[40].
Das folgende BASIC-Programmbeispiel zeigt die Adressierung der numerischen Konstanten im Adressbereich des Vektors DATA, wobei unter „LOCATION" die Adresse und unter „VALUE" der jeweilige Wertinhalt angegeben sind. Die Adress- und Wertangaben der Konstanten sind grundsätzlich nur mit dem DUMP-Statement[41] zu erhalten.

```
10  REM SPEICHERUNG NUMERISCHER KONSTANTEN
20  LET A=123                        ! 123 STEHT IN LOCATION 324
30  LET A1=22.3                      ! 22.3 STEHT IN LOCATION 325
40  LET B(1,2)=17                    !  17 STEHT IN LOCATION 326
50  DUMP
60  FOR I=B(1,2) TO A STEP A1        ! LAUFBEREICHSANFANG STEHT IN LOCATION 327
70    PRINT I,                       ! LAUFBEREICHSENDE STEHT IN LOCATION 328
80  NEXT I                           ! SCHRITTWEITE STEHT IN LOCATION 329
90  PRINT A; A1, B(1,2)
100 DUMP
110 END
```

40 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 324.1.
41 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 333.21.

```
****************** EVERYTHING SEEMS OK -- LET'S GO AHEAD

                    PERCENT OF AVAILABLE STORAGE USED              15.946
                    PERCENT OF AVAILABLE DATA STORAGE USED          0.000
                    PERCENT OF AVAILABLE NUMBERED STATEMENTS USED   3.235
```

DUMP vor Abarbeitung der FOR/NEXT-Schleife.

```
 CONSTANTS

LOCATION        VALUE
     314            0.00000
     315            1.00000
     316            2.00000
     317            3.00000
     318            4.00000
     319            5.00000
     320            6.00000
     321            7.00000
     322            8.00000
     323            9.00000
     324          123.00000
     325           22.30000
     326           17.00000
     327            0.00000
     328            0.00000
     329            0.00000
```

DUMP nach Abarbeitung der FOR/NEXT-Schleife.

```
 CONSTANTS

LOCATION        VALUE
     314            0.00000
     315            1.00000
     316            2.00000
     317            3.00000
     318            4.00000
     319            5.00000
     320            6.00000
     321            7.00000
     322            8.00000
     323            9.00000
     324          123.00000
     325           22.30000
     326           17.00000
     327           17.00000
     328          123.00000
     329           22.30000
```

322.34 Adressierung alphanumerischer Konstanter

Im Gegensatz zu den numerischen Konstanten richtet das Compilierungssystem für alphanumerische Konstante (strings)[42] keine Speicherplätze im Adressbereich von DATA ein, sondern speichert den verschlüsselten Text unmittelbar nach dem zur Stringverarbeitung erforderlichen Operationscode im FOSBIC-Code Programm. Dadurch erhalten gleiche alphanumerische Konstante, die in einem BASIC-Programm an verschiedenen Stellen vorkommen, unterschiedliche Speicheradressen.

42 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 322.2.

Ein Speicherplatz für alphanumerische Konstante kann fünf Zeichen aufnehmen. Ist die angegebene alphanumerische Konstante kleiner als fünf Zeichen, so wird der Speicherplatz rechtsbündig mit Leerzeichen aufgefüllt. Umfasst die Zeichenkette mehr als fünf Zeichen, so werden linksbündig die ersten fünf Zeichen aufgenommen, der Rest der Zeichenkette geht verloren. Eine Ausnahme bilden die alphanumerischen Konstanten im PRINT-Statement, die in eine entsprechende Zahl von Zeichenpaketen zu jeweils fünf Zeichen unterteilt und abgespeichert werden. Zeichenpakete zu jeweils fünf Zeichen werden nach einem speziellen Schlüsselsystem im Programm STRING in Ganzzahlen umgewandelt und bei der Textausgabe entsprechend zurücktransformiert.
Die Transformation aller alphanumerischen Konstanten in Ganzzahlen erlaubt es, den Wertinhalt von Variablen beliebig zu ändern, ohne dass dabei auf ursprüngliche numerische oder alphanumerische Zuordnung geachtet werden muss. Ein spezieller Adressbereich für alphanumerische Variable entfällt. Der innerhalb von BASIC übliche Zusatz eines $-Zeichens bei alphanumerischen Variablen ist lediglich ein Hinweis für den Benutzer und eine Kontrollmöglichkeit für das Compilierungssystem.

```
          10  REM SPEICHERUNG ALPHANUMERISCHER KONSTANTER
          20  LET A1$="AAAAA"          ' AAAAA=0 STEHT IN ADRESSE 3696
          30  LET A2$="BAAAA"          ' BAAAA=1 STEHT IN ADRESSE 3690
          40  LET A3$="CAAAA"          ' CAAAA=2 STEHT IN ADRESSE 3684
          50  LET A4$="AAAAB"          ' AAAAB=5308420 STEHT IN ADRESSE 3675
          60  FOR I=3216 TO 3265
          70    PRINT I,I$,
          80  NEXT I
          90  PRINT
         100 PRINT "SPEICHERUNG UND CODIERUNG ALPHANUMERISCHER KONSTANTER"
         110 PRINT A1,A1$,A2,A2$,A3,A3$,A4,A4$
         120 DUMP
         130 END

****************** EVERYTHING SEEMS OK -- LET'S GO AHEAD

                    PERCENT OF AVAILABLE STORAGE USED              13.459
                    PERCENT OF AVAILABLE DATA STORAGE USED          0.000
                    PERCENT OF AVAILABLE NUMBERED STATEMENTS USED   3.824

 3216    ATBAA          3217    BTBAA          3218    CTBAA          3219    DTBAA
 3220    ETBAA          3221    FTBAA          3222    GTBAA          3223    HTBAA
 3224    ITBAA          3225    JTBAA          3226    KTBAA          3227    LTBAA
 3228    MTBAA          3229    NTBAA          3230    OTBAA          3231    PTBAA
 3232    QTBAA          3233    RTBAA          3234    STBAA          3235    TTBAA
 3236    UTBAA          3237    VTBAA          3238    WTBAA          3239    XTBAA
 3240    YTBAA          3241    ZTBAA          3242    0TBAA          3243    1TBAA
 3244    2TBAA          3245    3TBAA          3246    4TBAA          3247    5TBAA
 3248    6TBAA          3249    7TBAA          3250    8TBAA          3251    9TBAA
 3252    *TBAA          3253     TBAA          3254    .TBAA          3255    .TBAA
 3256    =TBAA          3257    )TBAA          3258    (TBAA          3259    +TBAA
 3260     TBAA          3261    $TBAA          3262    -TBAA          3263    /TBAA
 3264    AUBAA          3265    BUBAA
SPEICHERUNG UND CODIERUNG ALPHANUMERISCHER KONSTANTER
    0.0   AAAAA             1   BAAAA             2   CAAAA       5308420   AAAAB
PROGRAM DUMP

STATEMENT NUMBER      10  INTERNAL LOCATION =    3700
          -12

STATEMENT NUMBER      20  INTERNAL LOCATION =    3699
           -1          2         -20          0         -8         80
```

```
 STATEMENT NUMBER      30  INTERNAL LOCATION =    3693
              -1          2         -20          1          -8         106

 STATEMENT NUMBER      40  INTERNAL LOCATION =    3687
              -1          2         -20          2          -8         132

 STATEMENT NUMBER      50  INTERNAL LOCATION =    3681
              -1          2         -20    5308416          -8         158

 STATEMENT NUMBER      60  INTERNAL LOCATION =    3675
              -1          1         324         -8         325          -1           1         328          -8         326
              -1          3         325        327          -2          -9           9         -15           9         325
            3641

 STATEMENT NUMBER      70  INTERNAL LOCATION =    3654
              -1          2           9         -7         -17         -67          -1           2           9          -7
             -30

 STATEMENT NUMBER      80  INTERNAL LOCATION =    3643
              -4        3658         -12

 STATEMENT NUMBER      90  INTERNAL LOCATION =    3640
             -10

 STATEMENT NUMBER     100  INTERNAL LOCATION =    3639
             -16          3    11511522    71260615    17411142         -16           3    42831500    33335092    38843180
             -16          3    22607472    37421969    75525940         -16           3    69054061   200542675   200590357
             -18

 STATEMENT NUMBER     110  INTERNAL LOCATION =    3618
              -1          1          80         -17         -67          -1           1          80          -30          -1
               1        106         -17          -1           1         106         -30          -1           1         132
             -17         -1           1         132         -30          -1           1         158          -17          -1
               1        158         -30         -18

 STATEMENT NUMBER     120  INTERNAL LOCATION =    3584
             -19

 STATEMENT NUMBER     130  INTERNAL LOCATION =    3583
             -11

 CONSTANTS

LOCATION         VALUE
     314           0.00000
     315           1.00000
     316           2.00000
     317           3.00000
     318           4.00000
     319           5.00000
     320           6.00000
     321           7.00000
     322           8.00000
     323           9.00000
     324        3216.00000
     325        3216.00000
     326        3265.00000
     327           1.00000
```

323 FOSBIC-Code Programm

Im Gegensatz zu maschinenorientierten Compilern erzeugt das FOSBIC-Compilersystem ein maschinenunabhängiges Object-Code Programm, was hier als FOSBIC-Code Programm bezeichnet wird.

323.1 Aufbau des FOSBIC-Code Programms

Das vom Compilierungssystem erzeugte FOSBIC-Code Programm wird in dem Vektor IPROG gespeichert. Dabei werden die Ausführungsbefehle, die vergleichbar sind mit den Operationscode eines normalen Object-Time-Programms, alle Adressen und alle alphanumerischen Konstanten beginnend bei der höchsten Adresse von IPROG in fallender sequentieller Folge gespeichert. Die Speicherung der Variablen und aller numerischen Konstanten erfolgt beginnend bei der Adresse 1 im Vektor DATA. Die beiden Vektoren DATA und IPROG müssen gleiche Feldgrössen erhalten und werden durch eine EQUIVALENCE-Anweisung identisch übereinandergelegt. Abbildung 13 zeigt den formalen Aufbau eines FOSBIC-Code Programms.

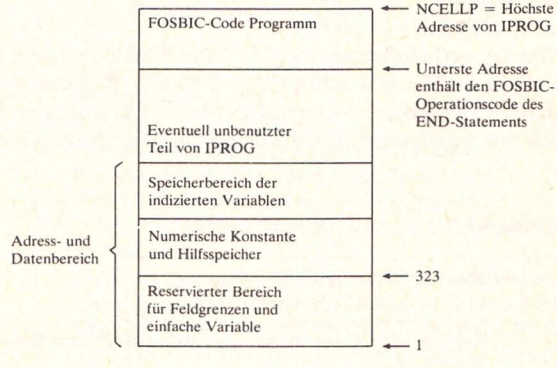


Abbildung 13. Aufbau des FOSBIC-Code Programms

Im Gegensatz zu der herkömmlichen Aufteilung eines Befehlswortes in Operations-, Index- und Adressteil enthält das FOSBIC-Code Programm je Wort nur einen Operationscode oder nur eine Adresse.
Innerhalb jedes FOSBIC-Code Programms existiert im Vektor DATA ein unterer Bereich mit den Adressen 1 bis 323, der zur Speicherung von Adressen einfacher Variabler, der Feldanfänge und der Feldgrenzen indizierter Variabler reserviert ist. An diesen Bereich schliesst sich die Speicherung der im Programm explizit vorgegebenen numerischen Konstanten an. Für die im Programm enthaltenen FOR/NEXT-Schleifen

werden in diesem Bereich sogenannte Hilfsregister für den Laufbereichsanfang, die Laufbereichsgrenze und die Schrittweite angelegt. Abgeschlossen wird der Adressbereich durch die Speicherfelder der indizierten Variablen.
Zwischen dem Adressbereich, der grundsätzlich nur über den Vektor DATA angesprochen wird und dem FOSBIC-Code Programm, das nur über den Vektor IPROG erreicht wird, liegt gewöhnlich ein unbenutzter Teil. Wird dieser Teil kleiner oder gleich Null, so liegt eine Überspeicherung vor, d. h. Programm bzw. Variablenfelder verlangen mehr Speicherplatz als verfügbar ist. Das Compilierungssystem zeigt dies durch eine entsprechende Meldung an.

323.2 FOSBIC-Operationscode

Das FOSBIC-Code Programm setzt sich aus einer sequentiellen Folge von allgemeinen und speziellen Operationsbefehlen und Adressen zusammen, wobei je Wort nur ein Befehl oder eine Adresse gespeichert wird.

323.21 Allgemeine Operationsbefehle

Die Liste der allgemeinen Operationsbefehle umfasst 71 Codeschlüssel, die alle negative Ganzzahlen darstellen[43]. Die folgende Tabelle 3 enthält alle Schlüssel in fallender Reihenfolge der Codenummern, die BASIC-Statements aus der der Codeschlüssel erzeugt werden kann und eine Kurzerläuterung, der auf Grund dieses Operationscode durchgeführten Ausführungen im Ausführungssystem des FOSBIC-Compilersystems.

Tabelle 3. Allgemeine Operationsbefehle

Code-schlüssel	BASIC-anweisung	Kurzbezeichnung der Ausführung
–1 E	LET	Beginn einer Operations- und Adresskette zur Berechnung eines arithmetischen Ausdrucks
–2 E	GOSUB	Speichern der Rücksprungadresse aus einem Unterprogramm in NIRET
–3 E	GOTO	Sprung auf Grund einer extern vorgegebenen Statementnummer
–4 E	NEXT/GOTO	Sprung auf Grund einer internen Adresse
–5 E	RETURN	Rücksprung aus einem Unterprogramm
–6 E	IF	Durchführung eines Vergleichs
–7 E	READ	Lesen eines Wertes in den Akkumulator ACC.

43 Version 6/76–04.

Code-schlüssel		BASIC-anweisung	Kurzbezeichnung der Ausführung
–8	X		Umspeichern eines Wertes aus dem Akkumulator in den Adressbereich direkt
–9	X		Umspeichern eines Wertes aus dem Akkumulator in den Adressbereich indirekt (indiziert)
–10	E	PRINT	Vorschub um eine Leerzeile
–11	E	STOP/END	Logisches Ende des FOSBIC-Code-Programms
–12	E	REM	Leerbefehl
–13	E	PAGE	Vorschub um eine Seite
–14	E	IF	Laden des Akkumulators in Vergleichsregister XXX (1) oder XXX (2)
–15	E	FOR	Setzen der Adressen für eine Schleife, Initialisierung der Schleife
–16	E	PRINT	Druckaufbereitung alphanumerischer Werte innerhalb des Standard-PRINT
–17	E	PRINT	Druckaufbereitung numerischer Werte innerhalb des Standard-PRINT
–18	E	PRINT	Druckausgabe der Standard-PRINT-Zeile
–19	E	DUMP	DUMP des FOSBIC-Code-Programms, der numerischen Konstanten und der Hilfsspeicher
–20	E	PRINT ALL	Auflistung aller Variablenwerte, die größer als ± SMALL sind.
–21	E	STOP/END	Sprung zum Ende des Ausführungssystems.
–22	E	RESTORE	Setzen des Positionszeigers im Vektor DATAN
–23	M	MAT	Matrizenaddition
–24	M	MAT	Matrizenmultiplikation
–25	M	MAT	Matrizensubtraktion
–26	M	TRN	Matrizentransponierung
–27	M	GOTO/ON	Sprung auf Grund eines bedingten GOTO
–28	M	MAT READ	Lesen eines Feldes ohne Redimensionierung
–29	M	MAT READ	Lesen eines Feldes mit aktuel. Feldgrenzen
–30	E	PRINT	Alphanumerischer Variablenwert über Akkumulator zur Druckaufbereitung bereit stellen
–31	M	MAT PRINT	Drucken eines numerischen Feldes ohne Redimensionierung
–32	M	MAT PRINT	Drucken eines numerischen Feldes mit aktuellen Feldgrenzen
–33	M	MAT PRINT	Drucken eines alphanumerischen Feldes ohne Redimensionierung
–34	M	MAT PRINT	Drucken eines alphanumerischen Feldes mit aktuellen Feldgrenzen
–35	M	IDN	Bilden einer Einheitsmatrix ohne Redimensionierung
–36	M	IDN	Bilden einer Einheitsmatrix mit Redimensionierung
–37	M	CON	Bilden einer Einermatrix ohne Redimensionierung
–38	M	CON	Bilden einer Einermatrix mit Redimensionierung
–39	M	ZER	Bilden einer Nullmatrix ohne Redimensionierung
–40	M	ZER	Bilden einer Nullmatrix mit Redimensionierung
–41	M	MAT LET	Akkumulatorwert in ein Feld laden

Fortsetzung der Tabelle 3.

Code-schlüssel		BASIC-anweisung	Kurzbezeichnung der Ausführung
–42	M	MAT LET	Umspeichern eines Feldes in ein anderes Feld
–43	M	MAT LET	Skalaroperation mit einem Feld, wobei sich der Skalar aus einem arithmetischen Ausdruck ergeben kann
–44	E	GOSUB/ON	Ausführung eines bedingten Sprungs in ein Unterprogramm
–45	E	INPUT	Lesen eines Wertes zum INPUT-Befehl
–46	M	MAT INPUT	Lesen eines Feldes ohne Redimensionierung
–47	M	MAT INPUT	Lesen eines Feldes mit aktuellen Feldgrenzen
–48	F	OPEN	Öffnen und definieren eines Files
–49	F	PUT	Akkumulator auf ein sequentielles File schreiben
–50	F	PUT	Akkumulator auf ein index-sequentielles File schreiben
–51	F	GET	Lesen eines Wortes eines sequentiellen Files in den Akkumulator
–52	F	GET	Lesen eines Wortes eines index-sequentiellen Files in den Akkumulator
–53	F	RESET	Zurücksetzen eines sequentiellen Files
–54	F	RESET	Positionieren eines index-sequentiellen Files
–55	F	CLOSE	Schließen und löschen eines Files
–56	F	RESET	Wortposition in einem Satz vom Akkumulator nach MPOINT-Register laden
–57	F	RESET	Satznummer eines index-sequentiellen Files von Akkumulator nach MSATZ-Register laden
–58	F	MAT GET	Lesen eines Feldes von einem sequentiellen File
–59	F	MAT GET	Lesen eines Feldes von einem index-sequentiellen File
–60	F	MAT PUT	Schreiben eines Feldes auf ein sequentielles File
–61	F	MAT PUT	Schreiben eines Feldes auf ein index-sequentielles File
–62	E	%	Aktualisieren einer IMAGE-Maske aus dem IMAGE-File
–63	E	PRINT USING	Druckaufbereitung einer alphanumerischen Konstante innerhalb des PRINT USING
–64	E	PRINT USING	Druckaufbereitung eines numerischen Akkumulatorwertes innerhalb des PRINT USING
–65	E	PRINT USING	Drucken einer Zeile innerhalb des PRINT USING
–66	E	PRINT USING	Druckaufbereitung eines alphanumerischen Variablenwertes innerhalb des PRINT USING
–67	X	.	Umschalten auf Standardeinteilung der Druckzeile
–68	X	;	Umschalten auf gepackte Einteilung der Druckzeile
–69	E	TAB	Tabulieren auf eine Druckposition, Positionswert in ACC
–70	E	PRINT USING	PRINT USING Druckaufbereitung ohne Druckliste
–71	M	INV	Inversion

E = Elementaranweisungen
F = Fileanweisungen
M = Matrizenanweisungen
X = Allgemeiner Operationscode

Fortsetzung der Tabelle 3.

323.22 Spezielle Operationsbefehle

Die Auflösung der arithmetischen Ausdrücke durch das Unterprogramm ZTRANX geschieht nach der sogenannten inversen polnischen Notation[44]. Als Operationsbefehle werden hierzu spezielle Codeschlüssel innerhalb des FOSBIC-Code Programms benutzt. Die FOSBIC-Version 6/76–04 umfasst 20 Codeschlüssel, die vor allem durch die Anzahl der intrinsic (eingebauten) Funktionen bestimmt werden. Eine Codefolge der speziellen Operationsbefehle wird stets durch den allgemeinen Operationsbefehl −1 eingeleitet.

Tabelle 4. Spezielle Operationsbefehle

Codeschlüssel	Ausführung
−1	Addition
−2	Subtraktion
−3	Multiplikation
−4	Division
−5	Exponentiation
−6	Vorzeichenumkehr
−7	Einfache Variable in Akkumulator laden
−8	Indizierte Variable in Akkumulator laden
−9	Sinusberechnung
−10	Cosinusberechnung
−11	Tangensberechnung
−12	Arcustangensberechnung
−13	Exponentiation zur Basis e
−14	Bilden des Absolutwertes
−15	Logarithmus zur Basis e
−16	Quadratwurzelberechnung
−17	Ganzzahl bilden durch Abschneiden
−18	Zufallszahl bilden
−19	Akkumulator in indizierte Variable laden
−20	Alphanumerische Variable in Akkumulator laden

44 Higman, Programmiersprachen, S. 43–45

323.3 FOSBIC-Code Beispiele

Das FOSBIC-Compilersystem verfügt über ein spezielles Diagnosestatement zur Kontrolle des Compilierungsablaufs und des erzeugten FOSBIC-Code Programms. Mittels des DUMP-Statements[45] wird der Inhalt des Programmspeichers (Vektor IPROG) beginnend bei der obersten Adresse NCELLP[46] bis zum Operationscode des END-Statements ausgedruckt. Dieser FOSBIC-Code Programmausdruck erfolgt nur, wenn mindestens ein BASIC-Statement numeriert und das END-Statement vorhanden ist.
Der DUMP-Ausdruck beginnt beim ersten numerierten BASIC-Statement und wird in Zeilen zu maximal 10 Operationsschlüssel eingeteilt. Für jedes numerierte BASIC-Statement wird ein neuer Abschnitt begonnen, der durch die Angabe der Statementnummer und der ersten Programmadresse (internal location) dieses BASIC-Statements eingeleitet wird. Abgeschlossen wird ein DUMP-Ausdruck durch die Liste der numerischen Konstanten und Hilfsspeicher des Adressbereichs, die auf jeden Fall die numerischen Konstanten 0 bis 9 umfasst.
Das Compilierungssystem erzeugt aus dem einfachen BASIC-Programm

```
10 DUMP
20 PRINT 1234
30 END
```

folgendes FOSBIC-Code Programm:

```
PROGRAM DUMP

STATEMENT NUMBER      10  INTERNAL LOCATION =    3700
          -19

STATEMENT NUMBER      20  INTERNAL LOCATION =    3699
          -1          1       324       -17         -18

STATEMENT NUMBER      30  INTERNAL LOCATION =    3694
          -11
```

Die Bedeutung der einzelnen FOSBIC-Codeschlüssel können Tabelle 3 und 4 entnommen werden. Das PRINT-Statement wird demnach in einen arithmetischen Ausdruck einfachster Struktur (−1,1,324), in die Druckaufbereitung numerischer Werte (−17) und in die Ausgabe der Standarddruckzeile (−18) aufgelöst. Der arithmetische Ausdruck wird durch den allgemeinen Operationscode −1 eingeleitet. Die nächstfolgende Ganzzahl gibt die Anzahl der Adressen und speziellen Operationsschlüssel an, aus denen sich der arithmetische Ausdruck zusammengesetzt. In diesem einfachen Fall folgt nur die Adresse der numerischen Konstanten.

45 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 333.21.
46 Vgl. Tabelle 9.

Etwas komplexer ist der DUMP-Ausdruck des folgenden BASIC-Programms:

```
10 DUMP
20 LET A1=105
30 LET A2=A1*3
40 LET A3=(A2+A1)/A1
50 PRINT *ERGEBNISSE*,A1,A2,A3
60 END

PROGRAM DUMP

STATEMENT NUMBER      10  INTERNAL LOCATION =    3700
          -19

STATEMENT NUMBER      20  INTERNAL LOCATION =    3699
           -1          1         324          -8          80

STATEMENT NUMBER      30  INTERNAL LOCATION =    3694
           -1          3          80         317          -3          -8         106

STATEMENT NUMBER      40  INTERNAL LOCATION =    3687
           -1          5         106          80          -1          80          -4          -8         132

STATEMENT NUMBER      50  INTERNAL LOCATION =    3678
          -16          3     5765428    23266180   200590357         -67          -1           1          80         -17
           -1          1         106         -17          -1           1         132         -17         -18

STATEMENT NUMBER      60  INTERNAL LOCATION =    3659
          -11
```

Die FOSBIC-Codefolge des BASIC-Statements 40 ist wie folgt zu interpretieren:

–1	Anfang eines arithmetischen Ausdrucks
5	Anzahl der Adressen und speziellen Operationscodes des arithmetischen Ausdrucks
106	Adresse der einfachen BASIC-Variablen A2 im Adressbereich des Vektors DATA
80	Adresse von A1 im Adressbereich von DATA
–1	spezieller Operationscode zur Addition der beiden vorausgehenden Adressen und Speicherung des Ergebnisses in einem Akkumulator
80	Adresse von A1 im Adressbereich von DATA
–4	Division des Akkumulatorwertes durch die vorausgehende Adresse und Abspeichern des Ergebnisses im Akkumulator
–8	allgemeiner Operationscode zur direkten Wertübertragung aus dem Akkumulator in die nachfolgend angegebene Adresse
132	Adresse von A3 im Adressbereich von DATA.

Durch die Verwendung von Matrizenanweisungen lässt sich die Länge eines FOSBIC-Code Programms erheblich verkürzen, wie im folgenden Programmbeispiel durch die Gegenüberstellung von BASIC-Elementar- und Matrizenanweisungen mit gleichem Ausführungsergebnis gezeigt wird.

```
                    10 DUMP
                    20 DIM A(25),B(10,15)
                    30 MAT READ A
                    40 MAT READ B
                    50 FOR I=1 TO 25
                    55   READ A(I)
                    60 NEXT I
                    65 FOR I=1 TO 10
                    70   FOR J=1 TO 15
                    75     READ B(I,J)
                    80   NEXT J
                    90 NEXT I
                    95 END

PROGRAM DUMP

STATEMENT NUMBER      10  INTERNAL LOCATION =    3700
          -19

STATEMENT NUMBER      20  INTERNAL LOCATION =    3699
          -12

STATEMENT NUMBER      30  INTERNAL LOCATION =    3698
          -28
             1

STATEMENT NUMBER      40  INTERNAL LOCATION =    3696
          -28
             2

STATEMENT NUMBER      50  INTERNAL LOCATION =    3694
           -1          1          315          -8          324          -1           1          327          -8          325
           -1          3          324         326           -2          -9           9          -15           9          324
         3664

STATEMENT NUMBER      55  INTERNAL LOCATION =    3673
           -7         -1            4            1           9          -7         -19

STATEMENT NUMBER      60  INTERNAL LOCATION =    3666
           -4       3677          -12

STATEMENT NUMBER      65  INTERNAL LOCATION =    3663
           -1          1          315          -8          328          -1           1          331          -8          329
           -1          3          328         330           -2          -9           9          -15           9          328
         3604

STATEMENT NUMBER      70  INTERNAL LOCATION =    3642
           -1          1          315          -8          332          -1           1          335          -8          333
           -1          3          332         334           -2          -9          10          -15          10          332
         3607

STATEMENT NUMBER      75  INTERNAL LOCATION =    3621
           -7         -1            9            2           9          -7          29           -3          10           -7
           -1        -19

STATEMENT NUMBER      80  INTERNAL LOCATION =    3609
           -4       3625          -12

STATEMENT NUMBER      90  INTERNAL LOCATION =    3606
           -4       3646          -12

STATEMENT NUMBER      95  INTERNAL LOCATION =    3603
          -11
```

Für das Einlesen eines Vektors mit Elementaranweisungen werden 31 Adressen und Codeschlüssel benötigt (Programmadressen 3694 bis 3664). Bei Matrizenanweisungen reichen ein Operationscode und eine

Adresse aus (Programmadressen 3697 und 3698). Bei Matrizen ist das Verhältnis bei Verwendung von Elementar- oder Matrizenanweisungen 60 zu 2.
Das folgende Beispiel illustriert noch einmal den Zusammenhang zwischen BASIC- und FOSBIC-Code Programm. Mit den Tabellen 3 und 4 kann der Leser leicht die Verbindung zwischen BASIC Source-Statement und FOSBIC-Codefolge herstellen.

```
                    10 DUMP
                    15 REM PROGRAMM ZUR BERECHNUNG DES MITTELWERTES UND DER VARIANZ
                    20 READ A
                    30 IF A$=≠ENDE≠ GOTO 55
                    35 LET S1=S1+A
                    40 LET S2=S2+A**2
                    45 LET N=N+1
                    50 GOTO 20
                    55 PRINT ≠   ANZAHL≠,≠MITTELWERT≠,≠  VARIANZ≠
                    60 PRINT N,S1/N,(N*S2-S1**2)/(N*(N-1))
                    65 STOP
                    70 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
                    75 DATA ≠ENDE≠
                    80 END

          ********** EVERYTHING SEEMS OK -- LET≠S GO AHEAD

                     PERCENT OF AVAILABLE STORAGE USED              12.865
                     PERCENT OF AVAILABLE DATA STORAGE USED          5.455
                     PERCENT OF AVAILABLE NUMBERED STATEMENTS USED   4.118


          ANZAHL       MITTELWERT       VARIANZ
            17             9             25.5

PROGRAM DUMP

STATEMENT NUMBER     10  INTERNAL LOCATION =    3700
          -19

STATEMENT NUMBER     15  INTERNAL LOCATION =    3699
          -12

STATEMENT NUMBER     20  INTERNAL LOCATION =    3698
           -7        -9          1

STATEMENT NUMBER     30  INTERNAL LOCATION =    3695
           -1         2          1          -7         -14          1          -1          2          -20   196861300
          -14         2         -6           5          -3         55

STATEMENT NUMBER     35  INTERNAL LOCATION =    3679
           -1         4         98           1          -7         -1          -8         98

STATEMENT NUMBER     40  INTERNAL LOCATION =    3671
           -1         6        124           1          -7        316          -5         -1          -8         124

STATEMENT NUMBER     45  INTERNAL LOCATION =    3661
           -1         4         14          -7         315         -1          -9         14

STATEMENT NUMBER     50  INTERNAL LOCATION =    3653
           -3        20

STATEMENT NUMBER     55  INTERNAL LOCATION =    3651
          -16         3   69112940  197644057  200590357         -67         -16          3   23379084  102750251
```

```
     200590357        -16          3   90293612  199206152  200590357        -18

 STATEMENT NUMBER      60  INTERNAL LOCATION =     3634
            -1         2         14         -7        -17        -67         -1          4         98         14
            -7        -4        -17         -1         16         14         -7        124         -3         98
           316        -5         -2         14         -7         14         -7        315         -2         -3
            -4       -17        -18

 STATEMENT NUMBER      65  INTERNAL LOCATION =     3601
           -11

 STATEMENT NUMBER      70  INTERNAL LOCATION =     3600


 STATEMENT NUMBER      75  INTERNAL LOCATION =     3600


 STATEMENT NUMBER      80  INTERNAL LOCATION =     3600
           -11

 CONSTANTS

LOCATION        VALUE
     314             0.00000
     315             1.00000
     316             2.00000
     317             3.00000
     318             4.00000
     319             5.00000
     320             6.00000
     321             7.00000
     322             8.00000
     323             9.00000
```

324 Ausführungssystem

Dem Ausführungssystem obliegt die Aufgabe, das vom Compilierungssystem erstellte FOSBIC-Code Programm zu interpretieren und auszuführen.

324.1 Aufbaustruktur

Zentrales Programm im Aufbau des Ausführungssystems ist das Programm ZEXEC. Innerhalb der Gesamtjob-Struktur und der OVERLAY-Struktur steuert dieses Programm den gesamten Ausführungsablauf und gibt erst nach Beendigung des FOSBIC-Code Programms oder beim Auftreten eines Programmfehlers den Ablauf
- in der Gesamtjob-Struktur an das Programm MAIN und
- in der OVERLAY-Struktur an das Programm OVERT zurück.

Zum Ausführungssystem gehören weitere 12 Unterprogramme sowie die 5 intermediär benutzten Programme. Innerhalb der OVERLAY-Struktur steht nur der Objekt-Code dieser 17 Programme im Kernspeicher, während die Programme des Compilierungssystems auf einer Magnetplatte ausgelagert sind.

Der Aufbau des Ausführungssystems lässt sich wiederum in verschiedene Module unterteilen. Neben dem Zentralprogramm ZEXEC existieren zwei weitere wichtige Unterprogramme ZEXMAT und ZEXFIL. ZEXEC verarbeitet den FOSBIC-Code der Elementaranweisungen, ZEXMAT den FOSBIC-Code der Matrizenanweisungen und ZEXFIL den FOSBIC-Code der Fileanweisungen.
Neben diesen bedeutsameren Programmen existieren weitere Unterprogramme, die nach Bedarf aufgerufen werden. In diesem Zusammenhang erhalten vor allem die Programme EXERR, ZEVAL, ZINSNO, und ZIMAGE einige Bedeutung. EXERR enthält alle Fehlermeldungen des Ausführungssystems. ZIMAGE bildet auf Grund der IMAGE-Masken[47] die Ausgabezeile der PRINT USING-Anweisung. ZEVAL verarbeitet den FOSBIC-Code von arithmetischen Ausdrücken. Mit ZINSNO wird die Standarddruckzeile aufgebaut.
Abbildung 14 zeigt den Gesamtaufbau des Ausführungssystems.

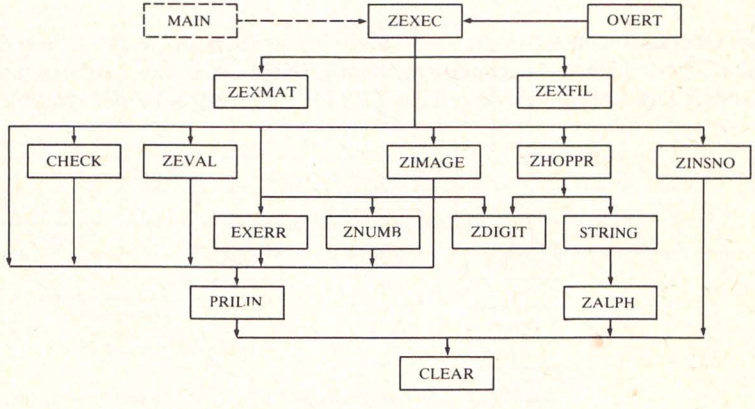


Abbildung 14. Aufbaustruktur des FOSBIC-Ausführungssystems

Ein Operationscode grösser −48 und kleiner als −23 und der Codeschlüssel −71 werden bis auf die Ausnahmen −27, −30 −44 und −45 dem Programm ZEXMAT zugeordnet. Es handelt sich hierbei um die Ausführung einer verschlüsselten Matrizenanweisung. Die Aufbaustruktur dieses Moduls zeigt Abbildung 15.

47 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 332.23

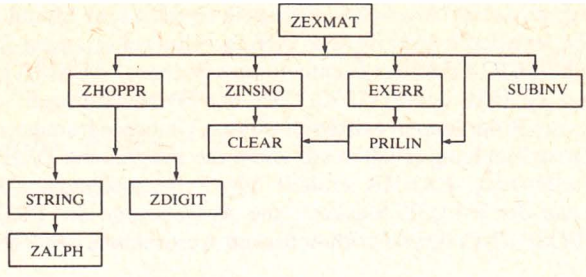


Abbildung 15. Aufbaustruktur des Ausführungsmoduls zu den Matrizenanweisungen

Liegt der Wert des Operationscodes zwischen −48 und −61 so übergibt das Programm ZEXEC die Ausführung der FOSBIC-Codefolge an das Unterprogramm ZEXFIL, da es sich hierbei um verschlüsselte Fileanweisungen handelt. Der Aufbau des Moduls mit ZEXFIL unterliegt dabei der Struktur nach Abbildung 16.

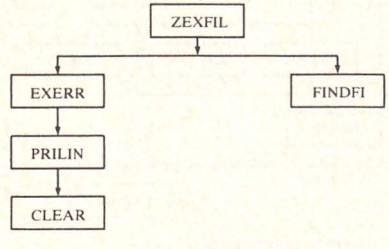


Abbildung 16. Aufbau des Ausführungsmoduls zu den Fileanweisungen

Die Gliederung des Ausführungssystems in drei Module erlaubt dem Anwender verschiedene Stufen des FOSBIC-Compilersystems in einfacher Weise zu implementieren[48].
Abbildung 17 zeigt die Aufbaustruktur des Ausführungssystems in einer Matrixdarstellung.

48 Vgl. Abs. 332.1

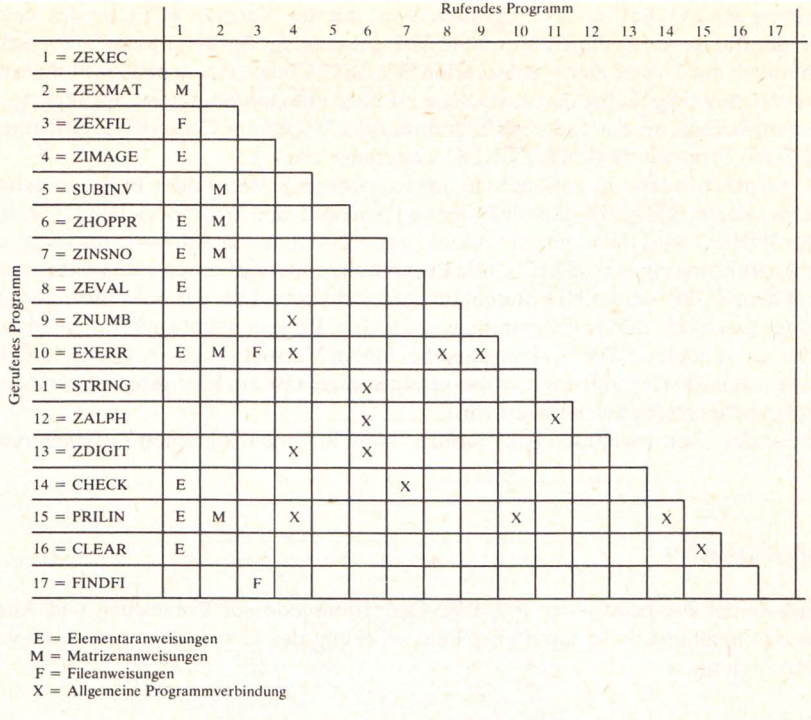


Abbildung 17. Programmatrix des FOSBIC-Ausführungssystems

324.2 Ablaufstruktur

Sobald der Systemablauf an das Ausführungssystem übertragen ist, erfolgt zunächst eine Überarbeitung des vom Compilierungssystem erzeugten FOSBIC-Code Programms.
Diese Überarbeitung ersetzt die Sprungadressen der bedingten GOTO- und GOSUB-Anweisungen, die noch als ursprüngliche Statementnummern vorhanden sind, durch die absoluten Adressen des FOSBIC-Codes. Nach einer kurzen Systemvorbereitung beginnt das Programm ZEXEC mit der Abarbeitung des FOSBIC-Code Programms.

Bei der Bearbeitung des FOSBIC-Code Programms wird mit der Adresse NCELLP des Vektors IPROG beginnend ein Operationscode in die Hilfsvariable IOP geladen. Ist der Inhalt von IOP positiv, so erfolgt ein Programmabbruch auf Grund eines fehlerhaften FOSBIC-Codes. Bei negativem Inhalt von IOP wird zunächst das Vorzeichen umgekehrt und anschliessend über eine umfangreiche Sprungliste, die Ausführung des Operationsbefehls an ein anderes Programm oder bei einem Code der Elementaranweisungen an einen bestimmten Programmteil von ZEXEC weitergegeben.
Der durch den entsprechenden Operationscode angesprochene System- oder Programmabschnitt verarbeitet die dem Operationscode nachstehende Folge von Adressen und/oder speziellen Operationsbefehlen. Das Zählregister INREG wird dabei um die Anzahl der entnommenen Adressen und Operationsbefehle verringert. Die Ausführung eines FOSBIC-Code Programms endet nur durch einen entsprechenden Operationsbefehl der dem STOP- oder END-Statement entspricht oder, falls das Ausführungssystem einen semantischen Fehler entdeckt, durch Programmabbruch. Ein Programmabbruch durch Maschinen- oder Dienstsystem der verwendeten EDV-Anlage kann bei einem Verstoss gegen Angaben in den maschinenorientierten Systemsteuerkarten auftreten, wobei erfahrungsgemäss am häufigsten gegen Zeit-, Zeilenausgabe- oder Kernspeichergrenzen verstossen wird.
Abbildung 18 zeigt den eben erwähnten Ausführungsablauf in seinem logischen Zusammenhang als Flowchart.

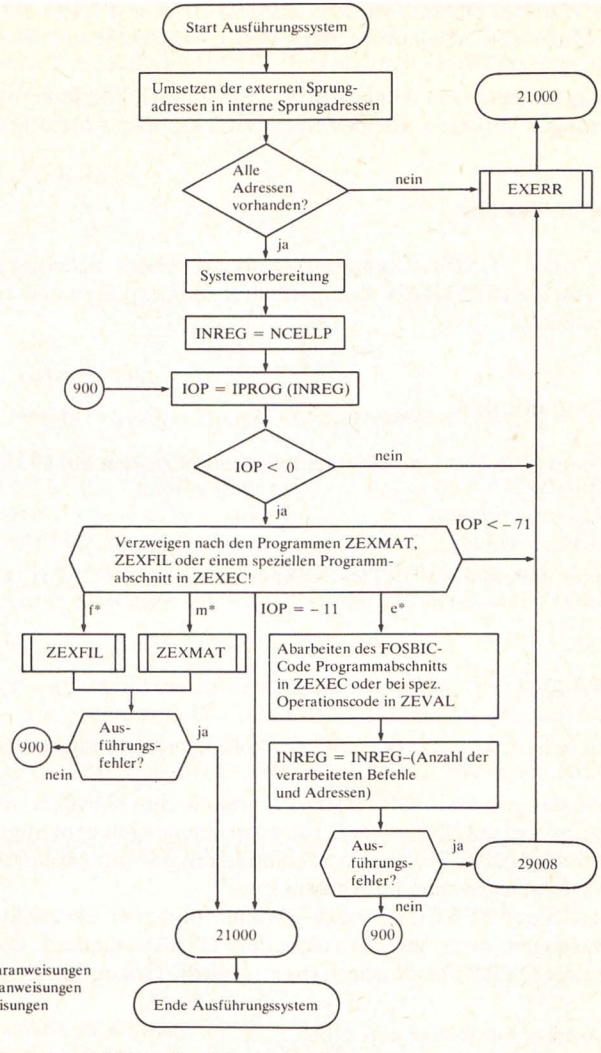
324.3 Ausführungsfehler

Das Ausführungssystem überprüft jeden FOSBIC-Operationscode auf Vorzeichen und Absolutwert. Ist der Operationscode nichtnegativ so kann eine Fehlcodierung des Compilierungssystems vorliegen, was durch die Fehlermeldungen

```
PROGRAM STOPPED -- ERROR AT  3700   CODE=          -11
                   ERROR CODE=  9
             ***** SEE BASIC TEXTBOOK =  C  324.3  *****
             ***** I WAS AT LINE NUMBER  123


PROGRAM STOPPED --
                   MAT CODE ERROR AT  3700 CODE=          -11
                   ERROR CODE= 11
             ***** SEE BASIC TEXTBOOK =  C  324.3  *****
             ***** I WAS AT LINE NUMBER  123


PROGRAM STOPPED --
                   FILE CODE ERROR AT  3700 CODE=          -11
                   ERROR CODE= 12
             ***** SEE BASIC TEXTBOOK =  C  324.3  *****
             ***** I WAS AT LINE NUMBER  123
```



*) e = Operationscode der Elementaranweisungen
m = Operationscode der Matrizenanweisungen
f = Operationscode der Fileanweisungen
Vgl. auch Tabelle 10.

Abbildung 18. Ablaufstruktur des FOSBIC-Ausführungssystems

angezeigt wird. Liegt der Wert des Operationscodes ausserhalb der in Tabelle 3 angegebenen Codewerte ($-71 \leq IOP \leq -1$), so wird ein DUMP und ein PRINT ALL-Statement ausgeführt und das BASIC-Programm abgebrochen.
Die obigen Fehlermeldungen können nur dann auftreten, wenn durch Manipulationen am Compilierungssystem Fehlprogrammierungen vorliegen, die den Systemaufbau oder Ablauf gestört haben.

33 FOSBIC-Implementation

Bei der Implementation[49] des FOSBIC-Compilersystems auf einem beliebigen Computersystem mit FORTRAN-IV Compiler ist die FORTRAN-Kompatibilität zu überprüfen und der zu realisierende BASIC-Sprachumfang festzulegen.

331 FORTRAN-Kompatibilität

Bei der FOSBIC-Implementation können Inkompatibilitäten bezüglich der FORTRAN-Statements zur
- OVERLAY-Struktur,
- END-OF-FILE Abfrage,
- Fileverarbeitung und

bezüglich des Zeichencodes in dem die FORTRAN-Anweisungen des FOSBIC-Compilersystems erstellt sind, auftreten.

331.1 OVERLAY-Struktur

Um Kernspeicherplatz zu sparen, wird das FOSBIC-Compilersystem vorzugsweise in OVERLAY-Struktur implementiert.
Diese Struktur erlaubt es, das gesamte FORTRAN-Programm einschliesslich aller Unterprogramme in zwei Elemente aufzuteilen, wobei jeweils nur ein Element zusammen mit dem Steuerelement im Kernspeicher steht. Je nachdem, ob ein BASIC-Programm zu compilieren oder auszuführen ist, steht das Compilierungssystem oder das Ausführungssystem im Kernspeicher[50].
Die Hauptprogramme MAIN und ZEXEC der beiden Module enthalten die zur Systemsteuerung notwendigen FORTRAN-Anweisungen; sie entsprechen nicht dem USASI-Standard. Diese Anweisungen sowie das gesamte Steuerprogramm OVERT sind zu entfernen, wenn die Gesamtjob-Struktur verwendet werden

49 Horning, Structuring Compiler Development, p. 505.
50 Vgl. Abb. 6

soll. Hierzu sind in den betreffenden Programmen ausreichend Kommentare eingefügt, die die notwendigen Änderungen angeben und erläutern[51]. In Gesamtjob-Struktur erhöht sich der Kernspeicherbedarf um ca. 40 Prozent. Die Bearbeitungszeit von BASIC-Programmen verringert sich dabei nur geringfügig (ca. 0.5 Sekunden pro BASIC-Programm für die CD 3300).

331.2 END-OF-FILE Abfrage

In den Programmen
- MAIN,
- ZHOPPR,
- ZEXEC und
- ZEXFIL

wird mit der FORTRAN-Anweisung

IF(IFEOF(⟨Kanalnummer⟩).EQ.−1) GOTO ⟨Statementnummer⟩

auf eine END-OF-FILE(EOF) Marke abgefragt.
Bei der Übernahme des FOSBIC-Compilersystems ist zu prüfen, ob die END-OF-FILE Abfrage von dem verwendeten FORTRAN-Compiler in der vorliegenden Form akzeptiert und richtig interpretiert wird.
Wird als Kanalnummer IRC angegeben, so handelt es sich um den Standardeingabekanal des Lochkartenlesers. In allen anderen Fällen wird auf das Fileende einer Datei abgefragt.
Im FOSBIC-Compilersystem existieren zwei Standardfiles mit den Namen (Data Set Identifier = DSI)

999 und NIMAGE

Die Datei 999 wird aufgebaut, sobald die Anweisung zum Auflisten der Daten nach dem END-Statement mit END LIST DATA[52] angegeben wird. Die Datei NIMAGE nimmt alle im BASIC-Programm enthaltenen IMAGE-Statements mit ihren IMAGE-Masken auf.
Bei IBM-Maschinen wird die END-OF-FILE Abfrage in die entsprechende READ-Anweisung einbezogen, indem gilt:

READ(IRC,⟨Formatstatementnummer⟩, END=⟨Statementnummer⟩) ⟨Liste⟩

51 Vgl. FOSBIC-Programmliste im Anhang
52 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 332.45

331.3 FORTRAN-Fileanweisungen

Vom FOSBIC-Compilersystem werden alle benutzereigenen Dateien binär angelegt. Die Standarddateien 999 und NIMAGE werden dagegen nach einem intern festgelegten Format im BCD-Code erstellt. Als FORTRAN-Anweisungen zur Fileverarbeitung werden verwendet:

READ (⟨DSI⟩) ⟨Liste⟩
WRITE(⟨DSI⟩) ⟨Liste⟩
READ (⟨DSI⟩, ⟨Formatstatementnummer⟩) ⟨Liste⟩

WRITE (⟨DSI⟩, ⟨Formatstatementnummer⟩) ⟨Liste⟩
REWIND ⟨DSI⟩

Diese FORTRAN-Anweisungen bedingen einen physikalischen Fileaufbau der nur einen sequentiellen Zugriff erlaubt. Der mit BASIC mögliche Aufbau von index-sequentiell organisierten Files wird intern simuliert. Auf ein bei einzelnen FORTRAN-Compilern mögliches FIND-Statement wurde aus Kompatibilitätsgründen verzichtet[53].
Die vom Benutzer anlegbaren Dateien werden intern beginnend mit dem DSI = 1000 aufsteigend durchnumeriert, wobei für das Computersystem CD 3300 keine Dateibeschreibung mittels externer Steuerkarten vorgegeben werden muss, solange es sich um temporär angelegte Scratchfiles handelt. Bei permanenten Dateien muss die Dateibeschreibung durch externe maschinenorientierte Systemsteuerkarten angegeben werden, wobei das DSI mit der intern vorgenommenen Zuordnung einer Ganzzahl zwischen $1000 \leqslant DSI \quad 1024$ übereinstimmen muss. Normalerweise geschieht die DSI-Zuordnung in der Reihenfolge der Dateieröffnung. Ausnahmen entstehen, sobald in einem Programm Dateien zeitweise geschlossen und wieder geöffnet werden[54]. In diesem Fall ist es möglich, dass zwei oder mehrere Files mit unterschiedlichen Dateinamen dasselbe DSI erhalten, da das DSI einer geschlossenen Datei intern wieder für neu geöffnete Files verwendet wird. Mittels des PRINT ALL-Statements kann sich der Benutzer darüber informieren, welche Dateinamen- und DSI-Zuordnung besteht[55].

331.4 Zeichencode der FORTRAN-Anweisungen

Die FORTRAN-Anweisungen des FOSBIC-Compilersystems sind in dem Zeichencode nach Tabelle 5 geschrieben.

53 Bei IBM-Maschinen existiert ein FORTRAN FIND-Statement. Vgl. Clarence B. Germain. Das Programmier-Handbuch der IBM/360. Ein Lehr- und Arbeitsbuch. 2. Aufl., München 1970, S. 459.
54 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 332.12
55 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 333.12

Tabelle 5. FORTRAN-Zeichencode des FOSBIC-Compiler-Systems

Drucker-zeichen	Intern Oktal	Kartenlocher Lochung	Kartenlocher Schrift
0	00	0	0
1	01	1	1
2	02	2	2
3	03	3	3
4	04	4	4
5	05	5	5
6	06	6	6
7	07	7	7
8	10	8	8
9	11	9	9
:	12	8,2	:
≠	13	8,3	=
=	14	8,4	≠
+	20	12	+
A	21	12,1	A
B	22	12,2	B
C	23	12,3	C
D	24	12,4	D
E	25	12,5	E
F	26	12,6	F
G	27	12,7	G
H	30	12,8	H
I	31	12,9	I
.	33	12,8,3	.
)	34	12,8,4	)
−	40	11	−
J	41	11,1	J
K	42	11,2	K
L	43	11,3	L
M	44	11,4	M
N	45	11,5	N
O	46	11,6	O
P	47	11,7	P
Q	50	11,8	Q
R	51	11,9	R
*	54	11,8,4	*

Tabelle 5. FORTRAN-Zeichencode des FOSBIC-Compiler-Systems (Fortsetzung)

Drucker-zeichen	Intern Oktal	Kartenlocher Lochung	Schrift
	60	keine	blank
/	61	0,1	/
S	62	0,2	S
T	63	0,3	T
U	64	0,4	U
V	65	0,5	V
W	66	0,6	W
X	67	0,7	X
Y	70	0,8	Y
Z	71	0,9	Z
,	73	0,8,3	,
(	74	0,8,4	(

332 BASIC-Implementation

Das FOSBIC-Compilersystem erlaubt die Verarbeitung von Elementar-, Matrizen- und Fileanweisungen, so dass es bereits wesentliche Teile des von der NBC Task Group[56] vorgesehenen BASIC-Spracherweiterungen enthält.
Die modulare Aufbaustruktur ermöglicht auch die Implementation von Teilstrukturen mit reduziertem BASIC-Sprachumfang. Leicht eliminierbar sind die Module zur Matrizen- und Fileverarbeitung, da für sie die Systemkopplungen durch wenige Schnittstellen eindeutig festgelegt sind. Ähnliche Implementationsflexibilität wird nur vom Dartmouth BASIC-6-System erreicht[57].

332.1 BASIC-Sprachinstallation

Das FOSBIC-Compilersystem lässt folgende Varianten zur BASIC-Sprachinstallation zu:
- Elementar–, Matrizen- und Fileanweisungen
- Elementar- und Fileanweisungen
- Elementar- und Matrizenanweisungen

56 Vgl. Weber und Türschmann, BASIC, Bd. 2, Tabelle 4.
57 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 111

– Elementaranweisungen und
– Teilstrukturen.

Ausserdem besteht die Möglichkeit, die Prioritätsregeln zur Verarbeitung der monadischen Operatoren auf zwei unterschiedliche Arten einzustellen[58].

Das Entfernen der Matrizenanweisungen im Compilierungssystem geschieht durch Elimination der Statements 187 und 188

```
CALL MATTRA
GOTO 50
```

in MAIN. Hierzu sind die entsprechenden Karten aus dem Programmdeck herauszunehmen oder durch ein C in der ersten Spalte unwirksam zu machen.

Soll auf die BASIC-Fileanweisungen verzichtet werden, so müssen die Statements 182 und 183

```
CALL ZFILE(NZ)
GOTO 50
```

aus MAIN und Statement 179 aus ZINITL entfernt oder unwirksam gemacht werden. Entsprechendes gilt auch für die Statements 53 und 54 im Programm MATTRA.

Das Unterdrücken einzelner BASIC-Anweisungen ist ebenfalls möglich, sollte aber nur bei genauer Kenntnis des FOSBIC-Compilersystems vorgenommen werden.

Die vorerwähnten Änderungen können jederzeit rückgängig gemacht werden, solange die durch obige FORTRAN-Anweisungen angesprochenen Programme nicht aus dem Gesamtsystem entfernt werden.

Durch die Eingriffe im Compilierungssystem erübrigt sich normalerweise die Elimination der entsprechenden Programme oder Programmabschnitte aus dem Ausführungssystem. Geschieht dies trotzdem, um eventuell Kernspeicherplatz zu sparen, so ist sicherzustellen, dass für das verbleibende Teilsystem die Aufbaustruktur ungestört bleibt[59]. Für FOSBIC-Compiler ohne Matrizenanweisungen ist im Programm ZEXEC das Statement 117

```
CALL ZEXMAT(IOP)
```

durch

```
GOTO 909
```

zu ersetzen. Bei Verzicht auf Fileanweisungen ist das Statement 79

```
CALL ZEXFIL(NPR)
```

58 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 324.1
59 Vgl. Abs. 322 und 324.

unwirksam zu machen und für Statement 124

```
CALL ZEXFIL(IOP)
```

analog zu oben das FORTRAN-Statement

```
GOTO 909
```

einzufügen. Die den Aufrufen direkt nachfolgenden Anweisungen werden damit wirkungslos.

Tabelle 6. Prioritätsregeln zur Verarbeitung von Operatoren

	Prioritätsregel 1	Prioritätsregel 2
Funktionen	1	1
* * | ↑ | ^	2	3
* | /	3	4
monad. Operator	4	2
+ | −	5	5

Normalerweise werden die monadischen und dyadischen Operatoren in der Rangfolge nach Prioritätsregel 1 in Tabelle 6 verarbeitet[60]. Der FOSBIC-Compiler lässt auch die Verarbeitung der Operatoren nach Prioritätsregel 2 zu, falls im Programm ZTRANX die Statements

```
IF((ITRAN(I).EQ.-3).OR.(ITRAN(I).EQ.-4).OR.(ITRAN(I).EQ.-6)) IPR=2
IF(ITRAN(I).EQ.-5) IPR=3
```

und

```
971   IF(IHOLD(LOC).EQ.-5) GOTO 972
```

durch die Anweisungen

```
IF((ITRAN(I).EQ.-3).OR.(ITRAN(I).EQ.-4)) IPR=2
IF((ITRAN(I).EQ.-5).OR.(ITRAN(I).EQ.-6)) IPR=3
```

und

```
971   IF((IHOLD(LOC.LE.-5).AND.(IPR.LE.3)) GOTO 972
```

ersetzt werden[61].

60 Vgl. Weber und Türschmann, BASIC, Bd. 1 Abs. 324.1
61 Siehe auch Text im Programm ZTRANX der FOSBIC-Programmliste.

332.2 BASIC-Parameterinstallation

Die BASIC-Parameterstruktur des FOSBIC-Compilersystems erlaubt es, die Programmiersprache BASIC völlig unabhängig vom verwendeten Computersystem aufzubauen. Es besteht ausserdem die Möglichkeit, mehrere unterschiedliche FOSBIC-Versionen auf derselben Maschine unabhängig voneinander zu installieren. Alle hierzu notwendigen Änderungen sind im Programm ZINITL zusammengefasst.

332.21 BASIC-Grundsymbole

Die durch das FOSBIC-Compilersystem dargestellte Programmiersprache BASIC verfügt über 56 Grundsymbole

26 Buchstaben
10 Ziffern
20 Sonderzeichen

wobei 8 Sonderzeichen nur als Steuerzeichen innerhalb von BASIC-Anweisungen verstanden werden. Sobald diese Zeichen in einem String stehen, werden sie durch ein „blank" ersetzt[62].
Alle Grundsymbole sind über FORTRAN-DATA Anweisungen anzugeben, wobei der Zeichencode der BASIC-Grundsymbole beliebig – abweichend vom FORTRAN-Zeichencode – gewählt werden kann. Ausserdem kann vom Benutzer selbst der Zeichencode der BASIC-Grundsymbole eingegeben werden. Diese Codestruktur gilt dann für alle Folgejobs solange keine Neueingabe oder Neucompilierung des gesamten FOSBIC-Compilersystems erfolgt. Die Reihenfolge der nach der Stern-Karte anzugebenden Zeichenkarten zeigt das folgende Beispiel.

```
* BUILD---
ABCDEFGHIJKLMNOPQRSTUVWXYZ                    1.KARTE  BUCHSTABEN
0123456789                                    2.KARTE  ZIFFERN
* ,.=)(+#$-/?<>↓%↑v↑                          3.KARTE  SONDERZEICHEN
10 DIM A(10,5)
20 READ X,Y,Z
  ...
  ...
```

Wird bei der FOSBIC-Implementation im Programm ZINITL das Statement 79

NSTZEI = 1

nicht entfernt, so bleibt für den BASIC-Benutzer die freie Eingabe der BASIC-Grundsymbole gesperrt. Die Tabellen 7 und 8 enthalten die für das FOSBIC-Compilersystem 56 gültigen BASIC-Grundsymbole, wovon 48 Symbole innerhalb von Texten mittels eines Codesystems verschlüsselt werden können. (ALPH von 1 bis 48).

62 Es handelt sich um die Zeichen 13 bis 20 des Vektors CHARTM im Programm ZINITL; vgl. Tabelle 8.

Für die mnemotechnischen Namen von Variablen unterteilt sich die Codetabelle in den Vektor ALPH von 1 bis 26 (Buchstaben) und in den Vektor DIGIT (Ziffern).
Die Übernahme aller 56 BASIC-Grundsymbole in die Textverschlüsselung ist möglich, erhöht jedoch die maximal notwendige Bitzahl je Wort bzw. verringert den Umfang des in einem Wort speicherbaren Textes[63]. Alle Sonderzeichen tragen einen mnemotechnischen FORTRAN-Kurznamen, der in allen COMMON-Blöcken enthalten sein muss.

Tabelle 7. Codesystem der BASIC-Grundsymbole – Buchstaben und Ziffern

BASIC-Grund-symbol	im COMMON-Block gespeichert in	in ZINITL gespeichert in	Codezahl in Strings	Codezahl in Variablen-namen
A	ALPH (1)	ATEMP (1)	1	1
B	ALPH (2)	ATEMP (2)	2	2
C	ALPH (3)	ATEMP (3)	3	3
D	ALPH (4)	ATEMP (4)	4	4
E	ALPH (5)	ATEMP (5)	5	5
F	ALPH (6)	ATEMP (6)	6	6
G	ALPH (7)	ATEMP (7)	7	7
H	ALPH (8)	ATEMP (8)	8	8
I	ALPH (9)	ATEMP (9)	9	9
J	ALPH (10)	ATEMP (10)	10	10
K	ALPH (11)	ATEMP (11)	11	11
L	ALPH (12)	ATEMP (12)	12	12
M	ALPH (13)	ATEMP (13)	13	13
N	ALPH (14)	ATEMP (14)	14	14
O	ALPH (15)	ATEMP (15)	15	15
P	ALPH (16)	ATEMP (16)	16	16
Q	ALPH (17)	ATEMP (17)	17	17
R	ALPH (18)	ATEMP (18)	18	18
S	ALPH (19)	ATEMP (19)	19	19
T	ALPH (20)	ATEMP (20)	20	20
U	ALPH (21)	ATEMP (21)	21	21
V	ALPH (22)	ATEMP (22)	22	22
W	ALPH (23)	ATEMP (23)	23	23
X	ALPH (24)	ATEMP (24)	24	24

63 Vgl. Abs. 332.22.

Tabelle 7. Codesystem der BASIC-Grundsymbole – Buchstaben und Ziffern (Fortsetzung)

BASIC-Grundsymbol	im COMMON-Block gespeichert in	in ZINITL gespeichert in	Codezahl in Strings	Codezahl in Variablennamen
Y	ALPH (25)	ATEMP (25)	25	25
Z	ALPH (26)	ATEMP (26)	26	26
O	ALPH (27)	DIGIT (1)	27	1
1	ALPH (28)	DIGIT (2)	28	2
2	ALPH (29)	DIGIT (3)	29	3
3	ALPH (30)	DIGIT (4)	30	4
4	ALPH (31)	DIGIT (5)	31	5
5	ALPH (32)	DIGIT (6)	32	6
6	ALPH (33)	DIGIT (7)	33	7
7	ALPH (34)	DIGIT (8)	34	8
8	ALPH (35)	DIGIT (9)	35	9
9	ALPH (36)	DIGIT (10)	36	10

Tabelle 8. Codesystem der BASIC-Grundsymbole – Sonderzeichen

BASIC-Grundsymbol	im COMMON-Block gespeichert in	in ZINITL gespeichert in	Codezahl in Strings	mnemotechnischer Kurzname in FORTRAN
*	ALPH (37)	CHARTM (1)	37	ASTRSK
blank	ALPH (38)	CHARTM (2)	38	BLANK
,	ALPH (39)	CAHRTM (3)	39	COMMA
.	ALPH (40)	CHARTM (4)	40	DECMAL
=	ALPH (41)	CHARTM (5)	41	EQUALS
)	ALPH (42)	CHARTM (6)	42	PARRT
(	ALPH (43)	CHARTM (7)	43	PARLFT
+	ALPH (44)	CHARTM (8)	44	PLUS
'	ALPH (45)	CHARTM (9)	45	QUOTE
$	ALPH (46)	CHARTM (10)	46	DOLSGN
−	ALPH (47)	CHARTM (11)	47	CMINUS
/	ALPH (48)	CHARTM (12)	48	SLASH

Tabelle 8. Codesystem der BASIC-Grundsymbole – Sonderzeichen

BASIC-Grund-symbol	im COMMON-Block gespeichert in	in ZINITL gespeichert in	Codezahl in Strings	mnemotech-nischer Kurz-name in FORTRAN
;	spezielle	CHARTM (13)		PUCO
<	BASIC-	CHARTM (14)		VLESS
>	Grundsymbole,	CHARTM (15)		VGREAT
↓	die nicht in	CHARTM (16)		DQUOTE
%	Strings ver-	CHARTM (17)		DOPU
↑	wendet werden	CHARTM (18)		EXSIGN
V	dürfen.	CHARTM (19)		XNULL
:		CHARTM (20)		DDOPU

332.22 BASIC-Hardwareparameter

Das FOSBIC-Compilersystem gestattet es, die BASIC-Hardwareparameter an alle notwendigen Maschinenkonstanten des verwendeten Computersystems anzupassen. Bei der FOSBIC-Implementation sind hierzu die entsprechenden Werte des verwendeten Maschinensystems in das Programm ZINITL zu installieren. Bei der Installation ist darauf zu achten, dass einige der in Tabelle 9 angegebenen Parameter in Beziehung zu den in den COMMON//-Anweisungen dimensionierten Variablen stehen und eine gleichgrosse Dimensionierung verlangen. Die Grössenordnung, in der die Felder dimensioniert werden, hängt nur von dem verfügbaren Kernspeicherplatz ab. Um eine einwandfreie Verarbeitung von Zeichenketten mit einer Länge von fünf Zeichen und einem Zeichenumfang von 48 Zeichen zu gewährleisten, muss die verwendete Maschine minimal 28 Bit pro Wort und mindestens eine Ganzzahl vom Wert 254 803 967 zulassen. Bei kleineren Bitstrukturen pro Wort kann der Zeichenumfang verringert werden, wodurch die Textverarbeitung mit BASIC allerdings eingeschränkt wird.
Die Bitanzahl berechnet sich aus der Gleichung

$$\text{Bitanzahl} \geq \frac{\lg \left\{ \sum_{k=0}^{4} (\text{Zeichenumfang}-1) \ * \text{Zeichenumfang}^{k} \right\}}{\lg 2}$$

Der in einem String verarbeitbare Zeichenumfang wird in dem Programm ZALPH durch die Variable INTZEI festgelegt[64]. Auf die Begrenzung durch die BASIC-Hardwareparameter NCELLP, NCELLD und NSTEND wird der Benutzer durch das Compilierungssystem hingewiesen, indem nach jeder fehlerfreien Compilierung die Mitteilung

```
********** EVERYTHING SEEMS OK -- LET'S GO AHEAD

          PERCENT OF AVAILABLE STORAGE USED              18.054
          PERCENT OF AVAILABLE DATA STORAGE USED          3.939
          PERCENT OF AVAILABLE NUMBERED STATEMENTS USED   5.882
```

erscheint, welche die Ausnutzung der Vektoren DATA resp. IPROG, DATAN und LISTST resp. ISTLST durch das vorausgehende BASIC-Programm angibt.

Tabelle 9. BASIC-Hardwareparameter

FORTRAN-Name	Bedeutung	Wert bei CD 3300	steht in Beziehung zu
NCELLP	Speicherplatzumfang in Worten für Programm, Konstanten, Felder	3700	DATA und IPROG
NCELLD	Speicherplatzumfang in Worten für die mittels dem DATA-Statement eingebbaren Daten	340	DATAN
IRC	Kanalnummer (DSI) des Standard Input System (Lochkartenleser)	60	
IWC	Kanalnummer (DSI) des Standard Output System (Schnelldrucker)	61	
IEXPO	Maximaler Exponent der Gleitkommazahlen	99 (307)	
INTMAX	Maximale Ganzzahl bei 48 Bit pro Wort (Bit-Struktur beachten)	140737488355327	
INTNUM	Maximaler Exponent der Ganzzahlen	14	
NIMAGE	Kanalnummer (DSI) des sequentiellen Files zur Aufnahme der Imagemasken	100	
MAXIMA	Maximale Anzahl der IMAGE-Statements	300	
IZONE	Anzahl der Druckstellen pro Druckzone der Elementaranweisung	15	
IIMAGE	Maximale Anzahl der Druckstellen pro Zeile mit PRINT USING	135	CARP
SMALL	kleinster absoluter Wert, der mit PRINT darstellbar ist	1.E-38	

64 Vgl. FOSBIC-Programmliste, Programm ZALPH.

Tabelle 9. BASIC-Hardwareparameter (Fortsetzung)

FORTRAN-Name	Bedeutung	Wert bei CD 3300	steht in Beziehung zu
ISTMAX	Maximale Statementnummer	9999	
NIFMAX	Maximale Verschachtelungstiefe von FOR/NEXT-Schleifen	20	IFOR
NIRMAX	Maximale Verschachtelungstiefe von Unterprogrammen	20	IRET
IPEND	Anzahl der Druckzonen je Standarddruckzeile	eingebbar 1–8	
NSTEND	Maximale Anzahl der numerierten Statements pro BASIC-Programm	340	LISTST/ISTLST
MAXFIL	Maximale Anzahl der anlegbaren Files	25	
MAXSAT	Maximale Anzahl der Sätze für ein File = höchst zulässiges DSI	1024	

333 BASIC-Steuerkarten

Die BASIC-Steuerkarten des FOSBIC-Compilersystems bestehen aus einer dem BASIC-Programm vorausgehenden und einer nachfolgenden Stern(*)-Karte, wobei der Stern in der ersten Spalte stehen muss. In der zweiten Spalte der dem BASIC-Programm vorausgehenden BASIC-Steuerkarte kann die gewünschte Anzahl der Druckzonen vom Benutzer angegeben werden[65]. Ein Wert von Null wird automatisch auf 1 und ein Wert grösser als 8 auf 8 gesetzt.
Mit den Angaben
- NO-HEADLINE
- BUILD---
- LIST-ERROR

beginnend in der dritten Spalte der vorausgehenden Steuerkarte können
- die Feldkennungen bei der Matrizenausgabe unterdrückt[66]
- die Basic-Grundsymbole benutzergesteuert eingegeben und
- alle BASIC-Fehlermeldungen aufgelistet

werden.

65 Vgl. Weber und Türschmann, BASIC, Bd. 1 Abs. 332.21
66 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 331.311

Alle Fehlermeldungen des FOSBIC-Compilersystems enthalten die Textzeile

SEE BASIC TEXTBOOK ⟨Hinweis⟩

wobei durch den Hinweis auf Band und Abschnitt eines angebbaren Textbuches verwiesen wird, der vom Benutzer zur Fehlerbehebung gelesen werden sollte. Die Version 6/76–04 des FOSBIC-Compilersystems ist auf das BASIC-Textbuch

Karl Weber und Carl Wolfram Türschmann,
BASIC, UTB 588 und 589, Bern und Stuttgart 1977

und den vorliegenden FOSBIC-Band eingestellt[67].
Die Angaben zum Textbuch – Autoren, Titel, Erscheinungsort und Erscheinungsjahr – enthält die erste Seite jedes BASIC-Programms, auf der ausserdem Informationen und eine Programmkennung, die mit dem Text der vorausgehenden Stern-Karte übereinstimmt, angegeben werden.

```
INFORMATIONEN DURCH:

PROF.DR.OEC.PUBL.K.WEBER,M.S.
DIPL.-ING.,DIPL.-OEC.C.W.TUERSCHMANN
PROFESSUR FUER BETRIEBSWIRTSCHAFTSLEHRE V
LICHER STRASSE 74
D-6300 GIESSEN
FEDERAL REPUBLIC OF GERMANY


********************************* PROGRAMMKENNUNG *********************************

*8NO-HEADLINE

*****************************************************************************

BASIC TEXTBOOK = WEBER, TUERSCHMANN. BASIC LEHR- UND HANDBUCH. BERN 1977. BAND 1 = A
****************                                                    BAND 2 = B
               WEBER, TUERSCHMANN. FOSBIC. BERN 1977.                       = C
```

Bei Textbuchänderung müssen die Formatanweisungen der Statements 53 bis 59 im Programm MAIN entsprechend umgestaltet werden. Simultan dazu sind alle Buchhinweise in den Programmen

STRING
COMERR und
EXERR

67 Vgl. auch: Horning, What the Compiler should tell the User, p. 543–548.

auf das angegebene Textbuch abzustimmen. Je Hinweis stehen maximal 8 Zeichen zur Verfügung, wobei dies jedoch von der Wortstruktur der verwendeten Maschine abhängt.

335 BASIC-Fehlermeldungen

Compilierungs- und Ausführungssystem des FOSBIC-Compilersystems enthalten zwei voneinander unabhängige Unterprogramme (COMERR und EXERR) mit der gesamten BASIC-Fehlerdiagnostik.
Die Auswahl der Fehlermeldung erfolgt durch den Parameter NERROR. Über zusätzliche Parameter können für den Fehlerausdruck benötigte Variablenwerte von dem rufenden Programm an das jeweilige Diagnoseprogramm übergeben werden.
Bei der Compilierung können drei Arten von Fehlermeldungen auftreten:
- ERRORs (N=0)
- WARNINGs (NN=1) und
- REASONs (NN=2),

wobei die Auswahl von dem rufenden Programm mittels des Parameters NN getroffen wird. Die Anzahl von 55 ERRORs, 9 WARNINGs und 12 REASONs kann beliebig erweitert werden[68].
ERROR-Meldungen führen stets zu einem Abbruch des BASIC-Programms, ohne dass die Ausführungsphase begonnen wird. Bei WARNING-Meldungen wird vom Compilierungssystem teilweise selbständig eine Korrektur des Programmfehlers durchgeführt[69]. REASON-Meldungen treten nur im Zusammenhang mit einer ERROR-Meldung bei fehlerhaften arithmetischen Ausdrücken auf[70].
Wird nach den DATA-Statements im Programm COMERR das FORTRAN-Statement

```
NN = NN - 3
```

eingefügt, so unterbleibt der Ausdruck des Fehlertextes und es erfolgt nur ein Hinweis auf die jeweilige ERROR-, WARNING- oder REASON-Nummer sowie der BASIC Textbuchhinweis.
Der speicherplatzaufwendige Text mit den inaktiven WRITE-Anweisungen kann dann aus dem Programm entfernt werden.
Um einzelne Fehlertexte zu unterdrücken, muss der NN-Parameter in dem rufenden Programm um 3 verringert werden. Eine Veränderung im Programm COMERR unterbleibt.
Das Programm EXERR enthält die 51 Fehlermeldungen des Ausführungssystems. Die Textunterdrückung wird hier bewirkt, indem das Statement 69

68 Vgl. FOSBIC-Programmliste im Anhang. Die WARNING-Meldungen 8 und 9 enthält das Programm STRING.
69 Vgl. auch: Weber und Türschmann. BASIC, Bd. 1, Abs. 332.43. – Horning, What the Compiler should tell the User, p. 533, 537–539 and 544. – Gries, Compiler Construction for Digital Computers, p. 315–326.
70 Vgl. Weber und Türschmann. BASIC, Bd. 1, Abs. 324.1.

NN=1

durch das Statement

NN=O

ersetzt wird. Der Hinweis auf die Fehlernummer, der BASIC-Textbuchhinweis und der Hinweis auf die Zeile, in der der Fehler gefunden wurde, bleiben erhalten.
Die folgenden Tabellen 10, 11, 12 und 13 enthalten in aufsteigender Nummernfolge alle BASIC-Fehlermeldungen des FOSBIC-Compilersystems mit den dazugehörigen BASIC-Textbuchhinweisen[71].

Tabelle 10. ERROR-Meldungen

```
        PROGRAM AND DATA EXCEED AVAILABLE STORAGE AT THIS POINT
SEE BASIC TEXTBOOK  A  331.2

        SORRY - I ACCEPT ONLY LINE NUMBERS BETWEEN 1 AND  9999
SEE BASIC TEXTBOOK  A  326.2

        ILLEGAL RESTORE STATEMENT
SEE BASIC TEXTBOOK  A 332.15

        I CANNOT FIND AN EQUAL SIGN IN THE LET STATEMENT ABOVE
SEE BASIC TEXTBOOK  A  332.3

        THE EQUAL SIGN ABOVE IS IN A FUNNY PLACE
SEE BASIC TEXTBOOK  A  332.3

        ITEM TO THE LEFT OF THE EQUAL SIGN MUST BE A VARIABLE
SEE BASIC TEXTBOOK  A  332.3

        THE QUOTATION MARKS ABOVE ARE NOT PAIRED
SEE BASIC TEXTBOOK  A  322.2

        SORRY - I CAN ONLY GO TO LINES WITH NUMBERS BETWEEN 1 AND  9999
SEE BASIC TEXTBOOK  A 332.41

        THE IF STATEMENT ABOVE IS HOPELESS
SEE BASIC TEXTBOOK  A 332.42

        I CANNOT ACCEPT AN IF STATEMENT WITHOUT A COMPARISON
SEE BASIC TEXTBOOK  A 332.42

        I CANNOT FIND THEN OR GOTO
SEE BASIC TEXTBOOK  A 332.42

        THEN OR GOTO MUST BE FOLLOWED BY A LINE NUMBER
SEE BASIC TEXTBOOK  A 332.42

        THE FOR STATEMENT ABOVE IS HOPELESS
SEE BASIC TEXTBOOK  A 332.43

        I CANNOT FIND AN EQUAL SIGN IN THE FOR STATEMENT ABOVE
SEE BASIC TEXTBOOK  A 332.43

        THE VARIABLE IN A FOR STATEMENT MUST BE UNSUBSCRIPTED
SEE BASIC TEXTBOOK  A 332.43
```

71 Vgl. Abs. 324.3 und 334.

```
        I CANNOT FIND A TO
SEE BASIC TEXTBOOK  A 332.43

        EITHER YOU FORGOT A FOR STATEMENT OR I REJECTED IT
SEE BASIC TEXTBOOK  A 332.43

        DIM STATEMENT INCORRECTLY WRITTEN
SEE BASIC TEXTBOOK  A 331.21

        WRONG CHARACTER WITHIN DATA
SEE BASIC TEXTBOOK  A 332.11

        YOU GAVE ME    0 MORE DATA IN DATA STATEMENTS THAN I CAN HANDLE
SEE BASIC TEXTBOOK  A 332.11

        YOU MAY UNDERSTAND THE COMMAND ABOVE BUT I DO NOT
SEE BASIC TEXTBOOK  A  326.1

        STATEMENT NUMBER    0 APPEARS MORE THAN ONCE
SEE BASIC TEXTBOOK  A  326.2

        THERE ARE     0 FOR STATEMENTS WITHOUT NEXT STATEMENTS
SEE BASIC TEXTBOOK  A 332.43

        COMPILATION STOPPED AT THIS POINT -- I CAN ONLY HANDLE   340 NUMBERED STATEMENTS
SEE BASIC TEXTBOOK  A  326.2

        AN IMAGE STATEMENT MUST HAVE A STATEMENT NUMBER
SEE BASIC TEXTBOOK  A 332.23

        I CAN ONLY HANDLE  300 IMAGE STATEMENTS
SEE BASIC TEXTBOOK  A 332.23

        THERE IS NO REFERENCE TO AN IMAGE STATEMENT WITHIN THE PRINT USING
SEE BASIC TEXTBOOK  A 332.23

        I DO NOT LIKE THE EXPRESSION SHOWN BELOW (REASONS FOLLOW)
SEE BASIC TEXTBOOK  A  324.1

        THERE IS SOMETHING I DO NOT LIKE ABOUT THE EXPRESSION SHOWN BELOW
SEE BASIC TEXTBOOK  A  324.1

        I CAN ONLY HANDLE NUMBERS BETWEEN 10** 99 AND 10**(- 99)
SEE BASIC TEXTBOOK  A 322.12

        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B    331

        I DO NOT UNDERSTAND THE FOLLOWING PART IN THE MAT STATEMENT ABOVE
0
SEE BASIC TEXTBOOK  B331.111

        I CANNOT FIND AN EQUAL SIGN IN THE MAT LET STATEMENT ABOVE
SEE BASIC TEXTBOOK  B 331.21

        ILLEGAL MAT LET STATEMENT
SEE BASIC TEXTBOOK  B331.211

        I DO NOT LIKE THE FOLLOWING CHARACTER IN THE MAT STATEMENT ABOVE *0*
SEE BASIC TEXTBOOK  B331.211

        VARIABLE *0* IS DIMENSIONED AS A VECTOR -- VARIABLE *0* IS DIMENSIONED AS A MATRIX
        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B331.211
```

```
        VARIABLE *0* IS NOT DIMENSIONED AS A N*N MATRIX
        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B331.211

        NO ACCEPTABLE MATRIX MULTIPLICATION
        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B331.211

        VARIABLE *0* IS NOT DIMENSIONED
        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B331.111

        VARIABLE *0* IS DIMENSIONED AS A VECTOR BUT YOU TREAT IT AS A MATRIX
        I CANNOT ACCEPT THE MAT STATEMENT ABOVE
SEE BASIC TEXTBOOK  B 331.21

        ERROR IN INPUT OR OUTPUT LIST -- UNCORRECT VARIABLE,SUBSCRIPT OR EXPRESSION
        THE PART I DO NOT LIKE IS
SEE BASIC TEXTBOOK  A323/324

        THE FILE STATEMENT ABOVE IS HOPELESS
SEE BASIC TEXTBOOK  B    332

        ILLEGAL FIELD NAME IN THE MAT STATEMENT ABOVE *0*
SEE BASIC TEXTBOOK  B 331.21

        FILE NAME ERROR -- THE QUOTATION MARKS ARE NOT PAIRED
SEE BASIC TEXTBOOK  B 331.21

        I AM VERY SORRY BUT I AM ONLY ALLOWED TO GIVE YOU   24 SENTENCES AND YOU ASK FOR    30
SEE BASIC TEXTBOOK  B 331.21

        THE FIRST FILE STATEMENT MUST BE A COMMON-FILE STATEMENT
SEE BASIC TEXTBOOK  B 332.13

        FILE SENTENCE ERROR
SEE BASIC TEXTBOOK  B 332.11

        THE FILE NAME ABOVE IS REDEFINED
SEE BASIC TEXTBOOK  B 332.11

        I AM VERY SORRY BUT I CAN ONLY ALLOW YOU TO DEFINE 25 FILES
SEE BASIC TEXTBOOK  B 332.11

        THE QUANTITY OF WORDS IN A INDEX SEQUENTIAL OPEN MUST BE GREATER THAN ZERO
SEE BASIC TEXTBOOK  B 332.11

        ERROR ON THE LEFT SIDE OF A MAT LET STATEMENT
        I DO NOT UNDERSTAND THE FOLLOWING PART IN THE MAT STATEMENT ABOVE

SEE BASIC TEXTBOOK  B 331.21

        DATA CAN ONLY BE READ INTO A VARIABLE
        THE PART I DO NOT LIKE IS
SEE BASIC TEXTBOOK  A 332.13

        MISSING PARENTHESIS IN MAT STATEMENT
        THE PART I DO NOT LIKE IS
SEE BASIC TEXTBOOK  B331.112

        DIMENSION NOT ALLOWED WITH EXPRESSION
        I DO NOT UNCERSTAND THE FOLLOWING PART IN THE MAT STATEMENT ABOVE

SEE BASIC TEXTBOOK  B331.211

        I CAN ONLY HANDLE  20 FOR STATEMENTS WITHOUT NEXT STATEMENTS
SEE BASIC TEXTBOOK  A 332.43
```

Tabelle 11. WARNING-Meldungen

```
        I EXPECTED YOU TO MENTION VARIABLE 00 -- I WILL ASSUME THAT YOU DID
SEE BASIC TEXTBOOK  A 332.43

        VARIABLE 0 REDIMENSIONED -- BUT I TAKE IT AS A  JOKE
SEE BASIC TEXTBOOK  A 331.21

        THERE ARE NO DATA IN THIS DATA STATEMENT
SEE BASIC TEXTBOOK  A 332.11

        YOU TREAT A NUMERIC VARIABLE AS AN ALPHANUMERIC ONE
SEE BASIC TEXTBOOK  A  323.1

        YOU TREAT AN ALPHANUMERIC VARIABLE AS A NUMERIC ONE
SEE BASIC TEXTBOOK  A  323.2

        I CANNOT ACCEPT A NUMBER FOR WORDS IN A COMMON-FILE  STATEMENT - I IGNORE IT
SEE BASIC TEXTBOOK  B 332.13

        MISSING COMMA OR SEMICOLON
SEE BASIC TEXTBOOK  A  332.2

        STRING CONSTANT HAS MORE THAN FIVE CHARACTERS -- I ONLY TAKE THE LEFT MOST FIVE
SEE BASIC TEXTBOOK  A  322.2

        UNKNOWN CHARACTER IN STRING CONSTANT *Z* I REPLACE IT BY A BLANK
SEE BASIC TEXTBOOK  A  322.2
```

Tabelle 12. REASON-Meldungen

```
A -- THIS IS AN ILLEGAL NAME

B -- NO RIGHT PARENTHESIS FOR THIS LEFT PARENTHESIS

C -- NO LEFT PARENTHESIS FOR THIS RIGHT PARENTHESIS

D -- THIS COMMA IS IN A FUNNY PLACE

E -- THIS PAIR OF OPERATORS IS ILLEGAL

F -- THERE IS SOMETHING FUNNY HERE

G -- A CONSTANT CANNOT BE FOLLOWED BY A LEFT PARENTHESIS

H -- THIS OPERATOR IS DANGLING

I -- A RIGHT PARENTHESIS CANNOT BE FOLLOWED BY A VARIABLE OR CONSTANT

J -- THESE VARIABLES ARE NEXT TO EACH OTHER

K -- THESE CONSTANTS ARE NEXT TO EACH OTHER

L -- A VARIABLE IS NEXT TO A CONSTANT
```

Tabelle 13. Fehlermeldungen des Ausführungssystems

```
PROGRAM STOPPED -- I CAN ONLY HANDLE  20 GOSUBS WITHOUT RETURNS
                   ERROR CODE=  1
              ***** SEE BASIC TEXTBOOK =  A 332.46  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- I CANNOT GO TO   12 BECAUSE IT IS NOT THERE
                   ERROR CODE=  2
              ***** SEE BASIC TEXTBOOK =  A 332.41  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- I CANNOT RETURN SINCE I DID NOT COME FROM A GOSUB
                   ERROR CODE=  3
              ***** SEE BASIC TEXTBOOK =  A 332.46  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- I CAN HANDLE ONLY NUMBERS BETWEEN 10** 99 AND 10**(- 99)
          END
                   ERROR CODE=  4
              ***** SEE BASIC TEXTBOOK =  A 322.12  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- WRONG CHARACTER IN DATA
          END
                   ERROR CODE=  5
              ***** SEE BASIC TEXTBOOK =  A 332.11  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- QUOTATION MARKS ARE NOT PAIRED IN THE FOLLOWING DATA-CARD
          END
                   ERROR CODE=  6
              ***** SEE BASIC TEXTBOOK =  A  322.2  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- YOU SET DATA POINTER ON POSITION NUMBER   -12 BUT I ONLY CAN DO IT UNTIL   330 OR ZERO
                   ERROR CODE=  7
              ***** SEE BASIC TEXTBOOK =  A 332.15  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- END OF DATA
                   ERROR CODE=  8
              ***** SEE BASIC TEXTBOOK =  A 332.13  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ERROR AT  3700   CODE=         -11
                   ERROR CODE=  9
              ***** SEE BASIC TEXTBOOK =  C  324.3  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ZERO STEP SIZE IN A FOR STATEMENT
                   ERROR CODE= 10
              ***** SEE BASIC TEXTBOOK =  A 332.43  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED --
                   MAT CODE ERROR AT  3700 CODE=         -11
                   ERROR CODE= 11
              ***** SEE BASIC TEXTBOOK =  C  324.3  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED --
                   FILE CODE ERROR AT  3700 CODE=         -11
                   ERROR CODE= 12
              ***** SEE BASIC TEXTBOOK =  C  324.3  *****
              ***** I WAS AT LINE NUMBER    0
```

```
PROGRAM STOPPED -- PERHAPS I AM CRAZY BUT I CANNOT FIND AN IMAGE STATEMENT WITH THE NUMBER    12
                   ERROR CODE= 13
              ***** SEE BASIC TEXTBOOK =  A 332.23  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THERE IS AN ATTEMPT TO DIVIDE BY ZERO
                   ERROR CODE= 14
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ILLEGAL EXPONENTIATION -- THE ARGUMENT IS NEGATIVE -0.39560357E+34 THE EXPONENT IS 10.12345000E+05
                   ERROR CODE= 15
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- EXPONENT OVERFLOW IN EXPONENTIATION  MAXIMUM = 99
                   ERROR CODE= 16
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THERE IS NOT ENOUGH SPACE FOR ITEM NUMBER        12 IN LIST OR TABLE A
                   ERROR CODE= 17
              ***** SEE BASIC TEXTBOOK =  A 331.21  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THE ARGUMENT OF LOG IS ZERO
                   ERROR CODE= 18
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THE ARGUMENT OF LOG IS NEGATIVE
                   ERROR CODE= 19
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THE ARGUMENT OF SQR IS NEGATIVE
                   ERROR CODE= 20
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- YOU WANT TO ALLOCATE MORE THAN   24 SENTENCES
                   ERROR CODE= 21
              ***** SEE BASIC TEXTBOOK =  B 332.11  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- THERE IS AN ATTEMPT TO OPEN AN ALREADY OPENED FILE
                   ERROR CODE= 22
              ***** SEE BASIC TEXTBOOK =  B 332.11  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- I CAN OPEN ONLY   25 FILES
                   ERROR CODE= 23
              ***** SEE BASIC TEXTBOOK =  B 332.11  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- INDEX-SEQUENTIAL FILE MUST BE OPENED BY A PRECEEDING OPEN STATEMENT
                   ERROR CODE= 24
              ***** SEE BASIC TEXTBOOK =  B 332.21  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- SENTENCE NUMBER    24 EXCEEDS MAXIMAL SENTENCE NUMBER    25
                   ERROR CODE= 25
              ***** SEE BASIC TEXTBOOK =  B 332.21  *****
              ***** I WAS AT LINE NUMBER    0
```

```
PROGRAM STOPPED -- YOU CANNOT GET DATA FROM AN UNOPENED FILE
                   ERROR CODE= 26
              ***** SEE BASIC TEXTBOOK =  B 332.32  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- YOU CANNOT RESET AN UNOPENED FILE
                   ERROR CODE= 27
              ***** SEE BASIC TEXTBOOK =  B 332.23  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- END OF FILE REACHED
                   ERROR CODE= 28
              ***** SEE BASIC TEXTBOOK =  B 332.32  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- YOU CANNOT CLOSE AN UNOPENED FILE
                   ERROR CODE= 29
              ***** SEE BASIC TEXTBOOK =  B 332.12  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ERROR IN MATRIX MULTIPLICATION
                   ERROR CODE= 30
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- NO CORRESPONDING SIZE OF THE SUBSCRIPTS FOR MATRIX TRANSPONATION
                   ERROR CODE= 31
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- COLUMN INDEX OF THE MATRIX *A* IS NEGATIVE OR ZERO   -12345.00
                   ERROR CODE= 32
              ***** SEE BASIC TEXTBOOK =  B331.312  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ROW INDEX OF THE MATRIX *A* IS NEGATIVE OR ZERO   -12345.00
                   ERROR CODE= 33
              ***** SEE BASIC TEXTBOOK =  B331.312  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ILLEGAL REDIMENSION OF THE MATRIX *A* ROWS=   24 COLUMNS=   25  MAXIMAL STORAGE ELEMENTS 12345.
                   ERROR CODE= 34
              ***** SEE BASIC TEXTBOOK =  B331.312  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- MATRIX *A* IS NOT DIMENSIONED N*N ROWS=   24 COLUMNS=   25
                   ERROR CODE= 35
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- ERROR IN MATRIX OPERATION -- ADDITION OR SUBTRACTION
                   ERROR CODE= 36
              ***** SEE BASIC TEXTBOOK =  B331.221  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- OUTPUT EXCEEDS  135 CHARACTERS AT IMAGE STATEMENT  300
                   ERROR CODE= 37
              ***** SEE BASIC TEXTBOOK =  A 332.23  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- EXPONENT OVERFLOW IN TANGENT  MAXIMUM = 99
                   ERROR CODE= 38
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0
```

```
PROGRAM STOPPED -- NO NEGATIVE COLUMN-NUMBER IN PRINT TAB ALLOWED=**********
                   ERROR CODE= 39
              ***** SEE BASIC TEXTBOOK =  A 332.22  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- EXPONENT OVERFLOW OR UNDERFLOW IN MULTIPLICATION 0.395603570E+34 *  0.123450000E+05  MAXIMUM = 99
                   ERROR CODE= 40
              ***** SEE BASIC TEXTBOOK =  A  324.1  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED --
THE VALUE OF THE EXPRESSION IN A COMPUTED GOTO OR GOSUB CANNOT BE USED TO SELECT A STATEMENT NUMBER - VALUE =       -24
                   ERROR CODE= 41
              ***** SEE BASIC TEXTBOOK =  A332.412  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- DIVISION ERROR IN MAT LET STATEMENT
                   THERE IS AN ATTEMPT TO DIVIDE BY ZERO
                   ERROR CODE= 42
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- EXPONENTIATION ERROR IN MAT LET STATEMENT
                   ILLEGAL EXPONENTIATION -- THE ARGUMENT IS NEGATIVE  0.39560357E+34 THE EXPONENT IS  0.12345000E+05
                   ERROR CODE= 43
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- EXPONENTIATION ERROR IN MAT LET STATEMENT
                   EXPONENT OVERFLOW IN EXPONENTIATION  MAXIMUM = 99
                   ERROR CODE= 44
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- MULTIPLICATION ERROR IN MAT LET STATEMENT
                   EXPONENT OVERFLOW OR UNDERFLOW IN MULTIPLICATION 0.395603570E+34 *  0.123450000E+05  MAXIMUM = 99
                   ERROR CODE= 45
              ***** SEE BASIC TEXTBOOK =  B331.211  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- SENTENCE  25 CANNOT BE RESET TO NEGATIVE INDEX-POSITION  24
                   ERROR CODE= 46
              ***** SEE BASIC TEXTBOOK =  B 332.23  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- SENTENCE NUMBER  24 IS LESS OR EQUAL TO ZERO
                   ERROR CODE= 47
              ***** SEE BASIC TEXTBOOK =  B 332.21  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- NO CORRESPONDING SIZE OF THE SUBSCRIPTS FOR MATRIX INVERSION
                   ERROR CODE= 48
              ***** SEE BASIC TEXTBOOK =  B331.221  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- DETERMINANT OF THE MATRIX *X* IS ZERO
                   ERROR CODE= 49
              ***** SEE BASIC TEXTBOOK =  B331.221  *****
              ***** I WAS AT LINE NUMBER    0

PROGRAM STOPPED -- I CAN ONLY INVERT A  24 *  24 MATRIX
                   ERROR CODE= 50
              ***** SEE BASIC TEXTBOOK =  B331.221  *****
              ***** I WAS AT LINE NUMBER    0
```

```
PROGRAM STOPPED -- MATRIX MULTIPLICATION OF THE FORM X=X*Y NOT ALLOWED - X MUST BE DIMENSIONED AS A ONE-ROW MATRIX
           ERROR CODE= 51
      ***** SEE BASIC TEXTBOOK =  8331.211  *****
      ***** I WAS AT LINE NUMBER    0
```

336 FOSBIC-Jobinstallation

Auf dem Computersystem CD 3300 des Hochschul-Rechenzentrums der Justus-Liebig-Universität Giessen wird das FOSBIC-Compilersystem gleichzeitig im Batchbetrieb und unter dem Teilnehmersystem MOTUS (Master Oriented Timesharing User System) im Dialogbetrieb verwendet.
Die FOSBIC-Installation im Batchbetrieb kann auf allen Anlagen mit FORTRAN IV-Compiler problemlos erfolgen. Die Installation im Timesharingbetrieb bedingt ein Betriebssystem, das sowohl die Jobeingabe als auch die Ergebnisausgabe über Terminals zulässt.

336.1 FOSBIC im Batchbetrieb

Für den Batchbetrieb sollte das FOSBIC-Compilersystem benutzerfreundlich – mit einem Minimum an maschinenorientierten Systemsteuerkarten – installiert werden, wozu sich erfahrungsgemäss am sinnvollsten die Anlage eines Binärdecks auf einer Magnetplatte oder einem Magnetband anbietet.
Der Grundaufbau eines BASIC-Jobs einschliesslich der beiden BASIC-Steuerkarten gestaltet sich wie folgt:

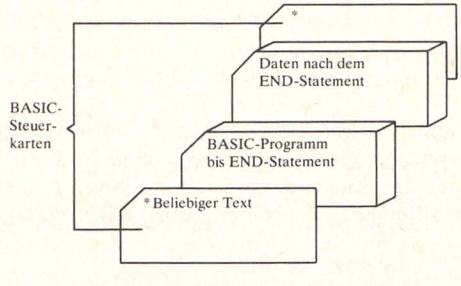


Abbildung 19. BASIC-Jobaufbau

Ein FOSBIC-Job enthält neben dem BASIC-Job die für das Maschinensystem notwendigen Systemsteuerkarten.

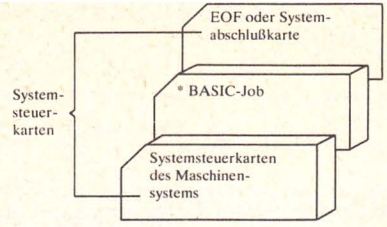


Abbildung 20. FOSBIC-Jobaufbau

Da mit einem Satz von Systemsteuerkarten mehrere BASIC-Jobs bearbeitet werden können, ist der Aufbau eines FOSBIC-Jobs in Stapelverarbeitung ähnlich einfach.

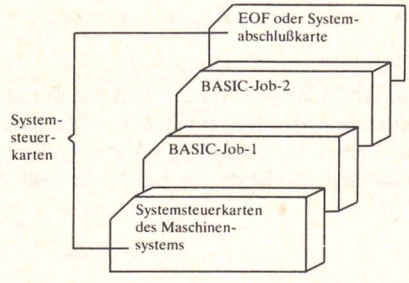


Abbildung 21. FOSBIC-Jobaufbau bei Stapelverarbeitung

Die obigen Abbildungen 19, 20 und 21 zeigen, dass der BASIC-Job frei von jeder Art von maschinenorientierten Steuerkarten ist. Stellt das zuständige Rechenzentrum dem Benutzer die Systemsteuerkarten vorgelocht zur Verfügung, so entfällt für den Anwender jede Einstellung auf ein besonderes Maschinensystem. Das folgende Beispiel zeigt einen für die CD 3300 typischen FOSBIC-Jobaufbau für den Batchbetrieb.

```
$JOB,150200,507BASIC,4,5000,,,,
$SCHED,CLASS=C,CORE=64,SCR=30,DP=1(101)
$*DEF(O,,PUM,150200,BASIC)
$BASI,PUM
*  1.KARTE DES 1.BASIC-JOBS
   ...
    ...
     1.BASIC-PROGRAMM
    ...
   ...
```

```
*  LETZTE KARTE DES 1.BASIC-JOBS
*  1.KARTE DES 2.BASIC-JOBS
   ...
    ...
     2.BASIC-PROGRAMM
    ...
   ...
*  LETZTE KARTE DES 2.BASIC-JOBS
   ...
    ...
     WEITERE BASIC-JOBS
    ...
   ...
EOF-KARTE 7/8 7/8 LOCHUNG
```

336.2 FOSBIC im Dialogbetrieb

Das von Dieter Wolff für die EDV-Anlage CD 3300 entwickelte Teilnehmersystem MOTUS[72] gestattet dem Benutzer vollständige FOSBIC-Jobs direkt über ein Terminal einzugeben, den Job zur Berechnung in die Warteschlange der Anlage einzureihen und sich die Programmergebnisse auf dem Terminal anzusehen.
Im Gegensatz zu der beim Timesharingbetrieb mit BASIC üblichen zeilenweisen Compilierung jedes eingegebenen BASIC-Statements, muss unter dem Teilnehmersystem MOTUS ein vollständiger FOSBIC-Job auf einem temporären oder permanenten File erstellt werden, bevor die Bearbeitung der im FOSBIC-Job enthaltenen BASIC-Jobs durch den FOSBIC-Compiler erfolgen kann. Der Benutzer wählt dabei, ob der Joboutput über den zentralen Schnelldrucker oder auf ein von ihm angelegtes File erfolgt. Im zweiten Fall wird es ihm ermöglicht, den Fileinhalt und damit die Ergebnisse seiner BASIC-Programme auf dem Terminal zu betrachten. Während einer Sitzung am Terminal kann somit der Benutzer einen Dialog mit dem FOSBIC-Compilersystem führen, wobei ihm ausserdem durch MOTUS die Möglichkeit gegeben wird, den Dialog auf einer Druckliste festzuhalten.
Bei der Programmausgabe über den zentralen Schnelldrucker gestaltet sich die Befehlsfolge mit MOTUS nach folgendem Beispiel:

```
MOTUS-ANWEISUNG                BEMERKUNG
---------------                ---------
LI,150200¬                     SITZUNGSBEGINN DES BENUTZERS 150200
A,,,PROGRAMMEINGABE¬           FILEDEFINITION ZUR PROGRAMMEINGABE
O,,INP,,PROGRAMMEINGABE,,,O¬   OEFFNEN DER PROGRAMMEINGABE
BUILD,INP¬                     ANFANGSBEFEHL ZUM AUFBAU EINES FOSBIC-JOBS
INP,1                          DER FOSBIC-JOB BEGINNT BEI RECORD 1
$JOB,150200,/507TEST,1,500,,,,          ANFANG DES FOSBIC-JOBS
$SCHED,CLASS=C,CORE=64,SCR=40,DP=1(101)
$*DEF(O,,PUM,150200,BASIC)
$BASI,PUM
*3  ANFANG DES BASIC-JOBS
```

72 Zum Teilnehmersystem MOTUS vgl. Dieter Wolff. Benutzer-Handbuch des Teilnehmersystems MOTUS für die EDV-Anlage CD 3300. Giessen 1976.

```
10 FOR I=1 TO 100
20   PRINT I, I*I, SQR(I)
30 NEXT I
40 END
*  ENDE DES BASIC-JOBS                          ENDE DES FOSBIC-JOBS
END¬                                 ENDE DER FOSBIC-JOBEINGABE
DEP,INP¬                             UEBERGABE DES FOSBIC-JOBS ZUR BEARBEITUNG

AM ENDE JEDES MOTUS-BEFEHLS STEHT EIN ¬ ZEICHEN ALS SENDEZEICHEN ZUR UEBERGABE
DER ANWEISUNG AN DEN COMPUTER CD 3300
```

Bei der Programmausgabe auf dem File „Programmergebnisse“ und einer anschliessenden Betrachtung durch den Terminalbenutzer sind lediglich einige dem FOSBIC-Job nachfolgende MOTUS-Anweisungen zu ändern.

```
MOTUS-ANWEISUNG                      BEMERKUNG
---------------                      ---------
LI,150200¬                           SITZUNGSBEGINN DES BENUTZERS 150200
A,,,PROGRAMMEINGABE¬                 FILEDEFINITION ZUR PROGRAMMEINGABE
O,,INP,,PROGRAMMEINGABE,,,O¬         OEFFNEN DER PROGRAMMEINGABE
BUILD,INP¬                           ANFANGSBEFEHL ZUM AUFBAU EINES FOSBIC-JOBS
INP,1                                DER FOSBIC-JOB BEGINNT BEI RECORD 1
$JOB,150200,/507TEST,1,500,,,,               ANFANG DES FOSBIC-JOBS
%SCHED,CLASS=C,CORE=64,SCR=40,OP=1(101)
$*DEF(O,,PUM,150200,BASIC)
$BASI,PUM
*3  ANFANG DES BASIC-JOBS
10 FOR I=1 TO 100
20   PRINT I, I*I, SQR(I)
30 NEXT I
40 END
*  ENDE DES BASIC-JOBS                          ENDE DES FOSBIC-JOBS
END¬                                 ENDE DER FOSBIC-JOBEINGABE
A,,,PROGRAMMAUSGABE¬                 ANLEGEN EINES FILES ZUR PROGRAMMAUSGABE
DEP,INP,,PROGRAMMAUSGABE¬            UEBERGABE DES FOSBIC-JOBS ZUR BEARBEITUNG
STA¬                                 WENN DIE STATUSABFRAGE ≠OUTPUT≠ ERGIBT KANN
                                     DAS FILE DER PROGRAMMAUSGABE GEOEFFNET WERDEN
O,,OUT,,PROGRAMMAUSGABE¬             OEFFNEN DER PROGRAMMAUSGABE
SHOW,OUT¬                            ANFANGSBEFEHL ZUM BETRACHTEN EINES FILES
DPF,1¬                               VORWAERTSBLAETTERN IN PROGRAMMAUSGABE AB RECORD 1
DPF,12¬                              ZWEITER BILDSCHIRMINHALT
DPF,23¬                              DRITTER BILDSCHIRMINHALT

AM ENDE JEDES MOTUS-BEFEHLS STEHT EIN ¬ ZEICHEN ALS SENDEZEICHEN ZUR UEBERGABE
DER ANWEISUNG AN DEN COMPUTER CD 3300
```

Die in den obigen Beispielen unter der Rubrik „Bemerkung“ stehenden Zeichenfolgen müssen bei der Eingabe unter MOTUS weggelassen werden; sie dienen hier nur zur Kurzerläuterung der MOTUS-Anweisungen.

34 FOSBIC-Variation

Für den Anwender besteht bei fundierten Kenntnissen in FORTRAN IV die Möglichkeit, Variationen am FOSBIC-Compilersystem vorzunehmen[73]. Zum Teil ergeben sich diese Variationsmöglichkeiten bereits aus dem bestehenden modularen Aufbau oder werden durch die Modulstruktur induziert.

341 Variation der Fehlerdiagnostik

Bis auf wenige Ausnahmen enthalten die Programme COMERR und EXERR alle Fehlermeldungen des FOSBIC-Compilersystems. Hierdurch bieten sich mehrere Variationsmöglichkeiten an, die einfach zu verwirklichen sind:

- Übersetzung der Diagnostik von Englisch in andere Sprachen
- Schaltereinbau zur Auswahl der Diagnostiksprache
- Speicherung der Diagnostik auf einem File.

Die Übersetzung der Diagnostik von Englisch in beliebige Sprachen erfordert mehr Sprach- als Programmierkenntnisse, da lediglich die entsprechenden FORMAT-Anweisungen durch übersetzte Anweisungen in den Diagnoseprogrammen zu ersetzen sind. Der Einbau eines Schalters zur Auswahl der Diagnostiksprache empfiehlt sich über die Stern(*)-Karte des BASIC-Jobs zu steuern, indem dort ab der dritten Spalte die entsprechende Sprache – eventuell mit einem Kurznamen – angegeben wird. Im Initialisierungsprogramm ZINITL wird auf Grund dieses Textes analog zum Schalter NSTZEI ein neuer Schalter – z. B. NDIAG – auf einen bestimmten Wert gesetzt (z. B. 1 = Englisch, 2 = Deutsch, 3 = Französisch usw). Dieser Schalter NDIAG wird als Variable mit in die COMMON-Blöcke aufgenommen. Entsprechend dem Wert dieses Schalters wählen die Diagnoseprogramme COMERR und EXERR die Fehlertexte aus. Da für den Fall, dass dem Benutzer mehrere Diagnostiksprachen angeboten werden und damit ein erheblicher Speicherplatzbedarf – pro Sprache ca. 1850 Doppelworte für COMERR und ca. 1410 Doppelworte für EXERR – entsteht, empfiehlt sich die Auslagerung der FORMAT-Anweisungen auf Permanent-Files. Pro Sprache wäre eine Datei aufzubauen. Die WRITE-Anweisungen der Diagnoseprogramme müssen diesfalls auf variable Formate umgestellt werden. Zum Beispiel würde WRITE(IWC,120) durch WRITE(IWC,IFOR) ersetzt. In den Vektor IFOR ist dann vor Ausführung der WRITE-Anweisung, von dem durch den Schalter NDIAG bestimmten File die durch NERROR angesprochene Fehlermeldung herauszusuchen und einzulesen.

Analog zur Variation der Fehlerdiagnostik bietet sich eine Erweiterung der Hinweise auf BASIC Textbücher an, die ebenfalls über Dateien speicherplatzsparend realisiert werden kann.

73 Vgl. auch: Horning, Structuring Compiler Development, p. 504–512.

342 Variation der Kennwortstruktur

Komplizierter ist die Variation der Kennwortstruktur, die zum Beispiel englische BASIC-Kennworte durch deutsche Kennworte ersetzt. Hierzu müssten die jeweiligen Kennwortabfragen auf die neuen (deutschen) Kennworte umgestellt werden. Da die Erkennung der Kennworte nicht in einem zentralen Modul geschieht, müssen hierzu alle Einzelabfragen in den Programmen MAIN, MATTRA und ZFILE ersetzt werden.
Sollte zum Beispiel das Kennwort READ in BASIC durch das Kennwort LESE ersetzt werden, so müssten die FORTRAN-Statements im Programm MAIN

```
300   IF((CARDP(IBEGST).NE.ALPH(18)).OR.(CARDP(IBEGST+2).NE.ALPH(1)))
     1 GOTO 305
       IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000
```

durch

```
300   IF((CARDP(IBEGST).NE.ALPH(12)).OR.(CARDP(IBEGST+2).NE.ALPH(19)))
     1 GOTO 305
       IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000
```

ersetzt werden.
Bei einem derartigen Eingriff in die FOSBIC-Struktur ist unbedingt darauf zu achten, dass alle Kennworte eindeutig voneinander unterschieden werden können und dass eine eventuelle Verringerung oder Erhöhung der Zeichenzahl des Kennworts berücksichtigt wird.

343 Variation der OVERLAY-Struktur

Zur Einsparung von Kernspeicherplatz lässt sich die gezeigte OVERLAY-Struktur[74] weiter verfeinern, indem die beiden Hauptelemente Compilierungs- und Ausführungssystem in weitere Teilsegmente aufgeteilt werden. Innerhalb der Hauptelemente müsste dann ein Zusatzprogramm die Steuerfunktion zur Verwaltung der Teilsegmente übernehmen.
Als Beispiel bietet sich hierzu die bereits vorgegebene Modulaufteilung in Elementar-, Matrizen- und Fileanweisungen an. Da die Matrizenanweisungen durch das Programm MATTRA verarbeitet werden, könnte dieses Programm nur in den Kernspeicher geladen werden, wenn ein MAT-Kennwort auftritt. Dasselbe gilt für die Verarbeitung der Fileanweisung durch das Programm ZFILE. Die Unterteilung der Hauptelemente in weitere Teilsegmente darf die gegebene Aufbaustruktur nicht zerstören.
Eine weitgehend unterteilte OVERLAY-Struktur sollte jedoch jederzeit einfach und in vorgegebener Weise aufgehoben werden können, um das Prinzip der Kompatibilität des FOSBIC-Compilersystems zu wahren.

74 Vgl. Abs. 32

35 FOSBIC-Extension

„Eine besondere Klasse von Compiler-Compilern" sind jene, „die gleichzeitig erweiterungsfähig sind"[75]. Das FOSBIC-Compilersystem zählt zur Klasse dieser Compiler-Compiler. Zusätzliche FOSBIC-Extensionen[76] bestehen in der Erweiterung
- des BASIC-Sprachumfangs
- der BASIC-Sprachstruktur und
- des FOSBIC-Compilersystems.

351 Extensionen des BASIC-Sprachumfangs

Die Extensionen des BASIC-Sprachumfangs können sich auf die
- Elementaranweisungen und die
- Sonderanweisungen

und bei den Sonderanweisungen insbesondere auf die
- Matrizenanweisungen
- Fileanweisungen und die
- Unterprogrammtechnik

erstrecken.

351.1 Extension der Elementaranweisungen

Der bisher gezeigte BASIC-Sprachumfang lässt die Datenausgabe auf Lochkarten nur über die Fileanweisungen zu, indem dort als Ausgabekanal die Kanalnummer des Lochkartenstanzers angegeben wird. Eine sinnvolle Extension des BASIC-Sprachumfangs ist die Integration eines PUNCH-Statements.
Im FOSBIC-Compilersystem könnte dieses Statement ohne grössere Programmiermassnahmen wie das PRINT- und das PRINT USING-Statement verarbeitet werden. Durch zusätzliche BASIC-Hardwareparameter würde die Begrenzung der Ausgabe auf 80 Spalten gewahrt und die maschinenorientierte Kanalnummer variabel gehalten. Ein zusätzlicher Steuerparameter in der Stern(*)-Karte könnte die Wahl ermöglichen, nach jedem ausgegebenen Datenelement ein Komma zu stanzen oder nicht. Dadurch wäre

75 Hopgood, Compiler, S. 152.
76 Terence W. Pratt. Programming Languages. Design and Implementation. Englewood Cliffs 1975, p. 6–10. „A substantial part of the programmer's task in constructing any large set of programs may be viewed as language extension", p. 9.

gewährleistet, dass die mit PUNCH-Statement ausgestanzten Lochkarten wiederum als Datenkarten ohne zusätzliche Massnahme in ein BASIC-Programm eingegeben werden können.
Eine Erhöhung der Anzahl der eingebauten Funktionen[77] (intrinsic function) erfordert beim FOSBIC-Compilersystem einfache Änderungen im Programm ZTRANX des Compilierungssystems und im Programm ZEVAL des Ausführungssystems. Der Umfang der speziellen Operationsbefehle müsste dazu entsprechend erweitert werden.

351.2 Extension der Sonderanweisungen

Die Ausweitung und Standardisierung der BASIC-Sonderanweisungen ist auch die zukünftige Aufgabe der ANSI BASIC-Kommission X3J2[78], mit dem Ziel ein einheitliches Enhencement BASIC zu schaffen. Das FOSBIC-Compilersystem bietet auf Grund seiner Modulstruktur die Möglichkeit, derartige Extensionen leicht einzubauen, zu testen und auf die mit den Benutzern und verschiedenen Maschinensystemen gemachten Erfahrungen abzustimmen.

351.21 Extension der Matrizenanweisungen

Bei den Matrizenanweisungen bieten sich analog zu den eingebauten Funktionen der Elementaranweisungen als zusätzliche MAT-Funktionen die Erzeugung von Diagonalmatrizen an. Zusätzliche MAT-Verarbeitungsfunktionen[79] zur
- Berechnung der Determinante (DET)
- Lösung linearer Gleichungssystme (SIM)
- Berechnung des Ranges einer Matrix (RANG) und
- Berechnung der Eigenwerte und Eigenvektoren (EIGEN)

könnten neue Wege in der Programmierung komplexer Probleme erschliessen. Bedeutsam erscheint in diesem Zusammenhang auch die Erweiterung der Matrizenanweisung auf Einzel-Operationen mit Zeilen und Spalten. Bei der Programmierung matrizenorientierter Probleme macht sich das Fehlen dieser Anweisungen besonders bemerkbar.
Dieser Mangel könnte durch Neustrukturierung des MAT LET-Statements[80] mit erweitertem numerischen MAT-Ausdruck und erweitertem numerischen Zielfeld erreicht werden.

77 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 325.
78 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 14.
79 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 331.221.
80 Vgl. Weber und Türschmann, BASIC, Bd. 2, Abs. 331.211.

⟨Erweitertes MAT LET-Statement⟩:=

MAT LET ⟨Zielfeld⟩ = ⟨MAT-Ausdruck⟩

⟨Zielfeld⟩ : = ⟨Numerisches Zielfeld⟩ 231 | ⟨Erweitertes numerisches Zielfeld⟩

⟨Erweitertes numerisches Zielfeld⟩ : = ⟨Feldname⟩ 40 (⟨Arithmetischer Ausdruck⟩ 49 ,*) | ⟨Feldname⟩ 40 (*, ⟨Arithmetischer Ausdruck⟩ 49) |

⟨Feldname⟩ 40 (*) | ⟨Numerische Variable⟩ 36

Der MAT-Ausdruck setzt sich nun aus dem bekannten numerischen MAT-Ausdruck und seiner Erweiterung zusammen.

⟨MAT-Ausdruck⟩ : = ⟨Numerischer MAT-Ausdruck⟩ 232 | ⟨Erweiterter numerischer MAT-Ausdruck⟩

⟨Erweiterter numerischer MAT-Ausdruck⟩ : = ⟨Erweitertes numerisches Herkunftsfeld⟩ ⟨Dyadischer Operator⟩ 12 ⟨⟨Arithmetischer Ausdruck⟩ 49⟩ |

⟨⟨Arithmetischer Ausdruck⟩ 49⟩ ⟨Dyadischer Operator⟩ 12 ⟨Erweitertes numerisches Herkunftfeld⟩ |

⟨Erweitertes numerisches Herkunftfeld⟩ ⟨Dyadischer MAT-Operator⟩ 233 ⟨Erweitertes numerisches Herkunftfeld⟩

Für das erweiterte numerische Herkunftfeld könnte gelten:

⟨Erweitertes numerisches Herkunftfeld⟩ := ⟨Feldname⟩ (⟨Arithmetischer Ausdruck⟩ .*) |
40 49

⟨Feldname⟩ (*. ⟨Arithmetischer Ausdruck⟩) |
40 49

⟨Feldname⟩ (*)
40

Eine Spaltenaddition der 3. Spalte der Matrix B mit der 2. Spalte von C abgespeichert in der 1. Spalte von A lautet dann

```
MAT LET A(*,1)=B(*,3)+C(*,2)
```

Die entsprechende Zeilenoperation als Subtraktion könnte zum Beispiel durch Angabe des Statements

```
MAT LET A(1,*)=B(3,*)-C(2,*)
```

erfolgen.
Ein Zeilen-Spaltenprodukt würde durch das Statement

```
MAT LET A(3,4)=B(4,*)*C(*,8)
```

durchgeführt. Die 4. Spalte der Matrix A könnte mit der Anweisung

```
MAT LET R(*)=A(*,4)
```

in den Zeilenvektor R übertragen werden.
Der Leser mag sich an weiteren selbst entworfenen Beispielen die vielfältige Einsatzmöglichkeit dieses erweiterten numerischen MAT-Ausdrucks in der Matrizenrechnung klarmachen.
Durch die Einführung einer Summenfunktion (SUM) könnte die Verbindung von der Matrizenrechnung zu BASIC-MAT-Anweisungen direkter gestaltet werden. Die Berechnung der Summe

$$C = \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{(x_{ij} - e_{ij})^2}{e_{ij}}$$

könnte dann allein durch das Statement

```
MAT LET C=SUM((X-E)**(2)/E)
```

geschehen.

351.22 Extension der Fileanweisungen

Mit den Fileanweisungen des FOSBIC-Compilersystems wird es dem BASIC-Benutzer ermöglicht, sich in die Grundlagen der Fileverarbeitung und der Datenorganisation einzuarbeiten. Umfangreiche, an praktischen Problemen orientierte Datenbanksysteme lassen sich mit den FORTRAN-Fileanweisungen des Ausführungssystems nur mit erheblichem Zeitaufwand darstellen, da bei FORTRAN intern kein direkter Datenzugriff zulässig ist.
Auf der CD 3300 des Hochschul-Rechenzentrums der Justus-Liebig-Universität Giessen existiert ein sogenanntes System LISA (**L**inked **I**ndex **S**equentiell **A**ccess), das einen direkten Zugriff auf verkettete Dateien zulässt. Mit FORTRAN können über eine Reihe von maschinenorientierten Unterprogrammen[81] wie

```
CALL BUILDF
CALL GETF
CALL FINDF
CALL INSERTF
CALL DELETEF
```

LISA-Dateien aufgebaut, sequentiell gelesen, Daten im Direktzugriff gefunden, Sätze eingefügt und aus der Datei entfernt werden. Diese unvollständige Liste der FORTRAN-Aufrufe zum Aufbau und Verwalten von LISA-Dateien wird durch weitere maschinenorientierte Anweisungen und Unterprogramme ergänzt.
Eine Erweiterung des FOSBIC-Ausführungssystems mit einer eindeutigen Definition der Schnittstellen zur Übernahme derartiger maschinenorientierter Unterprogramme ist grundsätzlich denkbar.

351.23 Extension der Unterprogrammtechnik

Durch die Erweiterung der Elementaranweisungen des FOSBIC-Compilersystems auf die Verarbeitung von ein- und mehrzeiligen DEF-Statements[82] würden die Standards von ANSI Minimal BASIC vollumfänglich abgedeckt.

81 Vgl. Control Data Corporation. Computer System 3100/3200/3300/3500. LISA/MASTER/MSOS. Reference Manual. St. Paul 1969, p. 4–19 to 4–29.
82 Vgl. Weber und Türschmann, BASIC, Bd. 1, Abs. 331.22.

Wünschenswert ist auch der Einbau eines CALL-Statements, um
- benutzereigene BASIC-Unterprogramme und
- eine Programmverkettung[83] mit dem COMMON-FILE-Statement

zu ermöglichen.
Die FOSBIC-Compilerstruktur bietet ausserdem die Möglichkeit, ein „Linkage-Modul" zu entwickeln, mit dem der BASIC-Benutzer aus dem BASIC-Programm heraus einen Direkt-Zugriff auf eine FORTRAN-Programmbibliothek hat. Innerhalb des BASIC-Programms werden durch ein CALL-Statement mit einer entsprechenden Parameterliste alle notwendigen Eingabewerte für das verwendete FORTRAN-Programm übergeben. Das „Linkage-Modul" entnimmt zunächst auf Grund der Parameterliste alle Wertinhalte aus dem BASIC-Adressbereich und überträgt sie in einen „COMMON-Linkage-Block" und sucht in einer linearen Liste das dazugehörige FORTRAN-Programm. Dieses FORTRAN-Programm entnimmt alle entsprechend seiner (Formal-)Parameterliste enthaltenen Aktual-Parameter aus dem „COMMON-Linkage-Block" und gibt nach Beendigung alle Werte wieder dorthin zurück. Anschliessend überträgt das „Linkage-Modul" die Werte aus dem „COMMON-Linkage-Block" in den BASIC-Adressteil zurück, wodurch nunmehr alle Parameterwerte im laufenden BASIC-Programm zur Weiterverarbeitung zur Verfügung stehen.

352 Extension der BASIC-Sprachstruktur

Grundsätzlich besteht auch die Möglichkeit, Module direkt durch Erweiterung der BASIC-Sprachstruktur in das FOSBIC-Compilersystem zu integrieren. Die Vorteile dieser Spracherweiterung liegen in
- einer benutzerfreundlichen Programmierung der Problemstruktur
- Plausibilitätskontrollen der Problemstruktur durch das Compilierungssystem
- benutzerfreundliche Fehlerdiagnose und
- sprachorientierter Programmaufruf.

Die Anonymität der Parameterliste, wie sie beim Aufruf von Programmen aus einer Programmbibliothek gegeben ist, kann aufgelöst werden und eine problemnahe Programmierung mit benutzerfreundlichen Plausibilitätskontrollen erfolgen.
Aus dem Gebiet der Netzplantechnik bietet sich beispielsweise eine modifizierte Version von PECOS (Project Evaluation and Cost Optimization System)[84] zur Übernahme in das FOSBIC-System an. Die zur Lösung eines Netzplanproblems erforderlichen BASIC-Statements könnten etwa wie folgt strukturiert sein, wobei die Eingangsvektoren A, B und C in einem vorausgehenden Programmteil zu besetzen sind:

83 Vgl. Weber und Türschmann, BASIC, Bd. 2, Tab. 4 und Abs. 332.13.
84 I. W. Burgeson, C. J. Snyder, E. A. Schaefer. PECOS-Project Evaluation and Cost Optimization System. IBM 1620 General Program Library, 10.3.019.

```
  ...     VORAUSGEHENDER BASIC-PROGRAMMTEIL
  ...
  ...
300 CPM-START
310 FROM=A
320 TO=B
330 TIME=C
340 NOCOST
350 K=EARLY-START
360 L=LATE-START
370 M=EARLY-END
380 N=LATE-END
390 O=TOTAL-FLOAT
400 P=INDEP-FLOAT
410 X1=TOTAL-TIME
420 NOPRINT
430 CPM-END
  ...
  ...
  ...     NACHFOLGENDER BASIC-PROGRAMMTEIL
```

Vermerkt sei, dass die Vektoren A, B, C, K, L, M, N, O und P die gleichen Feldgrenzen aufweisen. A enthält alle Aktivitätsausgangspunkte, B alle Aktivitätsendpunkte und C die Zeitdauer aller Aktivitäten. NOCOST bewirkt Unterdrückung der Kostenanalyse. Die Ausgangsvektoren enthalten die Pufferzeiten, wobei O auch den kritischen Weg anzeigt.
Für die Simplexmethode ist folgende Zugriffsweise denkbar:

```
SIMPLEX-START          v BEGINN DES SIMPLEX-MODULS
Z=MAX(X*B)             v ZIELFUNKTION
S*X≤D                  v RESTRIKTIONEN
X≥0                    v NICHTNEGATIVITAETSBEDINGUNG
NOPRINT                v KEINE ZWISCHENTABLEAUAUSGABE
SIMPLEX-END            v ENDE DES SIMPLEX-MODULS
```

Dabei bezeichnet A den Koeffizientenvektor der Zielfunktion, R die Koeffizientenmatrix der Restikonen, C den Vektor der Beschränkungswerte und X den Lösungsvektor aller Variablen. X muss der Gesamtzahl der Entscheidungs- und Schlupfvariablen entsprechend dimensioniert werden.
Mit Hilfe der Simplexmethode lassen sich auch speziell strukturierte Probleme der nichtlinearen Optimierung bearbeiten, indem lineare OR-Module innerhalb frei programmierbarer Modellstrukturen höherer Ordnung beliebig oft benutzbar gemacht werden.
Als Beispiel sei hier angenommen, dass sich eine Optimierungsaufgabe lediglich bezüglich der Zielfunktion von einem typischen LP-Problem unterscheidet. Die Zielfunktion enthalte zwei nichtlinear – beispielsweise multiplikativ – miteinander verknüpfte Variable X_i und X_j; alle Restriktionen seien linear.
Dieses Problem lässt sich stufenweise unter Verwendung der Simplexmethode in der Weise bearbeiten, dass eine der beiden Variablen bei stufenweiser Problembearbeitung via Simplexalgorithmus konstant gehalten und durch ein übergeordnetes Programm in ihrem Wert innerhalb vorgegebener Grenzen sukzessive modifiziert wird (vgl. Abbildung 22). Dadurch wird das an sich nichtlineare Problem auf ein typisch lineares – über eine entsprechende Programmschleife mehrfach abzuarbeitendes – LP-Problem reduziert. Ver-

merkt sei, dass eine effiziente Problembearbeitung nur bei leichtem Zugriff auf das OR-Grundmodul möglich ist.

```
 10 DIM A(10), B(9), C(5), D(5), R(5,10), S(5,9), X(14)
 20 MAT READ A,R,C
 30 READ G1, G2, G3, M, I0, J0
 40 FOR I9=G1 TO G2 STEP G3
 50   LET K=0
 60   FOR J=1 TO 10
 70     IF J = J0 GOTO 130
 80     LET K=K+1
 90     IF I = I0 GOTO 120
100     LET B(K)=A(J)
110     GOTO 130
120     LET B(K)=A(J)*I9
130     FOR I=1 TO 5
140       IF J = J0 GOTO 170
150       LET S(I,K)=R(I,J)
160       GOTO 180
170       LET D(I)=C(I)-R(I,J)*I9
180     NEXT I
190   NEXT J
200   SIMPLEX-START
210   Z=MAX(B*X)
220   R*X≤D
230   X≥0
240   NOPRINT
250   SIMPLEX-END
260   LET Z=Z-A(J0)*I9
270   IF Z ≥ M GOTO 310
275   PRINT ≠ZWISCHENERGEBNIS≠
280   MAT PRINT X
290   PRINT Z
300 NEXT I9
310 PRINT ≠ENDERGEBNIS≠
320 MAT PRINT X
330 PRINT Z
340 END LIST DATA
```

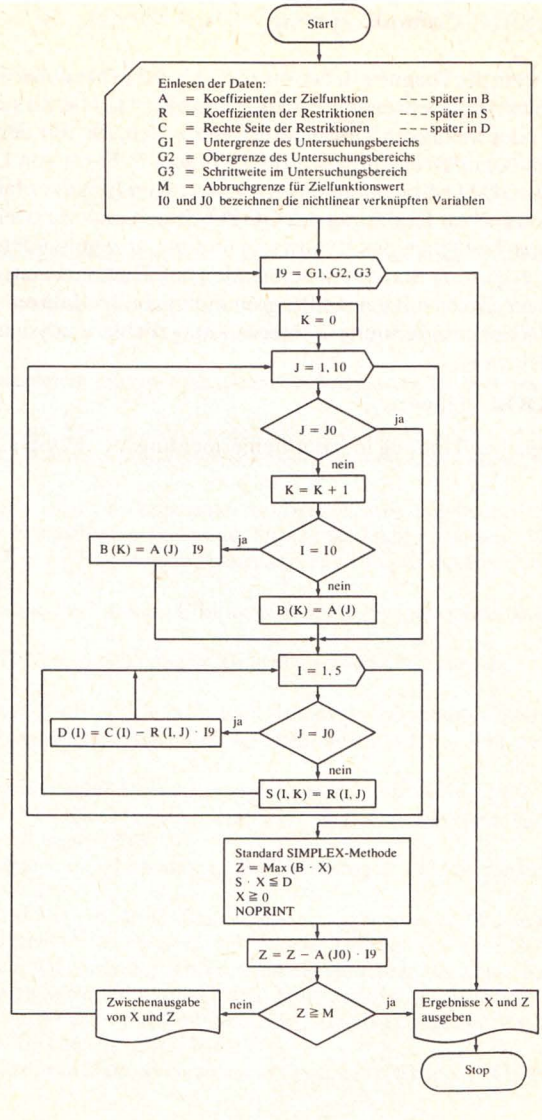


Abbildung 22. OR-Modell

353 Extension des FOSBIC-Compilersystems

Da im FOSBIC-Compilersystem die Diagnostik bereits weitgehend in Modulstruktur[85] vorliegt, bietet sich hier der Ausbau zu einem selbständig lehrenden BASIC-Compiler an. Hierzu könnte jede Fehlermeldung eine Verzweigung zu einem oder mehreren Lehrmodulen herstellen, die mit dem Benutzer im Dialog das als falsch erkannte Statement oder den entsprechenden Teilinhalt in Form von Lehrschleifen durchgehen. Würde zum Beispiel in einem GOTO-Statement das Kennwort vom Benutzer falsch angegeben, so könnte zunächst in eine spezielle Routine zur Erklärung des GOTO-Statements verzweigt werden. Anschliessend können einige Fragestellungen bezüglich des Kennworts und der anzugebenden Statementnummer überprüfen, ob der Benutzer das erläuterte Statement verstanden hat. Erst nachdem alle Fragen vom Benutzer richtig beantwortet sind, kann dieser mit seiner Programmeingabe fortfahren.
Eine Rückkopplung zwischen Fehlererkennung, Fehlermeldung, Fehleranalyse und Fehlerbehebung durch den Benutzer ist mit dem Hinweis

SEE BASIC TEXTBOOK ⟨Hinweis⟩

für den im Batchbetrieb möglichen Umfang in jeder Fehlermeldung des FOSBIC-Compilersystems bereits realisiert[86].

85 Vgl. Abs. 335
86 Vgl. Abs. 334

4 Literatur

A[merican N[ational] S[tandards] I[nstitute]. X3J2 BASIC Standard Committee. Proposed American National Standard for Minimal BASIC. January 1976. Hanover 1976

Bates, D. (ed.). Commercial Language Systems. Maidenhead: Infotech, 1974 = Infotech State of the Art Report 19

Bauer, Friedrich L. and Eickel, J. (ed.). Compiler Construction. An Advanced Course. Berlin: Springer, 1974 = Lecture Notes in Computer Science 21

Blumenthal, Sherman C. Management Information Systems. A Framework for Planning and Development. Englewood Cliffs: Prentice-Hall, 1969

Boon, C. (ed.). Computer Design. Maidenhead: Infotech, 1974 = Infotech State of the Art Report 17

Bull, G. M.; Freeman, W. and Garland, S. J. Specification for Standard BASIC. Manchester: National Computing Center, 1973 (out of print)

Burch, John G., Jr. and Strater, Felix R., Jr. Information Systems. Theory and Practice. Santa Barbara: Hamilton, 1974

Burge, William H. Recursive Programming Techniques. Reading: Addison-Wesley, 1975

Cocke, John and Schwartz, J. T. (ed.). Programming Languages and Their Compilers. 2nd ed., New York: Courant Institute of Mathematical Sciences, 1970

Couger, J. Daniel. Computers and the Schools of Business. Boulder: School of Business Administration, University of Colorado, 1967

Dunnagan, Trinka. CONDUIT Technical Transport Guidelines. Iowa City: American College Testing Program, 1973 = CONDUIT Report No. 2

Garland, Stephan J. Dartmouth BASIC: A Specification. Hanover: Dartmouth College, 1973 = TMO28

Germain, Clarence B. Das Programmier-Handbuch der IBM/360. Ein Lehr- und Arbeitsbuch. 2. Aufl., München: Hanser, 1970

Gries, David. Compiler Construction for Digital Computers. New York: Wiley, 1971

Gritsch, Anna und Gritsch, Rüdiger. Das Programmieren von Computern. Ein Lehr- und Lernbuch unter Verwendung von FORTRAN. München: Hanser, 1972

Hackl, Clemens E. (ed.). Programming Methodology. Berlin: Springer, 1975 = Lecture Notes in Computer Science 23

Henshon, Elaine M. VIEWIT: A Take-Apart Compiler. M.S. thesis. Worcester: Worcester Polytechnic Institute, 1973

Higman, B. Programmiersprachen – Eine vergleichende Studie. Leipzig: Teubner, 1971

Hopgood, F. R. Compiler – Die Übersetzung von Programmiersprachen. München: Hanser, 1970

Horowitz, Ellis. Practical Strategies for Developing Large Software Systems. Reading: Addison-Wesley, 1975

Isaacs, Gerald L. Interdialect Translatability of the BASIC Programming Language. Iowa City: American College Testing Program, 1972 = ACT Technical Bulletin No. 11

Johnson, Lyle R. System Structure in Data, Programs and Computers. Englewood Cliffs: Prentice-Hall, 1970

Joslin, Edward O. An Introduction to Computer Systems. 2nd ed., Arlington: College Readings, 1969
Kemeny, John G. and Kurtz, Thomas E. BASIC. User's Manual. [1st ed.] Hanover: Dartmouth College Computation Center, 1964
Kemeny, John G. and Kurtz, Thomas E. BASIC Programming. 2nd ed., New York: Wiley, 1971
Lee, John A. The Anatomy of a Compiler. 2nd ed., New York: Van Nostrand Reinhold, 1974
Luehrmann, Arthur. An Elaboration of Some Thoughts on a Graphical Syntax in BASIC. Hanover: Dartmouth College, 1974
Maurer, Hermann. Theoretische Grundlagen der Programmiersprachen. Mannheim: Hochschultaschenbücher, 1969
McKeeman, William M.; Horning, James J. and Wortman, David B. A Compiler Generator. Englewood Cliffs: Prentice-Hall, 1970
Myers, Glenford J. Reliable Software Through Composite Design. New York: Petrocelli/Charter, 1975
Naur, Peter. Concise Survey of Computer Methods. New York: Petrocelli/Charter, 1974
Nederlands Normalisatie-instituut. Automatische gegevensverwerking. Programmeertaal Elementair Basic. Rijswijk: Nederlands Normalisatieinstituut, 1975 = NPR 3591
Nederlands Normalisatie-instituut. Proposal for ISO-Standard on Programming Language Elementary BASIC. Rijswijk: Nederlands Normalisatie-instituut, 1976 = ISO/TC 97/SC5 (Netherlands)
Pollack, Bary W. (ed.). Compiler Techniques. Princeton: Auerbach, 1972
Pratt, Terrence W. Programming Languages. Design and Implementation. Englewood Cliffs: Prentice-Hall, 1975
Rohl, J. S. An Introduction to Compiler Writing. New York: American Elsevier, 1975
Schneider, Hans Jürgen. Compiler-Aufbau und Arbeitsweise. Berlin: de Gruyter, 1975
Sharpe, William F. University of Washington BASIC Interpretive Compiler UWBIC. Seattle: University of Washington, Graduate School of Business Administration, 1967 = Technical Report Series, No. 3
Tou, Julius T. (ed.). Software Engineering. Coins III. 2 Vol., New York: Academic Press, 1970
Weber, Karl und Türschmann, Carl Wolfram. BASIC. Lehr- und Handbuch der Programmiersprache BASIC mit wirtschaftswissenschaftlichen Anwendungsbeispielen. UTB 588 und 589, Bern und Stuttgart: Haupt, 1977
Wolff, Dieter. Benutzer-Handbuch des Teilnehmersystems MOTUS für die EDV-Anlage CD 3300. Giessen: Hochschulrechenzentrum der Justus Liebig-Universität Giessen, 1976

5 FOSBIC-Programmlisten

51 Hauptprogramm und intermediäre Unterprogramme

```
LN 0001     C---- PROGRAMM OVERT                                                  STE00001
LN 0002     C---- DIE FOLGENDE ROUTINE IST EINE ZUSATZROUTINE = HAUPTELEMENT, DIE  STE00002
LN 0003     C---- NUR BEIM ARBEITEN IN OVERLAY-TECHNIK NOTWENDIG IST.              STE00003
LN 0004     C---- DIESE ROUTINE IST BEI NORMALEN ABLAUF DES SYSTEMS ZU ENTFERNEN.  STE00004
LN 0005     C---- AUSSERDEM SIND NOCH EINIGE EINFACHE AENDERUNGEN IN DEN PROGRAMMEN STE00005
LN 0006     C----               MAIN                                              STE00006
LN 0007     C---- UND           ZEXEC                                             STE00007
LN 0008     C---- VORZUNEHMEN UND EINIGE UNTERPROGRAMME UMZUSTELLEN.              STE00008
LN 0009     C---- DIE UMZUSTELLENDEN UNTERPROGRAMME SIND                          STE00009
LN 0010     C----               ZHOPPR                                            STE00010
LN 0011     C----               STRING                                            STE00011
LN 0012     C----               ZDIGIT                                            STE00012
LN 0013     C----               ZALPH                                             STE00013
LN 0014     C----               ENDFT                                             STE00014
LN 0015     C---- DIE VORZUNEHMENDEN AENDERUNGEN SIND DURCH TEXT AUSFUEHRLICH      STE00015
LN 0016     C---- GEKENNZEICHNET,WOBEI DER TEXT DURCH C---- EINGELEITET WIRD.      STE00016
LN 0017     C---- ES EMPFIEHLT SICH DIE JEWEILIGEN AENDERUNGEN SO VORZUNEHMEN,DASS STE00017
LN 0018     C---- DAS URSPRUENGLICHE PROGRAMM ERKENNBAR BLEIBT.                    STE00018
LN 0019     C***  ZUSATZROUTINE -HAUPTELEMENT BEI OVERLAY-TECHNIK                 STE00019
LN 0020           COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   STE00020
LN 0021          1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOP,NIRET,NSTLST,INEXT, STE00021
LN 0022          2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL, STE00022
LN 0023          3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRT,NIMAGE, STE00023
LN 0024          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI   STE00024
LN 0025           COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,MIRMAX,   STE00025
LN 0026          1NIFMAX,INTZEI                                                   STE00026
LN 0027           COMMON// CARDI(80),MERKER(26,2),CARP(140),                      STE00027
LN 0028          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    STE00028
LN 0029          1IRET(20),XXX(4),NFILE(25,3)                                     STE00029
LN 0030           COMMON// ISTLST(340),LISTST(340)                                STE00030
LN 0031           COMMON// DATAN(330)                                             ***
LN 0032           COMMON// DATA(3700)                                             STE00032
LN 0033           DIMENSION IPROG(3700)                                           STE00033
LN 0034           EQUIVALENCE (DATA(1),IPROG(1))                                  STE00034
LN 0035           OVERLAY(0)                                                      STE00035
LN 0036           DATA IDISK/3HPUM/                                               STE00036
LN 0037      2    CALL UFOVER(1,IDISK)                                            STE00037
LN 0038           CALL UFOVER(2,IDISK)                                            STE00038
LN 0039           IF(CARDP(1).EQ.ASTRSK) GOTO 2                                   STE00039
LN 0040     20    READ(IRC,1) CARD                                                STE00040
LN 0041      1    FORMAT(80A1)                                                    STE00041
LN 0042           IF(IFEOF(IRC).EQ.-1) GOTO 10                                    STE00042
LN 0043           IF(CARD(1).EQ.ASTRSK) GOTO 2                                    STE00043
LN 0044           WRITE(IWC,3) CARD                                               STE00044
LN 0045      3    FORMAT(20X,80A1)                                                STE00045
LN 0046           GOTO 20                                                         STE00046
LN 0047     10    STOP                                                            STE00047
LN 0048           END                                                             STE00048


     USASI FORTRAN DIAGNOSTIC RESULTS FOR FTN.MAIN

   LINE    S   ERRNUM                    MESSAGE

   0035    N    0026     NON-USASI OVERLAY OR SEGMENT STATEMENT USED.
```

ZHOPPR

```
LN 0001           SUBROUTINE ZHOPPR(VALUE,NSTOP,IFR,NSW)                          ZH000001
LN 0002     C---- DIE SUBROUTINE **ZHOPPR** MUSS NORMALERWEISE DEM PROGRAMM **MAIN**ZH000002
LN 0003     C---- FOLGEN UND NICHT WIE JETZT VORAUSGEHEN,WENN IN OVERLAY-TECHNIK    ZH000003
LN 0004     C---- GEARBEITET WIRD.DIESE ROUTINE WIRD SOWOHL WAEHREND               ZH000004
LN 0005     C---- COMPILIERUNG ALS AUCH WAEHREND DER AUSFUEHRUNG DURCH ZEXEC BENUTZTZH000005
LN 0006           COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,     ZH000006
LN 0007          1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZH000007
LN 0008          2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZH000008
LN 0009          3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZH000009
LN 0010          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZH000010
LN 0011           COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     ZH000011
LN 0012          1NIFMAX,INTZEI                                                    ZH000012
LN 0013           COMMON// CARDT(80),MERKER(26,2),CARP(140),                       ZH000013
LN 0014          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),     ZH000014
LN 0015          1IRET(20),XXX(4),NFILE(25,3)                                      ZH000015
LN 0016           COMMON// ISTLST(340),LISTST(340)                                 ZH000016
LN 0017           COMMON// DATAN(330)                                              ***
LN 0018           COMMON// DATA(3700)                                              ZH000018
LN 0019           DIMENSION IPROG(3700)                                            ZH000019
LN 0020           EQUIVALENCE (DATA(1),IPROG(1))                                   ZH000020
LN 0021     C                                                                      ZH000021
LN 0022     C*****IF NSW=-1 THIS IS AN INITIALIZATION CALL                         ZH000022
LN 0023     C*****IF NSW= 0 THIS IS A COMPILATION OF A DATA STATEMENT              ZH000023
LN 0024     C*****IF NSW= 1 THIS IS AN EXECUTION TO ABTAIN A DATA                  ZH000024
LN 0025     C*****IF NSW= 2 THIS IS AN EXECUTION OF A RESTORE COMMAND              ZH000025
LN 0026     C*****IF NSW= 3 THIS IS AN EXECUTION OF AN INPUT COMMAND               ZH000026
LN 0027           NSTOP=0                                                          ZH000027
LN 0028           IF(NSW.EQ.3) GOTO 10                                             ZH000028
LN 0029           IF(NSW.GE.0) GOTO 5                                              ZH000029
LN 0030           IPOS=0                                                           ZH000030
LN 0031           IPOSDN=1                                                         ZH000031
LN 0032           RETURN                                                           ZH000032
LN 0033         5 IF(NSW.EQ.0) GOTO 15                                             ZH000033
LN 0034           IF(NSW.EQ.2) GOTO 500                                            ZH000034
LN 0035     C                                                                      ZH000035
LN 0036     C*****EXTRACT VALUE FROM INTERNAL DATA VECTOR IF ANY LEFT              ZH000036
LN 0037           IF(IPOSDN.GT.NEXTDT) GOTO 10                                     ZH000037
LN 0038     C*****INTERNAL DATA LEFT -- EXTRACT NEXT ONE                           ZH000038
LN 0039           VALUE=DATAN(IPOSDN)                                              ZH000039
LN 0040           IPOSDN=IPOSDN+1                                                  ZH000040
LN 0041           RETURN                                                           ZH000041
LN 0042     C*****NEW POSITION OF IPOSDN BY RESTORE COMMAND                        ZH000042
LN 0043      500  NEWPOS=IFR                                                      ZH000043
LN 0044           IPOSDN=IPOSDN-NEWPOS                                             ZH000044
LN 0045           IF(NEWPOS.EQ.0) IPOSDN=1                                         ZH000045
LN 0046           IF(IPOSDN.GT.NCELLD) GOTO 502                                    ZH000046
LN 0047           IF(IPOSDN.GT.0) GOTO 501                                         ZH000047
LN 0048      502  VALUE=IPOSDN                                                    ZH000048
LN 0049           NSTOP=4                                                          ZH000049
LN 0050      501  RETURN                                                          ZH000050
LN 0051     C                                                                      ZH000051
LN 0052     C*****NO INTERNAL DATA LEFT -- CHECK READING BUFFER                    ZH000052
LN 0053      10   IF(IPOS.LE.0) GOTO 12                                           ZH000053
LN 0054     C                                                                      ZH000054
LN 0055     C*****READING BUFFER NOT EMPTY -- DELIVER NEXT NUMBER                  ZH000055
LN 0056           IWORD=ITOT-IPOS+1                                                ZH000056
LN 0057           VALUE=BUFFER(IWORD)                                              ZH000057
LN 0058           IPOS=IPOS-1                                                      ZH000058
LN 0059           RETURN                                                           ZH000059
LN 0060     C                                                                      ZH000060
LN 0061     C*****READING BUFFER EMPTY -- FILL IT UP BY READING A CARD             ZH000061
```

```
LN 0062     12   READ(IRC,1) (CARDT(I),I=1,80)                                    ZH000062
LN 0063     C*-*-                                                                 ZH000065
LN 0064     C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(IRC)-          ZH000066
LN 0065     C*-*-      ----CHECK IT----        ----CHECK IT----                   ZH000067
LN 0066     C*-*-                                                                 ZH000068
LN 0067           IF(IFEOF(IRC).EQ.-1) GOTO 220                                   ZH000069
LN 0068           NSW=3                                                           ***
LN 0069           NQUOTE=1                                                        ***
LN 0070       1   FORMAT(80A1)                                                    ZH000070
LN 0071     C     COMPRESS CARDT                                                  ZH000071
LN 0072           LNGCRP=0                                                        ZH000072
LN 0073           DO 400 I=1,80                                                   ZH000073
LN 0074           CARDP(I)=BLANK                                                  ZH000074
LN 0075           IF(CARDT(I).EQ.DQUOTE) CARDT(I)=QUOTE                           ZH000075
LN 0076           IF(CARDT(I).EQ.QUOTE) NQUOTE=-NQUOTE                            ZH000076
LN 0077           IF((CARDT(I).EQ.BLANK).AND.(NQUOTE.EQ.1)) GOTO 400              ZH000077
LN 0078           LNGCRP=LNGCRP+1                                                 ZH000078
LN 0079           CARDP(LNGCRP)=CARDT(I)                                          ZH000079
LN 0080     400   CONTINUE                                                        ZH000080
LN 0081           IF(LNGCRP.EQ.0) GOTO 12                                         ZH000081
LN 0082           IF(NQUOTE.EQ.1) GOTO 403                                        ZH000082
LN 0083           NSTOP=3                                                         ZH000083
LN 0084           RETURN                                                          ZH000084
LN 0085     C*****CHECK FOR A CONTROL CARD (ASTERISK IN COLUMN 1)                 ZH000085
LN 0086     403   IF(CARDT(1).NE.ASTRSK) GOTO 15                                  ***
LN 0087     220   NSTOP=5                                                         ZH000087
LN 0088           RETURN                                                          ZH000088
LN 0089     C                                                                     ZH000089
LN 0090      15   DO 20 K=1,40                                                    ZH000090
LN 0091      20   BUFFER(K)=0.                                                    ZH000091
LN 0092           IPOS=1                                                          ZH000092
LN 0093           IDEC=-1                                                         ZH000093
LN 0094           ISGN=0                                                          ZH000094
LN 0095           INUM=0                                                          ZH000095
LN 0096           IEF=0                                                           ZH000096
LN 0097      90   DO 200 I=IFR,81                                                 ZH000097
LN 0098           IF(I.GE.LNGCRP+1) GOTO 180                                      ZH000098
LN 0099           IF(CARDP(I).EQ.QUOTE) GOTO 405                                  ZH000099
LN 0100           CALL ZDIGIT(CARDP(I),J)                                         ZH000100
LN 0101           IF(J.LE.10) GOTO 150                                            ZH000101
LN 0102           IF((CARDP(I).EQ.ALPH(5)).OR.(CARDP(I).EQ.DECMAL).OR.(CARDP(I).EQ. ZH000102
LN 0103          1PLUS).OR.(CARDP(I).EQ.COMMA).OR.(CARDP(I).EQ.CMINUS)) GOTO 2302   ZH000103
LN 0104           NSTOP=2                                                         ZH000104
LN 0105            RETURN                                                         ZH000105
LN 0106      2302 IF(CARDP(I).NE.DECMAL) GOTO 120                                 ZH000106
LN 0107           IDEC=I                                                          ZH000107
LN 0108           GOTO 200                                                        ZH000108
LN 0109      120  IF(CARDP(I).NE.CMINUS) GOTO 180                                 ZH000109
LN 0110           ISGN=-1                                                         ZH000110
LN 0111           GOTO 200                                                        ZH000111
LN 0112     C*****DIGIT FOUND                                                     ZH000112
LN 0113      150  INUM=1                                                          ZH000113
LN 0114           DIG=J-1                                                         ZH000114
LN 0115           BUFFER(IPOS)=DIG+(BUFFER(IPOS)*10.)                             ZH000115
LN 0116           GOTO 200                                                        ZH000116
LN 0117     C*****NON-NUMERIC CHARACTER FOUND                                     ZH000117
LN 0118      180  IF(CARDP(I).EQ.PLUS) GOTO 200                                   ZH000118
LN 0119           IF(INUM.EQ.0) GOTO 200                                          ZH000119
LN 0120           INUM=0                                                          ZH000120
LN 0121           IF(ISGN.LT.0) BUFFER(IPOS)=-BUFFER(IPOS)                        ZH000121
LN 0122           ISGN=0                                                          ZH000122
```

```
LN 0123           IF(IDEC.LT.1) GOTO 190                                ZH000123
LN 0124           IDIFF=I-IDEC-1                                        ZH000124
LN 0125           IDEC=-1                                               ZH000125
LN 0126           IF(IDIFF.LE.0) GOTO 190                               ZH000126
LN 0127           BUFFER(IPOS)=BUFFER(IPOS)/(10.**IDIFF)                ZH000127
LN 0128      190  IF(IEF.EQ.1) GOTO 240                                 ZH000128
LN 0129           IF(CARDP(I).NE.ALPH(5)) GOTO 230                      ZH000129
LN 0130           IEF=1                                                 ZH000130
LN 0131           BUFIPO=BUFFER(IPOS)                                   ZH000131
LN 0132           BUFFER(IPOS)=0.                                       ZH000132
LN 0133           GOTO 200                                              ZH000133
LN 0134      240  IEF=0                                                 ZH000134
LN 0135           IBUFF=BUFFER(IPOS)                                    ZH000135
LN 0136           IF(IABS(IBUFF).GE.IEXPO) GOTO 219                     ZH000136
LN 0137           BUFF=10**IABS(IBUFF)                                  ZH000137
LN 0138           IF(IBUFF.GE.0) BUFFER(IPOS)=BUFIPO*BUFF               ZH000138
LN 0139           IF(IBUFF.LT.0) BUFFER(IPOS)=BUFIPO/BUFF               ZH000139
LN 0140      230  IPOS=IPOS+1                                           ZH000140
LN 0141      200  CONTINUE                                              ZH000141
LN 0142           IPOS=IPOS-1                                           ZH000142
LN 0143           ITOT=IPOS                                             ZH000143
LN 0144           IF(NSW.EQ.3) GOTO 10                                  ZH000144
LN 0145           NUMBUF=ITOT                                           ZH000145
LN 0146           IPOS=0                                                ZH000146
LN 0147           RETURN                                                ZH000147
LN 0148      219  NSTOP=1                                               ZH000148
LN 0149           RETURN                                                ZH000149
LN 0150      C                                                          ZH000150
LN 0151      C*****TAKE STRINGS                                         ZH000151
LN 0152       405 IFR=I+1                                               ZH000152
LN 0153           DO 406 J=IFR,LNGCRP                                   ZH000153
LN 0154           IF(CARDP(J).EQ.QUOTE) GOTO 407                        ZH000154
LN 0155       406 CONTINUE                                              ZH000155
LN 0156       407 ITO=J-1                                               ZH000156
LN 0157       412 IEND=IFR+4                                            ZH000157
LN 0158           IF(IEND.GT.ITO) IEND=ITO                              ZH000158
LN 0159           CALL STRING(IFR,IEND,IX)                              ZH000159
LN 0160           BUFFER(IPOS)=IX                                       ZH000160
LN 0161           IPOS=IPOS+1                                           ZH000161
LN 0162           IF(IEND.GE.ITO) GOTO 415                              ZH000162
LN 0163           IFR=IEND+1                                            ZH000163
LN 0164           GOTO 412                                              ZH000164
LN 0165       415 IFR=ITO+2                                             ZH000165
LN 0166           GOTO 90                                               ZH000166
LN 0167           END                                                   ZH000167
```

```
     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZHOPPR

          NO ERRORS
```

```
LN 0001          SUBROUTINE ZDIGIT(ZEICH,IA)                                    ZDI00001
LN 0002     C---- DIE SUBROUTINE **STRING** MUSS NORMALERWEISE DEM PROGRAMM **MAIN**ZDI00002
LN 0003     C---- FOLGEN UND NICHT WIE JETZT VORAUSGEHEN,WENN IN OVERLAY-TECHNIK    ZDI00003
LN 0004     C---- GEARBEITET WIRD.                                                 ZDI00004
LN 0005          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,      ZDI00005
LN 0006         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT, ZDI00006
LN 0007         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZDI00007
LN 0008         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZDI00008
LN 0009         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZDI00009
LN 0010          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     ZDI00010
LN 0011         1NIFMAX,INTZEI                                                     ZDI00011
LN 0012          COMMON// CARDT(80),MERKER(26,2),CARP(140),                        ZDI00012
LN 0013         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),      ZDI00013
LN 0014         1IRET(20),XXX(4),NFILE(25,3)                                       ZDI00014
LN 0015          COMMON// ISTLST(340),LISTST(340)                                  ZDI00015
LN 0016          COMMON// DATAN(330)                                               ***
LN 0017          COMMON// DATA(3700)                                               ZDI00017
LN 0018          DIMENSION IPROG(3700)                                             ZDI00018
LN 0019          EQUIVALENCE (DATA(1),IPROG(1))                                    ZDI00019
LN 0020          DO 10 I=1,10                                                      ZDI00020
LN 0021          IF(ZEICH.EQ.DIGIT(I)) GOTO 20                                     ZDI00021
LN 0022      10  CONTINUE                                                         ZDI00022
LN 0023          I=11                                                              ZDI00023
LN 0024      20  IA=I                                                            ZDI00024
LN 0025          RETURN                                                          ZDI00025
LN 0026          END                                                             ZDI00026

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZDIGIT

          NO ERRORS
```

STRING

```
LN 0001           SUBROUTINE STRING(IST,ITO,IX)                                  00001
LN 0002     C---- DIE SUBROUTINE **STRING** MUSS NORMALERWEISE DEM PROGRAMM **MAIN**  00002
LN 0003     C---- FOLGEN UND NICHT WIE JETZT VORAUSGEHEN, WENN IN OVERLAY-TECHNIK    00003
LN 0004     C---- GEARBEITET WIRD.                                                 00004
LN 0005           COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    00005
LN 0006          1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST, INEXT,  00006
LN 0007          2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DOUOTE,MAXFIL,  00007
LN 0008          3IRC,IWC,NSTEND,IEXPO,IREGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,  00008
LN 0009          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     00009
LN 0010           COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    00010
LN 0011          1NIFMAX,INTZEI                                                   00011
LN 0012           COMMON// CARDT(80),MERKER(26,2),CARP(140),                      00012
LN 0013          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    00013
LN 0014          1IRET(20),XXX(4),NFILE(25,3)                                     00014
LN 0015           COMMON// ISTLST(340),LISTST(340)                                00015
LN 0016           COMMON// DATAN(330)                                          ***
LN 0017           COMMON// DATA(3700)                                             00017
LN 0018           DIMENSION IPROG(3700)                                           00018
LN 0019           EQUIVALENCE (DATA(1),IPROG(1))                                  00019
LN 0020           INTZEI=48                                                       00020
LN 0021           IK=0                                                            00021
LN 0022           IF((ITO-IST).LE.4) GOTO 584                                     00022
LN 0023           WRITE(IWC,583)                                                  00023
LN 0024      583  FORMAT(20X,79HSTRING CONSTANT HAS MORE THAN FIVE CHARACTERS -- I O  00024
LN 0025          1NLY TAKE THE LEFT MOST FIVE)                                    00025
LN 0026           IK=8                                                            00026
LN 0027      584  IX=0                                                            00027
LN 0028           K=0                                                             00028
LN 0029           DO 582 I=1,5                                                    00029
LN 0030           LOCN=IST-1+I                                                    00030
LN 0031           IF(LOCN.GT.ITO) GOTO 587                                        00031
LN 0032           CALL ZALPH(CARDP(LOCN),L)                                       00032
LN 0033           IF(L.LE.INTZEI) GOTO 581                                        00033
LN 0034           IK=9                                                            00034
LN 0035           WRITE(IWC,585) CARDP(LOCN)                                      00035
LN 0036      585  FORMAT(20X,38HUNKNOWN CHARACTER IN STRING CONSTANT *,A1,25H* I REP  00036
LN 0037          1LACE IT BY A BLANK)                                             00037
LN 0038      587  L=38                                                            00038
LN 0039           GOTO 588                                                    ***
LN 0040      581  IF(L.EQ.38) L=45                                            ***
LN 0041      588  IX=IX+((L-1)*(INTZEI**K))                                   ***
LN 0042           K=K+1                                                           00040
LN 0043      582  CONTINUE                                                        00041
LN 0044           IF(IK.NE.0) WRITE(IWC,586) IK                                   00042
LN 0045      586  FORMAT(1H+,110X,18H***WARNING NUMBER=,I3/11X,28HSFE BASIC TEXTBOOK  00043
LN 0046          1  A  322.2/)                                                    00044
LN 0047           RETURN                                                          00045
LN 0048           END                                                             00046
```

```
     USASI FORTRAN DIAGNOSTIC RESULTS FOR STRING

          NO ERRORS
```

ZALPH

```
LN 0001          SUBROUTINE ZALPH(ZEICH,IA)                                    ZAL00001
LN 0002     C---- DIE SUBROUTINE **ZALPH*** MUSS NORMALERWEISE DEM PROGRAMM **MAIN**ZAL00002
LN 0003     C---- FOLGEN UND NICHT WIE JETZT VORAUSGEHEN,WENN IN OVERLAY-TECHNIK    ZAL00003
LN 0004     C---- GEARBEITET WIRD.                                                ZAL00004
LN 0005          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,     ZAL00005
LN 0006         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT, ZAL00006
LN 0007         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZAL00007
LN 0008         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZAL00008
LN 0009         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZAL00009
LN 0010          COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     ZAL00010
LN 0011         1NIFMAX,INTZEI                                                     ZAL00011
LN 0012          COMMON// CARDT(80),MERKER(26,2),CARP(140),                        ZAL00012
LN 0013         3ALPH(48),BUFFER(40),CARD(80),CAROP(80),DIGIT(10),IFOR(20,2),      ZAL00013
LN 0014         1IRET(20),XXX(4),NFILE(25,3)                                       ZAL00014
LN 0015          COMMON// ISTLST(340),LISTST(340)                                  ZAL00015
LN 0016          COMMON// DATAN(330)                                               ***
LN 0017          COMMON// DATA(3700)                                               ZAL00017
LN 0018          DIMENSION IPROG(3700)                                             ZAL00018
LN 0019          EQUIVALENCE (DATA(1),IPROG(1))                                    ZAL00019
LN 0020          DO 10 I=1,48                                                      ZAL00020
LN 0021          IF(ZEICH.EQ.ALPH(I)) GOTO 20                                      ZAL00021
LN 0022       10 CONTINUE                                                          ZAL00022
LN 0023          I=49                                                              ZAL00023
LN 0024       20 IA=I                                                              ZAL00024
LN 0025          RETURN                                                            ZAL00025
LN 0026          END                                                               ZAL00026

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZALPH

         NO ERRORS
```

FINDFI

```
LN 0001          SUBROUTINE FINDFI(IX,IA)                                    FIN00001
LN 0002     C---- DIE SUBROUTINE **FINDFI** MUSS NORMALERWEISE DEM PROGRAMM **MAIN**FIN00002
LN 0003     C---- FOLGEN UND NICHT WIE JETZT VORAUSGEHEN,WENN IN OVERLAY-TECHNIK    FIN00003
LN 0004     C---- GEARBEITET WIRD.                                              FIN00004
LN 0005          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   FIN00005
LN 0006         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST,INEXT, FIN00006
LN 0007         2NUMBUF,PAPLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL, FIN00007
LN 0008         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, FIN00008
LN 0009         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZET  FIN00009
LN 0010          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,   FIN00010
LN 0011         1NIFMAX,INTZEI                                                  FIN00011
LN 0012          COMMON// CARDT(80),MERKER(26,2),CARP(140),                     FIN00012
LN 0013         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),   FIN00013
LN 0014         1IRET(20),XXX(4),NFILE(25,3)                                    FIN00014
LN 0015          COMMON// ISTLST(340),LISTST(340)                               FIN00015
LN 0016          COMMON// DATAN(330)                                            ***
LN 0017          COMMON// DATA(3700)                                            FIN00017
LN 0018          DIMENSION IPROG(3700)                                          FIN00018
LN 0019          EQUIVALENCE (DATA(1),IPROG(1))                                 FIN00019
LN 0020          DO 10 K=1,NUMFIL                                               FIN00020
LN 0021          IF(NFILE(K,1).EQ.IX) GOTO 11                                   FIN00021
LN 0022       10 CONTINUE                                                       FIN00022
LN 0023          K=NUMFIL+1                                                     FIN00023
LN 0024       11 IA=K                                                           FIN00024
LN 0025          RETURN                                                         FIN00025
LN 0026          END                                                            FIN00026

      USASI FORTRAN DIAGNOSTIC RESULTS FOR FINDFI

           NO ERRORS
```

```
LN 0001    C*****MAIN ROUTINE ---- BASIC INTERPRETIVE COMPILER                    MAI00001
LN 0002    C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   MAI00002
LN 0003    C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*      MAI00003
LN 0004    C---- GEARBEITET WIRD.                                                 MAI00004
LN 0005    C---- (SIEHE BEMERKUNGEN IM HAUPTELEMENT DER *OVERLAY-TECHNIK*)        MAI00005
LN 0006          PROGRAM COMPIL                                                   MAI00006
LN 0007    C**** MAIN --HAUPTPROGRAMM OHNE *OVERLAY-TECHNIK.                      MAI00007
LN 0008          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,     MAI00008
LN 0009         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,MAI00009
LN 0010         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  MAI00010
LN 0011         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, MAI00011
LN 0012         4NPRUS,NCARD,MAXIMA,PUCO,OOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     MAI00012
LN 0013          COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     MAI00013
LN 0014         1NIFMAX,INTZEI                                                   MAI00014
LN 0015          COMMON// CARDT(80),MERKER(26,2),CARP(140),                      MAI00015
LN 0016         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    MAI00016
LN 0017         1IRET(20),XXX(4),NFILE(25,3)                                     MAI00017
LN 0018          COMMON// ISTLST(340),LISTST(340)                                MAI00018
LN 0019          COMMON// DATAN(330)                                             MAI00019
LN 0020          COMMON// DATA(3700)                                             MAI00020
LN 0021          DIMENSION IPROG(3700)                                           MAI00021
LN 0022          EQUIVALENCE (DATA(1),IPROG(1))                                  MAI00022
LN 0023    C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   MAI00023
LN 0024    C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*      MAI00024
LN 0025    C---- GEARBEITET WIRD.                                                 MAI00025
LN 0026          OVERLAY(1)                                                      MAI00026
LN 0027    C                                                                     MAI00027
LN 0028    C*****INITIALIZE                                                      MAI00028
LN 0029       2  CALL ZINITL                                                     MAI00029
LN 0030    C                                                                     MAI00030
LN 0031    C*****HEAD PAGE                                                       MAI00031
LN 0032          WRITE(IWC,45)                                                   MAI00032
LN 0033     45   FORMAT(1H ,40X,54HTESTCOMPILER -- BASIC BWL 5 GIESSEN -- VERSION 6MAI00033
LN 0034         1/76-04/)                                                        MAI00034
LN 0035    C                                                                     MAI00035
LN 0036    C*****READ A CARD                                                     MAI00036
LN 0037    C**** CLEAR CARD AND CARDT                                            MAI00037
LN 0038     50   DO 57 I=1,80                                                    MAI00038
LN 0039          CARD(I)=BLANK                                                   MAI00039
LN 0040          CARDT(I)=BLANK                                                  MAI00040
LN 0041     57   CONTINUE                                                        MAI00041
LN 0042          MCOM=0                                                          MAI00042
LN 0043     6100 READ(IRC,51) CARD                                               MAI00043
LN 0044    C*-*-                                                                 MAI00044
LN 0045    C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(IRC)-          MAI00045
LN 0046    C*-*-      ----CHECK IT----        ----CHECK IT----                   MAI00046
LN 0047    C*-*-                                                                 MAI00047
LN 0048          IF(IFEOF(IRC).EQ.-1) GOTO 8210                                  MAI00048
LN 0049     51   FORMAT(80A1)                                                    MAI00049
LN 0050          IF(CARD(1).NE.ASTRSK) GOTO 3                                    MAI00050
LN 0051          IF(MCOM.NE.1) WRITE(IWC,53) CARD                                MAI00051
LN 0052    C----AUF DIE FOLGENDEN ZEILEN LN=0052-0059 WIRD IM TEXT BEZUG GENOMMENvvMAI00052
LN 0053     53   FORMAT(1HO/////25X,20HINFORMATIONEN DURCH*//25X,29HPROF.DR.OEC.PUBMAI00053
LN 0054         1L.K.WEBER,M.S./25X,36HOIPL.-ING.,DIPL.-OEC.C.W.TUERSCHMANN/25X,41HMAI00054
LN 0055         2PROFESSUR FUER BETRIEBSWIRTSCHAFTSLEHRE V/25X,17HLICHER STRASSE 74MAI00055
LN 0056         3/25X,14HD-6300 GIESSEN/25X,27HFEDERAL REPUBLIC OF GERMANY////25X,3MAI00056
LN 0057         42(1H*),17H PROGRAMMKENNUNG ,31(1H*)//25X,80A1//25X,80(1H*)//25X,84MAI00057
LN 0058         5HBASIC TEXTBOOK = WEBER, TUERSCHMANN. BASIC LEHR- UND HANDBUCH. BEMAI00058
LN 0059         6RN 1977. BAND 1 = A/25X,16(1H*),58X,10HBAND 2 = B/42X,38HWEBER, TU***
LN 0060         7ERSCHMANN. FOSBIC. BERN 1977.,26X,3H= C)                        ***
LN 0061          GOTO 2                                                          MAI00061
```

```
LN 0062     C                                                                   MAIC0060
LN 0063      3    WRITE(IWC,52) CARD                                            MAI00061
LN 0064      52   FORMAT(20X,80A1)                                              MAI00062
LN 0065     C**** READ THROUGH CARDS UNTIL ASTRSK IS FOUND IF MCOM=1            MAI00063
LN 0066           IF(MCOM.EQ.1) GOTO 6100                                       MAI00064
LN 0067     C                                                                   MAI00065
LN 0068     C*****CHECK STORAGE                                                 MAI00066
LN 0069           IF(INEXT.LT.INREG) GOTO 60                                    MAI00067
LN 0070           NN=0                                                          MAI00068
LN 0071           NERROR=1                                                      MAI00069
LN 0072           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                            MAI00070
LN 0073           NERRS = NERRS + 1                                             MAI00071
LN 0074           INREG=NCELLP                                                  MAI00072
LN 0075     C                                                                   MAI00073
LN 0076     C*****COMPRESS CARD                                                 MAI00074
LN 0077      60   LNGCRP=0                                                      MAI00075
LN 0078           NDOL=0                                                        MAI00076
LN 0079           ITOT=1                                                        MAI00077
LN 0080           DO 110 I=1,80                                                 MAI00078
LN 0081           CARDP(I)=BLANK                                                MAI00079
LN 0082           IF(CARD(I).EQ.DDOPU) CARD(I)=DOPU                             MAI00080
LN 0083           IF(CARD(I).EQ.DQUOTE) CARD(I)=QUOTE                           MAI00081
LN 0084           IF(ITOT.EQ.-1) GOTO 66                                        MAI00082
LN 0085           IF(NDOL.EQ.1) GOTO 65                                         MAI00083
LN 0086           IF(CARD(I).EQ.XNULL) NDOL=2                                   MAI00084
LN 0087           IF(NDOL.EQ.2) GOTO 110                                        MAI00085
LN 0088           IF(CARD(I).EQ.DOPU) NDOL=1                                    MAI00086
LN 0089      66   IF(CARD(I).EQ.QUOTE) ITOT=-ITOT                               MAI00087
LN 0090           IF((CARD(I).EQ.BLANK).AND.(ITOT.EQ.1)) GOTO 110               MAI00088
LN 0091      65   LNGCRP=LNGCRP+1                                               MAI00089
LN 0092           CARDP(LNGCRP)=CARD(I)                                         MAI00090
LN 0093      110  CONTINUE                                                      MAI00091
LN 0094     C                                                                   MAI00092
LN 0095     C*****CHECK FOR A BLANK CARD                                        MAI00093
LN 0096           IF(LNGCRP.EQ.0) GOTO 50                                       MAI00094
LN 0097     C                                                                   MAI00095
LN 0098     C*****LOAD STATEMENT NUMBER                                         MAI00096
LN 0099           ISTNO=0                                                       MAI00097
LN 0100           DO 120 I=1,LNGCRP                                             MAI00098
LN 0101           CALL ZDIGIT(CARDP(I),J)                                       MAI00099
LN 0102           IF(J.GT.10) GOTO 130                                          MAI00100
LN 0103      120  ISTNO=(ISTNO*10)+(J-1)                                        MAI00101
LN 0104      130  IBEGST=I                                                     MAI00102
LN 0105           IF(IBEGST.GE.LNGCRP) GOTO 50                                  MAI00103
LN 0106     C                                                                   MAI00104
LN 0107     C**** CHECK FOR COMMENT                                             MAI00105
LN 0108           IF((CARDP(IBEGST+4).EQ.ALPH(5)).AND.(CARDP(IBEGST+5).EQ.ALPH(14)) MAI00106
LN 0109          1.AND.(CARDP(IBEGST+6).EQ.ALPH(20))) GOTO 50                   MAI00107
LN 0110           IF((CARDP(IBEGST).EQ.ALPH(18)).AND.(CARDP(IBEGST+1).EQ.ALPH(5)) MAI00108
LN 0111          1.AND.(CARDP(IBEGST+2).EQ.ALPH(13))) GOTO 140                  MAI00109
LN 0112           IF(ITOT.EQ.-1) GOTO 515                                       MAI00110
LN 0113      140  IF(ISTNO.LE.ISTMAX) GOTO 200                                  MAI00111
LN 0114           NERROR=2                                                      MAI00112
LN 0115      9999 NN=0                                                          MAI00113
LN 0116           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                            MAI00114
LN 0117           NERRS=NERRS+1                                                 MAI00115
LN 0118           GOTO 50                                                       MAI00116
LN 0119     C                                                                   MAI00117
LN 0120     C*****LOAD INTO LISTST                                              MAI00118
LN 0121      200  IF(ISTNO.EQ.0) GOTO 210                                       MAI00119
LN 0122           NSTLST=NSTLST+1                                               MAI00120
```

```
LN 0123          IF(NSTLST.LT.NSTEND) GOTO 9000                                 MAI00121
LN 0124          NN=0                                                           MAI00122
LN 0125          NERROR=24                                                      MAI00123
LN 0126          CALL COMERR(NERROR,NSTEND,I2,X1,X2,NN)                         MAI00124
LN 0127          MCOM=1                                                         MAI00125
LN 0128          GOTO 6100                                                      MAI00126
LN 0129     9000 LISTST(NSTLST)=ISTNO                                           MAI00127
LN 0130          ISTLST(NSTLST)=INPEG                                           MAI00128
LN 0131     C                                                                   MAI00129
LN 0132     C*********************                                             MAI00130
LN 0133     C*****TRANSLATE                                                     MAI00131
LN 0134     C**** SEARCH FOR AN IMAGE STATEMENT                                 MAI00132
LN 0135      210  IF(CARDP(IBEGST).NE.DOPU) GOTO 215                           MAI00133
LN 0136           IF(ISTNO.GT.0) GOTO 201                                       MAI00134
LN 0137           NERROR=25                                                     MAI00135
LN 0138           GOTO 9999                                                     MAI00136
LN 0139      201  IBEGST=IBEGST+1                                               MAI00137
LN 0140           LOC=0                                                         MAI00138
LN 0141           DO 203 I=IBEGST,80                                            MAI00139
LN 0142           LOC=LOC+1                                                     MAI00140
LN 0143           CARDT(LOC)=CARDP(I)                                           MAI00141
LN 0144      203  CONTINUE                                                      MAI00142
LN 0145           DO 206 I=1,80                                                 MAI00143
LN 0146           LOC=80-I+1                                                    MAI00144
LN 0147           IF(CARDT(LOC).EQ.DOPU) GOTO 202                               MAI00145
LN 0148           IF(CARDT(LOC).EQ.BLANK) GOTO 206                              MAI00146
LN 0149           CARDT(LOC+1)=DOPU                                             MAI00147
LN 0150           GOTO 202                                                      MAI00148
LN 0151      206  CONTINUE                                                      MAI00149
LN 0152      202  IF(NZIM.LE.MAXIMA) GOTO 205                                   MAI00150
LN 0153           NERROR=26                                                     MAI00151
LN 0154           GOTO 9999                                                     MAI00152
LN 0155      205  ISTLST(NSTLST)=NZIM+NOFLLP                                    MAI00153
LN 0156           WRITE(NIMAGE,204) ISTNO,(CARDT(I),I=1,80)                     MAI00154
LN 0157      204  FORMAT(I4,80A1)                                               MAI00155
LN 0158           NZIM=NZIM+1                                                   MAI00156
LN 0159           GOTO 50                                                       MAI00157
LN 0160     C**** SEARCH FOR A FILE COMMAND                                     MAI00158
LN 0161      215  NZ=0                                                          MAI00159
LN 0162     C**** COMMAND IS OPEN                                               MAI00160
LN 0163           IF((CARDP(IBEGST).EQ.ALPH(15)).AND.(CARDP(IBEGST+1).EQ.ALPH(16))) MAI00161
LN 0164          1 NZ=1                                                         MAI00162
LN 0165     C**** COMMAND IS PUT                                                MAI00163
LN 0166           IF((CARDP(IBEGST+1).EQ.ALPH(21)).AND.(CARDP(IBEGST+2).EQ.ALPH(20))MAI00164
LN 0167          1) NZ=2                                                        MAI00165
LN 0168     C**** COMMAND IS GET                                                MAI00166
LN 0169           IF((CARDP(IBEGST).EQ.ALPH(7)).AND.(CARDP(IBEGST+1).EQ.ALPH(5)))  MAI00167
LN 0170          1 NZ=3                                                         MAI00168
LN 0171     C**** COMMAND IS RESET                                              MAI00169
LN 0172           IF((CARDP(IBEGST).EQ.ALPH(18)).AND.(CARDP(IBEGST+2).EQ.ALPH(19))  MAI00170
LN 0173          1.AND.(CARDP(IBEGST+3).EQ.ALPH(5))) NZ=4                       MAI00171
LN 0174     C**** COMMAND IS CLOSE                                              MAI00172
LN 0175           IF((CARDP(IBEGST).EQ.ALPH(3)).AND.(CARDP(IBEGST+1).EQ.ALPH(12)))  MAI00173
LN 0176          1 NZ=5                                                         MAI00174
LN 0177     C**** COMMAND IS COMMON-FILE                                        MAI00175
LN 0178           IF((CARDP(IBEGST).EQ.ALPH(3)).AND.(CARDP(IBEGST+7).EQ.ALPH(6)))  MAI00176
LN 0179          1 NZ=6                                                         MAI00177
LN 0180           IF(CARDP(IBEGST+1).EQ.EQUALS) GOTO 213                        MAI00178
LN 0181           IF(NZ.EQ.0) GOTO 213                                          MAI00179
LN 0182     C----AUF DIE FOLGENDEN ZEILEN LN=0183-0184 WIRD IM TEXT BEZUG GENOMMEN**********
LN 0183           CALL ZFILE(NZ)                                                MAI00180
```

```
LN 0184           GOTO 50                                                        MAI00181
LN 0185      213  IF((CARDP(IBEGST).NE.ALPH(13)).OR.(CARDP(IBEGST+1).NE.ALPH(1)).OR.MAI00182
LN 0186          1(CARDP(IBEGST+2).NE.ALPH(20))) GOTO 212                       MAI00183
LN 0187     C----AUF DIE FOLGENDEN ZEILEN LN=0188-0189 WIRD IM TEXT BEZUG GENOMMENvv***
LN 0188           CALL MATTRA                                                    MAI00184
LN 0189           GOTO 50                                                        MAI00185
LN 0190     C***** COMMAND IS RESTORE                                            MAI00186
LN 0191      212  IF((CARDP(IBEGST).NE.ALPH(18)).OR.(CARDP(IBEGST+1).NE.ALPH(5))    MAI00187
LN 0192          1.OR.(CARDP(IBEGST+2).NE.ALPH(19)).OR.(CARDP(IBEGST+3).NE.ALPH(20))MAI00188
LN 0193          2) GOTO 3212                                                   MAI00189
LN 0194           IF(LNGCRP.LT.IBEGST+6) GOTO 3190                              MAI00190
LN 0195           IF(CARDP(IBEGST+7).NE.BLANK) GOTO 3170                        MAI00191
LN 0196           FNUM=0.                                                       MAI00192
LN 0197           GOTO 3150                                                     MAI00193
LN 0198      3190 NERROR=3                                                      MAI00194
LN 0199           GOTO 9999                                                     MAI00195
LN 0200      3150 IPROG(INREG)=-22                                              MAI00196
LN 0201           IPROG(INREG-1)=-22                                            MAI00197
LN 0202           IPROG(INREG-2)=FNUM                                           MAI00198
LN 0203           INREG=INREG-3                                                 MAI00199
LN 0204           GOTO 50                                                       MAI00200
LN 0205      3170 NZ=NERRS                                                      MAI00201
LN 0206           CALL ZTRANX(IBEGST+7,LNGCRP)                                  MAI00202
LN 0207           IF(NERRS.GT.NZ) GOTO 3190                                     MAI00203
LN 0208           IPROG(INREG)=-22                                              MAI00204
LN 0209           INREG=INREG-1                                                 MAI00205
LN 0210           GOTO 50                                                       MAI00206
LN 0211      3212 IF(CARDP(IBEGST).NE.ALPH(12)) GOTO 300                        MAI00207
LN 0212     C*****COMMAND IS LET                                                MAI00208
LN 0213           IF(CARDP(IBEGST+2).NE.ALPH(20)) GOTO 2000                     MAI00209
LN 0214           IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000                      MAI00210
LN 0215     C                                                                   MAI00211
LN 0216     C*****LOOK FOR EQUAL SIGN                                           MAI00212
LN 0217           DO 220 LOC=IBEGST,LNGCRP                                      MAI00213
LN 0218           IF(CARDP(LOC).EQ.EQUALS) GOTO 230                             MAI00214
LN 0219      220  CONTINUE                                                      MAI00215
LN 0220           NERROR=4                                                      MAI00216
LN 0221           GOTO 9999                                                     MAI00217
LN 0222     C                                                                   MAI00218
LN 0223      230  IF((LOC.GT.IBEGST+3).AND.(LOC.LT.LNGCRP)) GOTO 240           MAI00219
LN 0224           NERROR=5                                                      MAI00220
LN 0225           GOTO 9999                                                     MAI00221
LN 0226     C                                                                   MAI00222
LN 0227     C*****SET UP EXPRESSION ON RIGHT                                    MAI00223
LN 0228     240   ITOT=0                                                        MAI00224
LN 0229           IF(CARDP(LOC+1).NE.QUOTE) GOTO 241                            MAI00225
LN 0230           ITOT=1                                                        MAI00226
LN 0231           IPROG(INREG)=-1                                               MAI00227
LN 0232           IPROG(INREG-1)=2                                              MAI00228
LN 0233           IPROG(INREG-2)=-20                                            MAI00229
LN 0234           CALL STRING(LOC+2,LNGCRP-1,IX)                                MAI00230
LN 0235           IPROG(INREG-3)=IX                                             MAI00231
LN 0236           INREG=INREG-4                                                 MAI00232
LN 0237           GOTO 244                                                      MAI00233
LN 0238      241  LNGXX=LNGCRP                                                  MAI00234
LN 0239           IF((CARDP(LOC+2).EQ.DOLSGN).OR.(CARDP(LOC+3).EQ.DOLSGN)) ITOT=1   MAI00235
LN 0240           CALL ZTRANX(LOC+1,LNGXX)                                      MAI00236
LN 0241     C*****CHECK FOR SUBSCRIPTED VARIABLE                                MAI00237
LN 0242      244  IF((CARDP(IBEGST+4).EQ.DOLSGN).OR.(CARDP(IBEGST+5).EQ.DOLSGN))    MAI00238
LN 0243          1 ITOT=ITOT-1                                                 MAI00239
LN 0244           IF(CARDP(LOC-1).NE.PARRT) GOTO 260                            MAI00240
```

```
LN 0245     C*****SUBSCRIPTED VARIABLE FOUND                                                    MAI00241
LN 0246           NZ=NERRS                                                                      MAI00242
LN 0247           CALL ZTRANX(IPEGST+3,LOC-1)                                                   MAI00243
LN 0248           IF(NEPRS.GT.NZ) GOTO 50                                                       MAI00244
LN 0249           IF(IPROG(INREG+1).EQ.-8) GOTO 255                                             MAI00245
LN 0250     C*****ILLEGAL EXPRESSION ON LEFT                                                    MAI00246
LN 0251      245  NERROR=6                                                                     MAI00247
LN 0252           GOTO 9999                                                                     MAI00248
LN 0253     C                                                                                   MAI00249
LN 0254     C*****SUBSCRIPTED VARIABLE ON LEFT IS OK                                            MAI00250
LN 0255      255  IPROG(INREG+1)=-19                                                           MAI00251
LN 0256           GOTO 242                                                                      MAI00252
LN 0257     C                                                                                   MAI00253
LN 0258     C*****UNSUBSCRIPTED VARIABLE ON LEFT -- FIND LETTER                                 MAI00254
LN 0259      260  CALL ZALPH(CARDP(IBEGST+3),K)                                                MAI00255
LN 0260     C*****NON-ALPHA CHARACTER FOUND                                                     MAI00256
LN 0261           IF(K.GT.26) GOTO 245                                                          MAI00257
LN 0262           IF(LOC.EQ.IBEGST+4) GOTO 290                                                  MAI00258
LN 0263           IF(LOC.EQ.IBEGST+5) GOTO 275                                                  MAI00259
LN 0264           IF(LOC.EQ.IBEGST+6) GOTO 275                                                  MAI00260
LN 0265           GOTO 245                                                                      MAI00261
LN 0266     C*****TWO-CHARACTER NAME FOUND                                                      MAI00262
LN 0267      275  IF(CARDP(IBEGST+4).EQ.DOLSGN) GOTO 290                                       MAI00263
LN 0268           CALL ZDIGIT(CARDP(IBEGST+4),L)                                                MAI00264
LN 0269           IF(L.GT.10) GOTO 245                                                          MAI00265
LN 0270     C                                                                                   MAI00266
LN 0271           IPROG(INREG)=-8                                                               MAI00267
LN 0272           IPROG(INREG-1)=K+(26*(L-1))+53                                                MAI00268
LN 0273           INREG=INREG-2                                                                 MAI00269
LN 0274      242  IF(ITOT.EQ.0) GOTO 50                                                        MAI00270
LN 0275           NN=1                                                                          MAI00271
LN 0276           NERROR=4                                                                      MAI00272
LN 0277           IF(ITOT.EQ.-1) NERROR=5                                                       MAI00273
LN 0278           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                                            MAI00274
LN 0279           GOTO 50                                                                       MAI00275
LN 0280     C                                                                                   MAI00276
LN 0281     C     ONE-CHARACTER NAME FOUND                                                      MAI00277
LN 0282      290  IPROG(INREG)=-9                                                              MAI00278
LN 0283           IPROG(INREG-1)=K                                                              MAI00279
LN 0284           INREG=INREG-2                                                                 MAI00280
LN 0285           GOTO 242                                                                      MAI00281
LN 0286     C                                                                                   MAI00282
LN 0287      300  IF((CARDP(IBEGST).NE.ALPH(18)).OR.(CARDP(IBEGST+2).NE.ALPH(1)))              MAI00283
LN 0288          1 GOTO 305                                                                     MAI00284
LN 0289           IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000                                      MAI00285
LN 0290     C     COMMAND IS READ                                                               MAI00286
LN 0291           IX=0                                                                          MAI00287
LN 0292           NZ=9                                                                          MAI00288
LN 0293           MCOM=-7                                                                       MAI00289
LN 0294           CALL ZLISTE(IBEGST+4,MCOM,IX,NZ)                                              MAI00290
LN 0295           GOTO 50                                                                       MAI00291
LN 0296     C**** COMMAND IS INPUT                                                              MAI00292
LN 0297      305  IF((CARDP(IBEGST).NE.ALPH(9)).OR.(CARDP(IBEGST+1).NE.ALPH(14)))            MAI00293
LN 0298          1 GOTO 400                                                                     MAI00294
LN 0299           IX=0                                                                          MAI00295
LN 0300           NZ=9                                                                          MAI00296
LN 0301           MCOM=-45                                                                      MAI00297
LN 0302           CALL ZLISTE(IBEGST+5,MCOM,IX,NZ)                                              MAI00298
LN 0303           GOTO 50                                                                       MAI00299
LN 0304     C                                                                                   MAI00300
LN 0305      400  IF((CARDP(IBEGST).NE.ALPH(16)).OR.(CARDP(IBEGST+1).NE.ALPH(1)))             MAI00301
```

```
LN 0306          1 GOTO 500                                                    MAI00302
LN 0307     C*****COMMAND IS PAGE                                              MAI00303
LN 0308           IF(CARDP(IBEGST+3).NE.ALPH(5)) GOTO 2000                     MAI00304
LN 0309           IPROG(INREG)=-13                                             MAI00305
LN 0310           INREG=INREG-1                                                MAI00306
LN 0311           GOTO 50                                                      MAI00307
LN 0312     C                                                                  MAI00308
LN 0313      500  IF((CARDP(IBEGST).NE.ALPH(16)).OR.(CARDP(IBEGST+1).NE.ALPH(18)))MAI00309
LN 0314          1 GOTO 600                                                    MAI00310
LN 0315     C*****COMMAND IS PRINT                                             MAI00311
LN 0316           IF(CARDP(IBEGST+4).NE.ALPH(20)) GOTO 2000                    MAI00312
LN 0317     C*****CHECK FOR A LINE SKIP                                        MAI00313
LN 0318           IF(LNGCRP.GT.IBEGST+4) GOTO 505                              MAI00314
LN 0319           IPROG(INREG)=-10                                             MAI00315
LN 0320           INREG=INREG-1                                                MAI00316
LN 0321           GOTO 50                                                      MAI00317
LN 0322     C                                                                  MAI00318
LN 0323     C*****CHECK FOR A #PRINT ALL#                                      MAI00319
LN 0324      505  IF(LNGCRP.NE.(IBEGST+7)) GOTO 510                           MAI00320
LN 0325           IF(CARDP(IBEGST+5).NE.ALPH(1)) GOTO 510                      MAI00321
LN 0326           IF(CARDP(IBEGST+6).NE.ALPH(12)) GOTO 510                     MAI00322
LN 0327           IF(CARDP(IBEGST+7).NE.ALPH(12)) GOTO 510                     MAI00323
LN 0328     C*****COMMAND IS PRINT ALL                                         MAI00324
LN 0329           IPROG(INREG)=-20                                             MAI00325
LN 0330           INREG=INREG-1                                                MAI00326
LN 0331           GOTO 50                                                      MAI00327
LN 0332     C                                                                  MAI00328
LN 0333      516  NERROR=7                                                     MAI00329
LN 0334           GOTO 9999                                                    MAI00330
LN 0335     C                                                                  MAI00331
LN 0336     C*****SET UP EXPRESSION                                            MAI00332
LN 0337      510  IQTLOC=0                                                     MAI00333
LN 0338           IQT=0                                                        MAI00334
LN 0339           NPRI=0                                                       MAI00335
LN 0340           ICONT=NPRI                                                   MAI00336
LN 0341           LOC=0                                                        MAI00337
LN 0342           IF((CARDP(IBEGST+5).NE.ALPH(21)).OR.(CARDP(IBEGST+6).NE.ALPH(19)))MAI00338
LN 0343          1 GOTO 519                                                    MAI00339
LN 0344     C**** COMMAND IS PRINT USING                                       MAI00340
LN 0345     C**** COMMAND CODES FOR STANDARD PRINT ARE                         MAI00341
LN 0346     C****    NUMERIC VARIABLE OR EXPRESSION=-17                        MAI00342
LN 0347     C****    ALPHANUMERIC VARIABLE=-30                                 MAI00343
LN 0348     C****    COMMA=-67                                                 MAI00344
LN 0349     C****    PUCO=-69                                                  MAI00345
LN 0350     C****    END OF LINE=-18                                           MAI00346
LN 0351     C****    ALPHANUMERIC CONSTANT=-16                                 MAI00347
LN 0352     C**** COMMAND CODES FOR PRINT USING STATEMENT ARE                  MAI00348
LN 0353     C****    IMAGE STATEMENT NUMBER=-62                                MAI00349
LN 0354     C****    NUMERIC VARIABLE OR EXPRESSION=-64                        MAI00350
LN 0355     C****    ALPHANUMERIC VARIABLE=-66                                 MAI00351
LN 0356     C****    COMMA=-67                                                 MAI00352
LN 0357     C****    PUCO=-69                                                  MAI00353
LN 0358     C****    END OF LINE=-65                                           MAI00354
LN 0359     C****    ALPHANUMERIC CONSTANT=-63                                 MAI00355
LN 0360          IBEGST=IBEGST+10                                              MAI00356
LN 0361          IF(IBEGST.LE.LNGCRP) GOTO 563                                 MAI00357
LN 0362          NERROR=27                                                     MAI00358
LN 0363          GOTO 9999                                                     MAI00359
LN 0364      563 DO 561 I=IBEGST,LNGCRP                                        MAI00360
LN 0365          IF(CARDP(I).EQ.PUCO) GOTO 562                                 MAI00361
LN 0366          IF(CARDP(I).EQ.COMMA) GOTO 562                                MAI00362
```

```
LN 0367     561  CONTINUE                                                    MAI00363
LN 0368          I=LNGCRP+1                                                  MAI00364
LN 0369     562  CALL ZTRANX(IBEGST,I-1)                                     MAI00365
LN 0370          IPROG(INREG)=-62                                            MAI00366
LN 0371          INREG=INREG-1                                               MAI00367
LN 0372          IQT=47                                                      MAI00368
LN 0373          IQTLOC=36                                                   MAI00369
LN 0374          IFR=I+1                                                     MAI00370
LN 0375          IF(I.LT.LNGCRP) GOTO 525                                    MAI00371
LN 0376          IF(I.GT.LNGCRP) GOTO 564                                    MAI00372
LN 0377          IPROG(INREG)=-70                                            MAI00373
LN 0378          INREG=INREG-1                                               MAI00374
LN 0379          ICONT=-1                                                    MAI00375
LN 0380          IF(CARDP(I).EQ.COMMA) ICONT=1                               MAI00376
LN 0381          LOC=LNGCRP                                                  MAI00377
LN 0382          GOTO 525                                                    MAI00378
LN 0383     564  IPROG(INREG)=-65                                            MAI00379
LN 0384          INREG=INREG-1                                               MAI00380
LN 0385          GOTO 50                                                     MAI00381
LN 0386     519  IFR=IBEGST+5                                                MAI00382
LN 0387    C*****LOOK FOR FIRST QUOTE OR FREE COMMA                          MAI00383
LN 0388     525  ITOT=0                                                      MAI00384
LN 0389          IF(NPRI.EQ.ICONT) GOTO 547                                  MAI00385
LN 0390          IPROG(INREG)=-67                                            MAI00386
LN 0391          IF(ICONT.EQ.-1) IPROG(INREG)=-69                            MAI00387
LN 0392          INREG=INREG-1                                               MAI00388
LN 0393          NPRI=ICONT                                                  MAI00389
LN 0394     547  NDOL=0                                                      MAI00390
LN 0395          IF(LOC.GT.LNGCRP) GOTO 50                                   MAI00391
LN 0396          IF((LOC.EQ.LNGCRP).AND.(CARDP(LOC).EQ.COMMA)) GOTO 50       MAI00392
LN 0397          IF((LOC.EQ.LNGCRP).AND.(CARDP(LOC).EQ.PUCO)) GOTO 50        MAI00393
LN 0398          DO 530 LOC=IFR,LNGCRP                                       MAI00394
LN 0399          IF(CARDP(LOC).EQ.QUOTE) GOTO 545                            MAI00395
LN 0400          IF(CARDP(LOC).NE.DOLSGN) GOTO 531                           MAI00396
LN 0401          NDOL=1                                                      MAI00397
LN 0402          GOTO 530                                                    MAI00398
LN 0403     531  IF(CARDP(LOC).EQ.PAPLFT) ITOT=ITOT+1                        MAI00399
LN 0404          IF(CARDP(LOC).EQ.PARRT) ITOT=ITOT-1                         MAI00400
LN 0405          IF((CARDP(LOC).EQ.COMMA).AND.(ITOT.EQ.0)) GOTO 540          MAI00401
LN 0406          IF((CARDP(LOC).EQ.PUCO).AND.(ITOT.EQ.0)) GOTO 546           MAI00402
LN 0407     530  CONTINUE                                                    MAI00403
LN 0408    C*****REMAINDER OF CARD IS AN EXPRESSION                          MAI00404
LN 0409          LOC=LNGCRP+1                                                MAI00405
LN 0410          IF(IFR.GT.LNGCRP) GOTO 535                                  MAI00406
LN 0411          GOTO 541                                                    MAI00407
LN 0412     535  IPROG(INREG)=-18-IQT                                        MAI00408
LN 0413          INREG=INREG-1                                               MAI00409
LN 0414          GOTO 50                                                     MAI00410
LN 0415     540  ICONT=1                                                     MAI00411
LN 0416          GOTO 541                                                    MAI00412
LN 0417     546  ICONT=-1                                                    MAI00413
LN 0418    C**** COMMAND IS PRINT TAB(EXPRESSION)                            MAI00414
LN 0419     541  IF((CARDP(IFR).NE.ALPH(20)).OR.(CARDP(IFR+1).NE.ALPH(1)).OR.(CARDPMAI00415
LN 0420         1(IFR+2).NE.ALPH(2))) GOTO 542                               MAI00416
LN 0421          CALL ZTRANX(IFR+3,LOC-1)                                    MAI00417
LN 0422          IPROG(INREG)=-68                                            MAI00418
LN 0423          ICONT=-1                                                    MAI00419
LN 0424          IFR=LOC+1                                                   MAI00420
LN 0425          GOTO 526                                                    MAI00421
LN 0426     542  CALL ZTRANX(IFR,LOC-1)                                      MAI00422
LN 0427          IPROG(INREG)=-17-IQT                                        MAI00423
```

```
LN 0428           IF(NDOL.EQ.1) IPROG(INREG)=-30-IQTLOC                  MAI00424
LN 0429     C*****CHECK FOR FINAL COMMA                                  MAI00425
LN 0430      526  INREG=INREG-1                                          MAI00426
LN 0431           IF(LOC.GT.LNGCRP) GOTO 535                             MAI00427
LN 0432           IF(LOC.EQ.LNGCRP) GOTO 525                             MAI00428
LN 0433     C*****RESET IFR                                              MAI00429
LN 0434           IFR=LOC+1                                              MAI00430
LN 0435           GOTO 525                                               MAI00431
LN 0436     C*****QUOTE FOUND                                            MAI00432
LN 0437      545  IF(LOC.EQ.IFR) GOTO 550                               MAI00433
LN 0438     C*****EXPRESSION TRAPPED WITHOUT COMMA                       MAI00434
LN 0439           NN=1                                                   MAI00435
LN 0440           NERROR=7                                               MAI00436
LN 0441           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                     MAI00437
LN 0442           CALL ZTRANX(IFR,LOC-1)                                 MAI00438
LN 0443           IPROG(INREG)=-17-IQT                                   MAI00439
LN 0444           IF(NDOL.EQ.1) IPROG(INREG)=-30-IQTLOC                  MAI00440
LN 0445           INREG=INREG-1                                          MAI00441
LN 0446     C                                                            MAI00442
LN 0447     C*****FIND NEXT QUOTE                                        MAI00443
LN 0448      550  IFR=LOC+1                                             MAI00444
LN 0449           DO 555 LOC=IFR,LNGCRP                                  MAI00445
LN 0450           IF(CARDP(LOC).EQ.QUOTE) GOTO 560                       MAI00446
LN 0451      555  CONTINUE                                              MAI00447
LN 0452      560  ITO=LOC-1                                             MAI00448
LN 0453     C                                                            MAI00449
LN 0454     C*****SET UP AND LOAD CHARACTERS                             MAI00450
LN 0455           IST=IFR                                                MAI00451
LN 0456      570  ICT=0                                                 MAI00452
LN 0457           IPROG(INREG)=-16-TOT                                   MAI00453
LN 0458           NDOL=INREG-1                                           MAI00454
LN 0459           INREG=INREG-1                                          MAI00455
LN 0460      575  ICT=ICT+1                                             MAI00456
LN 0461           IEND=IST+4                                             MAI00457
LN 0462           IF(IEND.GT.ITO) GOTO 585                               MAI00458
LN 0463     C*****LOAD FIVE CHARACTERS                                   MAI00459
LN 0464           CALL STRING(IST,IEND,IX)                               MAI00460
LN 0465     C*****INSERT IN IPROG                                        MAI00461
LN 0466           LOCN=INREG-ICT                                         MAI00462
LN 0467           IPROG(LOCN)=IX                                         MAI00463
LN 0468     C*****STEP START                                             MAI00464
LN 0469           IST=IST+5                                              MAI00465
LN 0470     C*****CHECK FOR END                                          MAI00466
LN 0471           IF(IST.GT.ITO) GOTO 591                                MAI00467
LN 0472     C*****CHECK FOR THREE WORDS COMPLETED                        MAI00468
LN 0473           IF(ICT.LT.3) GOTO 575                                  MAI00469
LN 0474     C*****THIS IS THE THIRD WORD LOADED                          MAI00470
LN 0475           INREG=INREG-4                                          MAI00471
LN 0476           IPROG(NDOL)=ICT                                        MAI00472
LN 0477           GOTO 570                                               MAI00473
LN 0478     C                                                            MAI00474
LN 0479     C*****PARTIAL WORD FOUND                                     MAI00475
LN 0480      585  CALL STRING(IST,ITO,IX)                               MAI00476
LN 0481           LOCN=INREG-ICT                                         MAI00477
LN 0482           IPROG(LOCN)=IX                                         MAI00478
LN 0483     C                                                            MAI00479
LN 0484     C*****ADD BLANK WORDS IF NECESSARY                           MAI00480
LN 0485      591  IF(ICT.GE.3) GOTO 595                                 MAI00481
LN 0486           IF(CARDP(LOC+1).EQ.PUCO) GOTO 595                      MAI00482
LN 0487           ICT=ICT+1                                              MAI00483
LN 0488           LOCN=INREG-ICT                                         MAI00484
```

```
LN 0489          IPROG(LOCN)=200590357                                          MAI00485
LN 0490          GOTO 591                                                       MAI00486
LN 0491     C                                                                   MAI00487
LN 0492     C*****COMPLETE                                                      MAI00488
LN 0493      595 INREG=INREG-ICT-1                                              MAI00489
LN 0494          IPROG(NDOL)=ICT                                                MAI00490
LN 0495          ICONT=1                                                        MAI00491
LN 0496          IF(CARDP(LOC+1).EQ.PUCO) ICONT=-1                              MAI00492
LN 0497     C                                                                   MAI00493
LN 0498     C*****CHECK TO SEE IF QUOTE WAS LAST CHARACTER                      MAI00494
LN 0499          IF(LOC.EQ.LNGCRP) GOTO 535                                     MAI00495
LN 0500     C*****  SEE IF QUOTE IS FOLLOWED BY A COMMA OR A PUCO               MAI00496
LN 0501          IF((CARDP(LOC+1).EQ.PUCO).OR.(CARDP(LOC+1).EQ.COMMA)) GOTO 597 MAI00497
LN 0502          NN=1                                                           MAI00498
LN 0503          NERROR=7                                                       MAI00499
LN 0504          CALL COMERR(NERROR,I1,I2,X1,X2,NN)                             MAI00500
LN 0505     C*****COMMA MISSING                                                 MAI00501
LN 0506          IFR=LOC+1                                                      MAI00502
LN 0507          LOC=LOC+1                                                      MAI00503
LN 0508          GOTO 525                                                       MAI00504
LN 0509     C*****COMMA PRESENT                                                 MAI00505
LN 0510      597 IFR=LOC+2                                                      MAI00506
LN 0511          LOC=LOC+2                                                      MAI00507
LN 0512          GOTO 525                                                       MAI00508
LN 0513     C                                                                   MAI00509
LN 0514      600 IF((CARDP(IBEGST).NE.ALPH(7)).OR.(CARDP(IBEGST+2).NE.ALPH(20))) MAI00510
LN 0515        1 GOTO 650                                                       MAI00511
LN 0516     C*****COMMAND IS GOTO                                               MAI00512
LN 0517          IF(CARDP(IBEGST+1).NE.ALPH(15)) GOTO 2000                      MAI00513
LN 0518          IST=IBEGST+4                                                   MAI00514
LN 0519     C****  CHECK FOR COMPUTED GOTO OR GOSUB                             MAI00515
LN 0520      1130 DO 611 IX=IST,LNGCRP                                          MAI00516
LN 0521          IF((CARDP(IX).EQ.ALPH(15)).AND.(CARDP(IX+1).EQ.ALPH(14))) GOTO 612MAI00517
LN 0522          IF((CARDP(IX).EQ.ALPH(15)).AND.(CARDP(IX+1).EQ.ALPH( 6))) GOTO 612MAI00518
LN 0523      611 CONTINUE                                                       MAI00519
LN 0524          GOTO 610                                                       MAI00520
LN 0525     C****  COMPUTED GOTO *VERSION* GOTO(STATEMENTNUMBERS)ON/OF(EXPRESSION)  MAI00521
LN 0526     C****  COMPUTED GOSUB *VERSION* GOSUB(STATEMENTNUMBERS)ON/OF(EXPRESSION) MAI00522
LN 0527      612 CALL ZTRANX(IX+2,LNGCRP)                                       MAI00523
LN 0528      653 IPROG(INREG)=-27                                               MAI00524
LN 0529          INREG=INREG-2                                                  MAI00525
LN 0530          MIC=INREG+1                                                    MAI00526
LN 0531          MIV=0                                                          MAI00527
LN 0532          LNGCRP=IX-1                                                    MAI00528
LN 0533      615 DO 613 IX=IST,LNGCRP                                           MAI00529
LN 0534          IF(CARDP(IX).EQ.COMMA) GOTO 614                                MAI00530
LN 0535      613 CONTINUE                                                       MAI00531
LN 0536      614 CALL ZCONVN(IST,IX-1,FNUM)                                     MAI00532
LN 0537          IF(FNUM.LT.1.) GOTO 690                                        MAI00533
LN 0538          IF(FNUM.GT.FLOAT(ISTMAX)) GOTO 690                             MAI00534
LN 0539          MIV=MIV+1                                                      MAI00535
LN 0540          IPROG(INREG)=FNUM                                              MAI00536
LN 0541          INREG=INREG-1                                                  MAI00537
LN 0542          IST=IX+1                                                       MAI00538
LN 0543          IF(IST.LE.LNGCRP) GOTO 615                                     MAI00539
LN 0544          IPROG(MIC)=MIV                                                 MAI00540
LN 0545          GOTO 50                                                        MAI00541
LN 0546     C*****LOAD STATEMENT NUMBER                                         MAI00542
LN 0547      610 IF(IST.GT.LNGCRP) GOTO 690                                     MAI00543
LN 0548          LNGXX=LNGCRP                                                   MAI00544
LN 0549          CALL ZCONVN(IST,LNGXX,STNO)                                    MAI00545
```

```
LN 0550          ISTTO=STNO+.01                                                 MAI00546
LN 0551          IF(ISTTO.LT.1) GOTO 690                                        MAI00547
LN 0552          IF(ISTTO.GT.ISTMAX) GOTO 690                                   MAI00548
LN 0553    C*****COMMAND IS LEGAL                                               MAI00549
LN 0554          IPROG(INREG)=-3                                                MAI00550
LN 0555          IPROG(INREG-1)=ISTTO                                           MAI00551
LN 0556          INREG=INREG-2                                                  MAI00552
LN 0557          GOTO 50                                                        MAI00553
LN 0558    C*****ILLEGAL STATEMENT NUMBER                                       MAI00554
LN 0559     690  NERROR=8                                                       MAI00555
LN 0560          GOTO 9999                                                      MAI00556
LN 0561    C                                                                    MAI00557
LN 0562     650  IF((CARDP(IBEGST).EQ.ALPH(15)).AND.(CARDP(IBEGST+1).EQ.ALPH(6)))MAI00558
LN 0563         1 GOTO 654                                                      MAI00559
LN 0564          IF((CARDP(IBEGST).NE.ALPH(15)).OR.(CARDP(IBEGST+1).NE.ALPH(14)))MAI00560
LN 0565         1 GOTO 700                                                      MAI00561
LN 0566    C**** COMPUTED GOTO *VERSION* ON/OF(EXPRESSION)GOTO(STATEMENTNUMBERS)MAI00562
LN 0567    C**** COMPUTED GOSUB *VERSION* ON/OF(EXPRESSION)GOSUB(STATEMNTNUMBERS)MAI00563
LN 0568      654 IST=IBEGST+2                                                   MAI00564
LN 0569          DO 651 IX=IST,LNGCRP                                           MAI00565
LN 0570          IF((CARDP(IX).EQ.ALPH(7)).AND.(CARDP(IX+1).EQ.ALPH(15)).AND.   MAI00566
LN 0571         1(CARDP(IX+2).EQ.ALPH(20))) GOTO 652                            MAI00567
LN 0572          IF((CARDP(IX).EQ.ALPH(7)).AND.(CARDP(IX+1).EQ.ALPH(15)).AND.   MAI00568
LN 0573         1(CARDP(IX+2).EQ.ALPH(19))) GOTO 655                            MAI00569
LN 0574     651  CONTINUE                                                       MAI00570
LN 0575          GOTO 2000                                                      MAI00571
LN 0576     652  IQT=4                                                          MAI00572
LN 0577     656  CALL ZTRANX(IST,IX-1)                                          MAI00573
LN 0578          IST=IX+IQT                                                     MAI00574
LN 0579          IX=LNGCRP+1                                                    MAI00575
LN 0580          GOTO 653                                                       MAI00576
LN 0581     655  IPROG(INREG)=-44                                               MAI00577
LN 0582          INREG=INREG-1                                                  MAI00578
LN 0583          IQT=5                                                          MAI00579
LN 0584          GOTO 656                                                       MAI00580
LN 0585    C                                                                    MAI00581
LN 0586     700  IF(CARDP(IBEGST).NE.ALPH(9)) GOTO 800                          MAI00582
LN 0587    C*****COMMAND IS IF                                                  MAI00583
LN 0588          IF(CARDP(IBEGST+1).NE.ALPH(6)) GOTO 2000                       MAI00584
LN 0589          IF(LNGCRP.GT.IBEGST+7) GOTO 720                                MAI00585
LN 0590          NERROR=9                                                       MAI00586
LN 0591          GOTO 9999                                                      MAI00587
LN 0592    C                                                                    MAI00588
LN 0593     720  LM1=LNGCRP-1                                                   MAI00589
LN 0594          IBEG=IBEGST+2                                                  MAI00590
LN 0595    C*****LOOK FOR AN EQUAL SIGN                                         MAI00591
LN 0596          ITOT=1                                                         MAI00592
LN 0597          DO 725 LOC=IBEG,LM1                                            MAI00593
LN 0598          IF(CARDP(LOC).EQ.QUOTE) ITOT=-ITOT                             MAI00594
LN 0599          IF(ITOT.EQ.-1) GOTO 725                                        MAI00595
LN 0600          IF(CARDP(LOC).EQ.EQUALS) GOTO 726                              MAI00596
LN 0601          IF(CARDP(LOC).EQ.VLESS) GOTO 728                               MAI00597
LN 0602          IF(CARDP(LOC).EQ.VGREAT) GOTO 724                              MAI00598
LN 0603     725  CONTINUE                                                       MAI00599
LN 0604          GOTO 730                                                       MAI00600
LN 0605     726  ICOMP=5                                                       MAI00601
LN 0606          GOTO 727                                                       MAI00602
LN 0607     724  IF(CARDP(LOC+1).NE.EQUALS) GOTO 723                           MAI00603
LN 0608          ICOMP=2                                                        MAI00604
LN 0609          GOTO 760                                                       MAI00605
LN 0610     723  ICOMP=1                                                        MAI00606
```

```
LN 0611          GOTO 727                                                    MAI00607
LN 0612     728  IF(CARDP(LOC+1).NE.EQUALS) GOTO 722                        MAI00608
LN 0613          ICOMP=4                                                     MAI00609
LN 0614          GOTO 760                                                    MAI00610
LN 0615     722  IF(CARDP(LOC+1).NE.VGREAT) GOTO 721                       MAI00611
LN 0616          ICOMP=6                                                     MAI00612
LN 0617          GOTO 760                                                    MAI00613
LN 0618     721  ICOMP=3                                                     MAI00614
LN 0619     727  IEND1=LOC-1                                                 MAI00615
LN 0620          IBEG2=LOC+1                                                 MAI00616
LN 0621          GOTO 765                                                    MAI00617
LN 0622     C*****LOOK FOR TWO-CHARACTER OPERATOR                            MAI00618
LN 0623      730 DO 750 LOC=IBEG,LM1                                         MAI00619
LN 0624          IF(CARDP(LOC).EQ.QUOTE) ITOT=-ITOT                          MAI00620
LN 0625          IF(ITOT.EQ.-1) GOTO 750                                     MAI00621
LN 0626          CALL ZALPH(CARDP(LOC),K1)                                   MAI00622
LN 0627          IF(K1.GT.26) GOTO 750                                       MAI00623
LN 0628          CALL ZALPH(CARDP(LOC+1),K2)                                 MAI00624
LN 0629          IF(K2.GT.26) GOTO 750                                       MAI00625
LN 0630          CALL ZALPH(CARDP(LOC+2),K3)                                 MAI00626
LN 0631          CALL ZALPH(CARDP(LOC+3),K4)                                 MAI00627
LN 0632          ICOMP=0                                                     MAI00628
LN 0633          IF((K1.EQ.7).AND.(K2.EQ.20)) ICOMP=1                        MAI00629
LN 0634          IF((K1.EQ.12).AND.(K2.EQ.20)) ICOMP=3                       MAI00630
LN 0635          IF((K1.EQ.5 ).AND.(K2.EQ.17)) ICOMP=5                       MAI00631
LN 0636          IF(ICOMP.GT.0) GOTO 760                                     MAI00632
LN 0637          IF(K3.NE.17) GOTO 756                                       MAI00633
LN 0638          IF(K4.GT.26) GOTO 756                                       MAI00634
LN 0639          K1=K2                                                       MAI00635
LN 0640          K2=K3                                                       MAI00636
LN 0641     756  IF((K1.EQ.7).AND.(K2.EQ.5)) ICOMP=2                        MAI00637
LN 0642          IF((K1.EQ.12).AND.(K2.EQ.5)) ICOMP=4                        MAI00638
LN 0643          IF((K1.EQ.14).AND.(K2.EQ.5)) ICOMP=6                        MAI00639
LN 0644          IF(ICOMP.GT.0) GOTO 760                                     MAI00640
LN 0645     750  CONTINUE                                                    MAI00641
LN 0646          NERROR=10                                                   MAI00642
LN 0647          GOTO 9999                                                   MAI00643
LN 0648     C                                                                MAI00644
LN 0649     760  IEND1=LOC-1                                                 MAI00645
LN 0650          IBEG2=LOC+2                                                 MAI00646
LN 0651     765  IF(IBEG2.GT.LM1) GOTO 767                                   MAI00647
LN 0652          DO 766 LOC=IBEG2,LM1                                        MAI00648
LN 0653          IF(CARDP(LOC).EQ.QUOTE) ITOT=-ITOT                          MAI00649
LN 0654          IF(ITOT.EQ.-1) GOTO 766                                     MAI00650
LN 0655          IF((CARDP(LOC).EQ.ALPH(7)).AND.(CARDP(LOC+1).EQ.ALPH(15)))GOTO 780MAI00651
LN 0656          IF((CARDP(LOC).EQ.ALPH(20)).AND.(CARDP(LOC+1).EQ.ALPH(8)))GOTO 780MAI00652
LN 0657     766  CONTINUE                                                    MAI00653
LN 0658     767  NERROR=11                                                   MAI00654
LN 0659          GOTO 9999                                                   MAI00655
LN 0660     C                                                                MAI00656
LN 0661     780  IEND2=LOC-1                                                 MAI00657
LN 0662          ISTST=LOC+4                                                 MAI00658
LN 0663     C*****SET UP FIRST EXPRESSION AND STORE                          MAI00659
LN 0664          MCOM=1                                                      MAI00660
LN 0665          IF(CARDP(IBEG).NE.QUOTE) GOTO 781                           MAI00661
LN 0666          CALL STRING(IBEG+1,IEND1-1,IX)                              MAI00662
LN 0667     784  IPROG(INREG-3)=IX                                          MAI00663
LN 0668          IPROG(INREG)=-1                                             MAI00664
LN 0669          IPROG(INREG-1)=2                                            MAI00665
LN 0670          IPROG(INREG-2)=-20                                          MAI00666
LN 0671          INREG=INREG-4                                               MAI00667
```

```
LN 0672           GOTO 786                                                MAI00668
LN 0673     781   CALL ZTRANX(IBEG,IEND1)                                 MAI00669
LN 0674     786   IPROG(INREG)=-14                                        MAI00670
LN 0675           IPROG(INREG-1)=MCOM                                     MAI00671
LN 0676           INREG=INREG-2                                           MAI00672
LN 0677     C*****SET UP SECOND EXPRESSION AND STORE                      MAI00673
LN 0678           IF(MCOM.EQ.2) GOTO 787                                  MAI00674
LN 0679           MCOM=2                                                  MAI00675
LN 0680           IF(CARDP(IBEG2).NE.QUOTE) GOTO 782                      MAI00676
LN 0681           CALL STRING(IBEG2+1,IEND2-1,IX)                         MAI00677
LN 0682           GOTO 784                                                MAI00678
LN 0683     782   CALL ZTRANX(IBEG2,IEND2)                                MAI00679
LN 0684           GOTO 786                                                MAI00680
LN 0685     C******SET UP COMPARISON                                      MAI00681
LN 0686     787   IPROG(INREG)=-6                                         MAI00682
LN 0687           IPROG(INREG-1)=ICOMP                                    MAI00683
LN 0688           INREG=INREG-2                                           MAI00684
LN 0689     C*****FIND STATEMENT NUMBER                                   MAI00685
LN 0690           IF(ISTST.GT.LNGCRP) GOTO 785                            MAI00686
LN 0691           LNGXX=LNGCRP                                            MAI00687
LN 0692           CALL ZCONVN(ISTST,LNGXX,FNUM)                           MAI00688
LN 0693           IF(FNUM.GT.FLOAT(ISTMAX)) GOTO 690                      MAI00689
LN 0694           IF(FNUM.GT.0.) GOTO 795                                 MAI00690
LN 0695     785   NERROR=12                                               MAI00691
LN 0696           GOTO 9999                                               MAI00692
LN 0697     C*****SET UP TRANSFER                                         MAI00693
LN 0698     795   IPROG(INREG)=-3                                         MAI00694
LN 0699           IPROG(INREG-1)=FNUM+.5                                  MAI00695
LN 0700           INREG=INREG-2                                           MAI00696
LN 0701           GOTO 50                                                 MAI00697
LN 0702     C                                                             MAI00698
LN 0703     800   IF(CARDP(IBEGST).NE.ALPH(6)) GOTO 900                   MAI00699
LN 0704     C*****COMMAND IS FOR                                          MAI00700
LN 0705           IF(CARDP(IBEGST+2).NE.ALPH(18)) GOTO 2000               MAI00701
LN 0706           IF(CARDP(IBEGST+1).NE.ALPH(15)) GOTO 2000               MAI00702
LN 0707           IF(LNGCRP.GT.IBEGST+5) GOTO 820                         MAI00703
LN 0708     810   NERROR=13                                               MAI00704
LN 0709           GOTO 9999                                               MAI00705
LN 0710     C                                                             MAI00706
LN 0711     C*****LOOK FOR EQUAL SIGN                                     MAI00707
LN 0712     820   DO 825 LOCE=IBEGST,LNGCRP                               MAI00708
LN 0713           IF(CARDP(LOCE).EQ.EQUALS) GOTO 835                      MAI00709
LN 0714     825   CONTINUE                                                MAI00710
LN 0715     C*****NONE FOUND                                              MAI00711
LN 0716           NERROR=14                                               MAI00712
LN 0717           GOTO 9999                                               MAI00713
LN 0718     C                                                             MAI00714
LN 0719     C*****FIND VARIABLE NUMBER                                    MAI00715
LN 0720     835   IF(LOCE.EQ.IBEGST+5) GOTO 860                           MAI00716
LN 0721           IF(LOCE.EQ.IBEGST+4) GOTO 850                           MAI00717
LN 0722     840   NERROR=15                                               MAI00718
LN 0723           GOTO 9999                                               MAI00719
LN 0724     C                                                             MAI00720
LN 0725     C*****ONE-CHARACTER VARIABLE                                  MAI00721
LN 0726     850   CALL ZALPH(CARDP(LOCE-1),IV)                            MAI00722
LN 0727           IF(IV.LE.26) GOTO 875                                   MAI00723
LN 0728           GOTO 840                                                MAI00724
LN 0729     C                                                             MAI00725
LN 0730     C*****TWO-CHARACTER VARIABLE                                  MAI00726
LN 0731     860   CALL ZALPH(CARDP(LOCE-2),K)                             MAI00727
LN 0732           IF(K.GT.26) GOTO 840                                    MAI00728
```

```
LN 0733          CALL ZDIGIT(CARDP(LOCE-1),L)                                    MAI00729
LN 0734          IF(L.GT.10) GOTO 840                                            MAI00730
LN 0735          IV=K+(26*(L-1))+53                                              MAI00731
LN 0736     C                                                                    MAI00732
LN 0737      875 IBEG1=LOCE+1                                                    MAI00733
LN 0738          LM1=LNGCRP-1                                                    MAI00734
LN 0739     C*****LOOK FOR TO                                                    MAI00735
LN 0740          IF(IBEG1.GT.LM1) GOTO 881                                       MAI00736
LN 0741          DO 880 LOCT=IBEG1,LM1                                           MAI00737
LN 0742          IF((CARDP(LOCT).EQ.ALPH(20)).AND.(CARDP(LOCT+1).EQ.ALPH(15)))   MAI00738
LN 0743         1 GOTO 885                                                       MAI00739
LN 0744      880 CONTINUE                                                        MAI00740
LN 0745      881 NERROR=16                                                       MAI00741
LN 0746          GOTO 9999                                                       MAI00742
LN 0747     C                                                                    MAI00743
LN 0748      885 IEND1=LOCT-1                                                    MAI00744
LN 0749          IBEG2=LOCT+2                                                    MAI00745
LN 0750     C*****LOOK FOR STEP                                                  MAI00746
LN 0751          IF(IBEG2.GT.LM1) GOTO 887                                       MAI00747
LN 0752          DO 886 LOCS=IBEG2,LM1                                           MAI00748
LN 0753          IF((CARDP(LOCS).EQ.ALPH(19)).AND.(CARDP(LOCS+1).EQ.ALPH(20)))   MAI00749
LN 0754         1 GOTO 888                                                       MAI00750
LN 0755      886 CONTINUE                                                        MAI00751
LN 0756      887 IEND2=LNGCRP                                                    MAI00752
LN 0757          IBEG3=0                                                         MAI00753
LN 0758          GOTO 890                                                        MAI00754
LN 0759      888 IEND2=LOCS-1                                                    MAI00755
LN 0760          IBEG3=LOCS+4                                                    MAI00756
LN 0761     C                                                                    MAI00757
LN 0762     C*****STORE EXPRESSION 1                                             MAI00758
LN 0763      890 IF(IBEG1.GT.IEND1) GOTO 810                                     MAI00759
LN 0764          CALL ZTRANX(IBEG1,IEND1)                                        MAI00760
LN 0765          IPROG(INREG)=-8                                                 MAI00761
LN 0766          ILOC=INEXT                                                      MAI00762
LN 0767          INEXT=INEXT+3                                                   MAI00763
LN 0768          IPROG(INREG-1)=ILOC                                             MAI00764
LN 0769          INREG=INREG-2                                                   MAI00765
LN 0770     C*****STORE EXPRESSION 2                                             MAI00766
LN 0771          IF(IBEG2 .GT.IEND2) GOTO 810                                    MAI00767
LN 0772          CALL ZTRANX(IBEG2,IEND2)                                        MAI00768
LN 0773          IPROG(INREG)=-8                                                 MAI00769
LN 0774          IPROG(INREG-1)=ILOC+1                                           MAI00770
LN 0775          INREG=INREG-2                                                   MAI00771
LN 0776     C*****STORE EXPRESSION 3                                             MAI00772
LN 0777          IF(IBEG3.EQ.0) GOTO 895                                         MAI00773
LN 0778          IF(IBEG3.GT.LNGCRP) GOTO 810                                    MAI00774
LN 0779          LNGXX=LNGCRP                                                    MAI00775
LN 0780          CALL ZTRANX(IBEG3,LNGXX)                                        MAI00776
LN 0781          IPROG(INREG)=-8                                                 MAI00777
LN 0782          IPROG(INREG-1)=ILOC+2                                           MAI00778
LN 0783          INREG=INREG-2                                                   MAI00779
LN 0784          GOTO 896                                                        MAI00780
LN 0785      895 DATA(ILOC+2)=1.                                                 MAI00781
LN 0786     C                                                                    MAI00782
LN 0787     C*****SET UP INITIAL VALUE STORE                                     MAI00783
LN 0788      896 IPROG(INREG)=-1                                                 MAI00784
LN 0789          IPROG(INREG-1)=3                                                MAI00785
LN 0790          IPROG(INREG-2)=ILOC                                             MAI00786
LN 0791          IPROG(INREG-3)=ILOC+2                                           MAI00787
LN 0792          IPROG(INREG-4)=-2                                               MAI00788
LN 0793          INREG=INREG-5                                                   MAI00789
```

```
LN 0794          IF(IV.GT.26) IPROG(INREG)=-8                                  MAI00790
LN 0795          IF(IV.LE.26) IPROG(INREG)=-9                                  MAI00791
LN 0796          IPROG(INREG-1)=IV                                             MAI00792
LN 0797          INREG=INREG-2                                                 MAI00793
LN 0798     C*****RECORD VARIABLE AND LOCATION FOR                             MAI00794
LN 0799          NIFOR=NIFOR+1                                                 MAI00795
LN 0800          IF(NIFOR.GT.NIFMAX) GOTO 898                                  MAI00796
LN 0801          IFOR(NIFOR,1)=IV                                              MAI00797
LN 0802          IFOR(NIFOR,2)=INREG                                           MAI00798
LN 0803     C                                                                  MAI00799
LN 0804     C*****WRITE FOR STATEMENT                                          MAI00800
LN 0805          IPROG(INREG)=-15                                              MAI00801
LN 0806          IPROG(INREG-1)=IV                                             MAI00802
LN 0807          IPROG(INREG-2)=ILOC                                           MAI00803
LN 0808          INREG=INREG-4                                                 MAI00804
LN 0809          GOTO 50                                                       MAI00805
LN 0810      898 NERROR=55                                                     MAI00806
LN 0811          GOTO 9999                                                     MAI00807
LN 0812     C                                                                  MAI00808
LN 0813      900 IF((CARDP(IBEGST).NE.ALPH(14)).OR.(CARDP(IBEGST+1).NE.ALPH(5)))MAI00809
LN 0814         1 GOTO 1000                                                    MAI00810
LN 0815     C*****COMMAND IS NEXT                                              MAI00811
LN 0816     C*****FIND VARIABLE IN STACK                                       MAI00812
LN 0817          IF(NIFOR.GT.0) GOTO 920                                       MAI00813
LN 0818     C*****NONE IN STACK                                                MAI00814
LN 0819          NERROR=17                                                     MAI00815
LN 0820          GOTO 9999                                                     MAI00816
LN 0821     C*****EXTRACT INFORMATION                                          MAI00817
LN 0822      920 IV=IFOR(NIFOR,1)                                              MAI00818
LN 0823          LOC=IFOR(NIFOR,2)                                             MAI00819
LN 0824          NIFOR=NIFOR-1                                                 MAI00820
LN 0825     C*****FIND VARIABLE NAME                                           MAI00821
LN 0826          IF(IV.GT.26) GOTO 930                                         MAI00822
LN 0827          CH1=ALPH(IV)                                                  MAI00823
LN 0828          CH2=BLANK                                                     MAI00824
LN 0829          GOTO 940                                                      MAI00825
LN 0830      930 L=((IV-54)/26)+1                                              MAI00826
LN 0831          K=(IV-53)-(26*(L-1))                                          MAI00827
LN 0832          CH1=ALPH(K)                                                   MAI00828
LN 0833          CH2=DIGIT(L)                                                  MAI00829
LN 0834     C*****CHECK VARIABLE IN NEXT STATEMENT                             MAI00830
LN 0835      940 IF(LNGCRP.EQ.IBEGST+4) GOTO 970                               MAI00831
LN 0836          IF(LNGCRP.EQ.IBEGST+5) GOTO 980                               MAI00832
LN 0837      950 NERROR=1                                                      MAI00833
LN 0838          NN=1                                                          MAI00834
LN 0839          CALL COMERR(NERROR,I1,I2,CH1,CH2,NN)                          MAI00835
LN 0840          GOTO 990                                                      MAI00836
LN 0841     C                                                                  MAI00837
LN 0842      970 IF(CARDP(IBEGST+4).EQ.CH1) GOTO 990                           MAI00838
LN 0843          GOTO 950                                                      MAI00839
LN 0844     C                                                                  MAI00840
LN 0845      980 IF((CARDP(IBEGST+4).EQ.CH1).AND.(CARDP(IBEGST+5).EQ.CH2))GOTO990MAI00841
LN 0846          GOTO 950                                                      MAI00842
LN 0847     C                                                                  MAI00843
LN 0848     C*****INSERT NEXT LOCATION IN THE ASSOCIATED FOR COMMAND           MAI00844
LN 0849      990 IPROG(LOC-3)=INREG-2                                          MAI00845
LN 0850     C*****SET UP TRANSFER                                              MAI00846
LN 0851          IPROG(INREG)=-4                                               MAI00847
LN 0852          IPROG(INREG-1)=LOC                                            MAI00848
LN 0853          IPROG(INREG-2)=-12                                            MAI00849
LN 0854          INREG=INREG-3                                                 MAI00850
```

```
LN 0855           GOTO 50                                                    MAI00851
LN 0856      C                                                               MAI00852
LN 0857       1000 IF(CARDP(IBEGST).NE.ALPH(19)) GOTO 1100                   MAI00853
LN 0858      C*****COMMAND IS STOP                                           MAI00854
LN 0859           IF(CARDP(IBEGST+1).NE.ALPH(20)) GOTO 2000                  MAI00855
LN 0860           IF(CARDP(IBEGST+2).NE.ALPH(15)) GOTO 2000                  MAI00856
LN 0861           IPROG(INREG)=-11                                           MAI00857
LN 0862           INREG=INREG-1                                              MAI00858
LN 0863           GOTO 50                                                    MAI00859
LN 0864      C                                                               MAI00860
LN 0865       1100 IF((CARDP(IBEGST).NE.ALPH(7)).OR.(CARDP(IBEGST+2).NE.ALPH(19)))  MAI00861
LN 0866          1 GOTO 1200                                                 MAI00862
LN 0867      C*****COMMAND IS GOSUB                                          MAI00863
LN 0868           IF(CARDP(IBEGST+4).NE.ALPH(2)) GOTO 2000                   MAI00864
LN 0869           IF(CARDP(IBEGST+1).NE.ALPH(15)) GOTO 2000                  MAI00865
LN 0870           IST=IBEGST+5                                               MAI00866
LN 0871           DO 1110 I=IST,LNGCRP                                       MAI00867
LN 0872           IF(CARDP(I).EQ.ALPH(15)) GOTO 1120                         MAI00868
LN 0873       1110 CONTINUE                                                  MAI00869
LN 0874           IPROG(INREG)=-2                                            MAI00870
LN 0875           INREG=INREG-1                                              MAI00871
LN 0876           GOTO 610                                                   MAI00872
LN 0877      C*****COMPUTED GOSUB-STATEMENT NUMBERS-ON(OF)-EXPRESSION-       MAI00873
LN 0878       1120 IPROG(INREG)=-44                                          MAI00874
LN 0879           INREG=INREG-1                                              MAI00875
LN 0880           GOTO 1130                                                  MAI00876
LN 0881      C                                                               MAI00877
LN 0882       1200  IF((CARDP(IBEGST).NE.ALPH(18)).OR.(CARDP(IBEGST+2).NE.ALPH(20)))  MAI00878
LN 0883          1 GOTO 1300                                                 MAI00879
LN 0884      C*****COMMAND IS RETURN                                         MAI00880
LN 0885           IF(CARDP(IBEGST+5).NE.ALPH(14)) GOTO 2000                  MAI00881
LN 0886           IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000                   MAI00882
LN 0887           IPROG(INREG)=-5                                            MAI00883
LN 0888           INREG=INREG-1                                              MAI00884
LN 0889           GOTO 50                                                    MAI00885
LN 0890      C                                                               MAI00886
LN 0891       1300 IF((CARDP(IBEGST).NE.ALPH(4)).OR.(CARDP(IBEGST+1).NE.ALPH(9)))  MAI00887
LN 0892          1 GOTO 1400                                                 MAI00888
LN 0893      C*****COMMAND IS DIM                                            MAI00889
LN 0894           IF(CARDP(IBEGST+2).NE.ALPH(13)) GOTO 2000                  MAI00890
LN 0895      C*****INSERT NO-OP COMMAND                                      MAI00891
LN 0896           IPROG(INREG)=-12                                           MAI00892
LN 0897           INREG=INREG-1                                              MAI00893
LN 0898      C                                                               MAI00894
LN 0899      C*****START                                                     MAI00895
LN 0900           IST=IBEGST+3                                               MAI00896
LN 0901       1310 IF(IST.GE.LNGCRP) GOTO 1390                               MAI00897
LN 0902      C                                                               MAI00898
LN 0903      C*****CHECK LEFT PARENTHESIS                                    MAI00899
LN 0904           IF(CARDP(IST+1).NE.DOLSGN) GOTO 1315                       MAI00900
LN 0905           CARDP(IST+1)=CARDP(IST)                                    MAI00901
LN 0906           IST=IST+1                                                  MAI00902
LN 0907           GOTO 1310                                                  MAI00903
LN 0908       1315 IF(CARDP(IST+1).NE.PARLFT) GOTO 1390                      MAI00904
LN 0909      C*****LOOK FOR RIGHT PARENTHESIS                                MAI00905
LN 0910           DO 1320 IRT=IST,LNGCRP                                     MAI00906
LN 0911           IF(CARDP(IRT).EQ.PARRT) GOTO 1330                          MAI00907
LN 0912       1320 CONTINUE                                                  MAI00908
LN 0913      C*****NONE FOUND                                                MAI00909
LN 0914           GOTO 1390                                                  MAI00910
LN 0915      C                                                               MAI00911
```

```
LN 0916    C*****FIND VARIABLE                                                    MAI00912
LN 0917     1330 CALL ZALPH(CARDP(IST),JV)                                        MAI00913
LN 0918    C*****ILLEGAL VARIABLE                                                 MAI00914
LN 0919          IF(JV.GT.26) GOTO 1390                                          MAI00915
LN 0920    C                                                                      MAI00916
LN 0921    C*****CHECK FOR COMMA                                                  MAI00917
LN 0922          DO 1360 IC=IST,IRT                                              MAI00918
LN 0923          IF(CARDP(IC).EQ.COMMA) GOTO 1370                                MAI00919
LN 0924     1360 CONTINUE                                                        MAI00920
LN 0925    C                                                                      MAI00921
LN 0926    C*****NONE FOUND                                                       MAI00922
LN 0927          CALL ZCONVN(IST+2,IRT-1,SIZE)                                   MAI00923
LN 0928          IF(SIZE.LT.0.) GOTO 1390                                        MAI00924
LN 0929          IF(MERKER(JV,1).EQ.0) GOTO 1365                                 MAI00925
LN 0930          NERROR=2                                                        MAI00926
LN 0931          NN=1                                                            MAI00927
LN 0932          CALL COMERR(NERROR,I1,I2,ALPH(JV),X2,NN)                        MAI00928
LN 0933     1365 MERKER(JV,1)=SIZE                                               MAI00929
LN 0934          MERKER(JV,2)=0                                                  MAI00930
LN 0935          ILNG=SIZE+2.1                                                   MAI00931
LN 0936          GOTO 1380                                                       MAI00932
LN 0937    C                                                                      MAI00933
LN 0938    C*****COMMA FOUND                                                      MAI00934
LN 0939     1370 CALL ZCONVN(IST+2,IC-1,ROWS)                                    MAI00935
LN 0940          IF(ROWS.LT.0.) GOTO 1390                                        MAI00936
LN 0941          IF(MERKER(JV,1).EQ.0) GOTO 1375                                 MAI00937
LN 0942          NERROR=2                                                        MAI00938
LN 0943          NN=1                                                            MAI00939
LN 0944          CALL COMERR(NERROR,I1,I2,ALPH(JV),X2,NN)                        MAI00940
LN 0945     1375 CALL ZCONVN(IC+1,IRT-1,COLS)                                    MAI00941
LN 0946          IF(COLS.LT.0.) GOTO 1390                                        MAI00942
LN 0947          MERKER(JV,2)=COLS                                               MAI00943
LN 0948          MERKER(JV,1)=ROWS                                               MAI00944
LN 0949          DATA(JV+27)=COLS+1.                                             MAI00945
LN 0950          ILNG=(ROWS+1.)*(COLS+1.)+1.1                                    MAI00946
LN 0951    C                                                                      MAI00947
LN 0952    C*****ALTER ADDRESS                                                    MAI00948
LN 0953     1380 IPROG(JV)=ILNG                                                  MAI00949
LN 0954    C                                                                      MAI00950
LN 0955    C*****CHECK FOR COMPLETION                                             MAI00951
LN 0956          IF(IRT.EQ.LNGCRP) GOTO 50                                       MAI00952
LN 0957    C                                                                      MAI00953
LN 0958    C*****TAKE NEXT CASE                                                   MAI00954
LN 0959          IST=IRT+2                                                       MAI00955
LN 0960          GOTO 1310                                                       MAI00956
LN 0961    C                                                                      MAI00957
LN 0962    C*****ERROR FOUND                                                      MAI00958
LN 0963     1390 NERROR=18                                                       MAI00959
LN 0964          GOTO 9999                                                       MAI00960
LN 0965    C                                                                      MAI00961
LN 0966     1400 IF((CARDP(IBEGST).NE.ALPH(18)).OR.(CARDP(IBEGST+2).NE.ALPH(13))) MAI00962
LN 0967         1 GOTO 1500                                                      MAI00963
LN 0968          IF(CARDP(IBEGST+1).NE.ALPH(5)) GOTO 2000                        MAI00964
LN 0969    C*****COMMAND IS REM                                                   MAI00965
LN 0970          IPROG(INREG)=-12                                                MAI00966
LN 0971          INREG=INREG-1                                                   MAI00967
LN 0972          GOTO 50                                                         MAI00968
LN 0973    C                                                                      MAI00969
LN 0974     1500 IF((CARDP(IBEGST).NE.ALPH(4)).OR.(CARDP(IBEGST+1).NE.ALPH(21))) MAI00970
LN 0975         1 GOTO 1600                                                      MAI00971
LN 0976    C*****COMMAND IS DUMP                                                  MAI00972
```

```
LN 0977           IF(CARDP(IBEGST+3).NE.ALPH(16)) GOTO 2000                    MAI00973
LN 0978           IPROG(INREG)=-19                                             MAI00974
LN 0979           INREG=INREG-1                                                MAI00975
LN 0980           GOTO 50                                                      MAI00976
LN 0981      C                                                                 MAI00977
LN 0982      C*****CHECK FOR AN END COMMAND                                    MAI00978
LN 0983       1600 IF((CARDP(IBEGST).EQ.ALPH(5)).AND.(CARDP(IBEGST+2).EQ.ALPH(4)))  MAI00979
LN 0984           1 GOTO 3000                                                  MAI00980
LN 0985      C                                                                 MAI00981
LN 0986           IF((CARDP(IBEGST).NE.ALPH(4)).OR.(CARDP(IBEGST+2).NE.ALPH(20)))  MAI00982
LN 0987           1 GOTO 2000                                                  MAI00983
LN 0988      C*****COMMAND IS DATA                                             MAI00984
LN 0989           IF(CARDP(IBEGST+1).NE.ALPH(1)) GOTO 2000                     MAI00985
LN 0990           IFR=IBEGST+4                                                 MAI00986
LN 0991      C                                                                 MAI00987
LN 0992      C*****GET NUMBERS                                                 MAI00988
LN 0993           CALL ZHOPPR(VALUE,NSTOP,IFR,0)                               MAI00989
LN 0994           IF(NSTOP.EQ.1) GOTO 1736                                     MAI00990
LN 0995           IF(NSTOP.EQ.2) GOTO 1735                                     MAI00991
LN 0996           IF(NUMBUF.GT.0) GOTO 1740                                    MAI00992
LN 0997           GOTO 1737                                                    MAI00993
LN 0998       1735 NERROR=19                                                   MAI00994
LN 0999           GOTO 9999                                                    MAI00995
LN 1000       1737 NERROR=3                                                    MAI00996
LN 1001           NN=1                                                         MAI00997
LN 1002           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                           MAI00998
LN 1003           GOTO 50                                                      MAI00999
LN 1004       1736 NERROR=30                                                   MAI01000
LN 1005           GOTO 9999                                                    MAI01001
LN 1006      C                                                                 MAI01002
LN 1007       1740 IST=NEXTDT+1                                               MAI01003
LN 1008           IEND=NEXTDT+NUMBUF                                           MAI01004
LN 1009           IF(IEND.LT.NCELLD) GOTO 1750                                 MAI01005
LN 1010      C*****TOO MUCH DATA                                               MAI01006
LN 1011           IX=IEND-NCELLD                                               MAI01007
LN 1012           NERROR=20                                                    MAI01008
LN 1013           I1=IX                                                        MAI01009
LN 1014           GOTO 9999                                                    MAI01010
LN 1015      C                                                                 MAI01011
LN 1016      C*****LOAD DATA                                                   MAI01012
LN 1017       1750 DO 1760 IX=IST,IEND                                        MAI01013
LN 1018           LOC=IX-NEXTDT                                                MAI01014
LN 1019       1760 DATAN(IX)=BUFFER(LOC)                                      MAI01015
LN 1020           NEXTDT=IEND                                                  MAI01016
LN 1021           GOTO 50                                                      MAI01017
LN 1022      C                                                                 MAI01018
LN 1023      C                                                                 MAI01019
LN 1024      C*****TRY THE COMMAND AS A LET STATEMENT                          MAI01020
LN 1025       2000 DO 2001 LOC=IBEGST,LNGCRP                                  MAI01021
LN 1026           IF(CARDP(LOC).EQ.QUOTE) GOTO 2003                            MAI01022
LN 1027           IF(CARDP(LOC).EQ.EQUALS) GOTO 2002                           MAI01023
LN 1028       2001 CONTINUE                                                   MAI01024
LN 1029           GOTO 2003                                                    MAI01025
LN 1030       2002 IBEGST=IBEGST-3                                            MAI01026
LN 1031           GOTO 230                                                     MAI01027
LN 1032      C*****ILLEGAL COMMAND                                             MAI01028
LN 1033       2003 NERROR=21                                                  MAI01029
LN 1034           GOTO 9999                                                    MAI01030
LN 1035      C                                                                 MAI01031
LN 1036      C*****END CARD REACHED                                            MAI01032
LN 1037       3000 IF(CARDP(IBEGST+1).NE.ALPH(14)) GOTO 2000                  MAI01033
```

```
LN 1038    C*****LIST DATA AFTER END STATEMENT                                    MAI01034
LN 1039          IF((CARDP(IBEGST+3).NE.ALPH(12)).OR.(CARDP(IBEGST+6).NE.ALPH(20)))MAI01035
LN 1040         1 GOTO 3010                                                      MAI01036
LN 1041          REWIND 999                                                      MAI01037
LN 1042          WRITE(IWC,3012)                                                 MAI01038
LN 1043     3012 FORMAT(//20X,32HLIST OF DATA AFTER END STATEMENT//)             MAI01039
LN 1044     3011 READ(IRC,51) CARD                                               MAI01040
LN 1045          IF(IFEOF(IRC).EQ.-1) GOTO 3020                                  MAI01041
LN 1046    C*-*-                                                                MAI01042
LN 1047    C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(IRC)-         MAI01043
LN 1048    C*-*-      ----CHECK IT----        ----CHECK IT----                  MAI01044
LN 1049    C*-*-                                                               MAI01045
LN 1050          WRITE(999,51) CARD                                            MAI01046
LN 1051          WRITE(IWC,52) CARD                                            MAI01047
LN 1052          IF(CARD(1).NE.ASTRSK) GOTO 3011                               MAI01048
LN 1053     3020 IRC=999                                                       MAI01049
LN 1054          REWIND 999                                                    MAI01050
LN 1055     3010 IPROG(INREG)=-11                                              MAI01051
LN 1056    C                                                                   MAI01052
LN 1057    C*****CHECK FOR DUPLICATE STATEMENT NUMBERS                         MAI01053
LN 1058          IF(NSTLST.LE.1) GOTO 5000                                     MAI01054
LN 1059          NM1=NSTLST-1                                                  MAI01055
LN 1060          DO 3500 I=1,NM1                                               MAI01056
LN 1061          IP=I+1                                                        MAI01057
LN 1062          DO 3400 J=IP,NSTLST                                           MAI01058
LN 1063          IF(LISTST(I).EQ.LISTST(J)) GOTO 3450                          MAI01059
LN 1064     3400 CONTINUE                                                      MAI01060
LN 1065          GOTO 3500                                                     MAI01061
LN 1066     3450 NERROR=22                                                     MAI01062
LN 1067          NN=0                                                          MAI01063
LN 1068          CALL COMERR(NERROR,LISTST(I),I2,X1,X2,NN)                     MAI01064
LN 1069          NERRS=NERRS+1                                                 MAI01065
LN 1070     3500 CONTINUE                                                      MAI01066
LN 1071    C                                                                   MAI01067
LN 1072    C*****CHECK FOR IMBALANCE BETWEEN FOR AND NEXT COMMANDS             MAI01068
LN 1073     5000 IF(NIFOR.EQ.0) GOTO 6000                                      MAI01069
LN 1074          NERRS=NERRS+NIFOR                                             MAI01070
LN 1075          NN=0                                                          MAI01071
LN 1076          NERROR=23                                                     MAI01072
LN 1077          CALL COMERR(NERROR,NIFOR,I2,X1,X2,NN)                         MAI01073
LN 1078    C                                                                   MAI01074
LN 1079    C*****SEE IF PROGRAM IS ACCEPTABLE                                  MAI01075
LN 1080     6000 IF(NERRS.EQ.0) GOTO 7000                                      MAI01076
LN 1081    C                                                                   MAI01077
LN 1082    C*****PROGRAM IS NOT ACCEPTABLE                                     MAI01078
LN 1083          WRITE(IWC,6001) NERRS                                         MAI01079
LN 1084     6001 FORMAT(1H0,26H****** SORRY BUT THERE ARE,I5,25H ERRORS I CANNOT OVMAI01080
LN 1085         1ERCOME)                                                       MAI01081
LN 1086          WRITE(IWC,6002)                                               MAI01082
LN 1087     6002 FORMAT(1H0,7X,21HBETTER LUCK NEXT TIME///)                    MAI01083
LN 1088    C                                                                   MAI01084
LN 1089          MCOM=1                                                        MAI01085
LN 1090          GOTO 6100                                                     MAI01086
LN 1091    C                                                                   MAI01087
LN 1092    C                                                                   MAI01088
LN 1093    C*****PROGRAM IS ACCEPTABLE                                         MAI01089
LN 1094     7000 WRITE(IWC,7250)                                               MAI01090
LN 1095     7250 FORMAT(1H0,191H*),38H EVERYTHING SEEMS OK -- LET*S GO AHEAD)      ***
LN 1096    C                                                                   MAI01093
LN 1097    C*****SET UP ADDRESSES FOR SUBSCRIPTED VARIABLES                    MAI01094
LN 1098          DO 8100 I=2,26                                                MAI01095
```

```
LN 1099      8100 IPROG(I)=IPROG(I)+IPROG(I-1)                                  MAI01096
LN 1100           DO 8150 I=1,26                                                MAI01097
LN 1101           J=28-I                                                        MAI01098
LN 1102      8150 DATA(J)=INEXT+IPROG(J-1)                                      MAI01099
LN 1103           DATA(1)=INEXT                                                 MAI01100
LN 1104      C                                                                  MAI01101
LN 1105      C*****CHECK FOR OVERLAP                                            MAI01102
LN 1106           ILSTD=DATA(27)                                                MAI01103
LN 1107           IF(ILSTD.LT.INREG) GOTO 8190                                  MAI01104
LN 1108           IDIFF=ILSTD-INREG                                             MAI01105
LN 1109           WRITE(IWC,8250) IDIFF                                         MAI01106
LN 1110      8250 FORMAT(1H0,58H**** YOU ARE TOO GREEDY -- YOUR PROGRAM PLUS DATA REMAI01107
LN 1111          1QUIRES,I10,35H MORE LOCATIONS THAN I CAN GIVE YOU)            MAI01108
LN 1112           WRITE(IWC,6002)                                               MAI01109
LN 1113           MCOM=1                                                        MAI01110
LN 1114           GOTO 6100                                                     MAI01111
LN 1115      C                                                                  MAI01112
LN 1116      C*****ENTER UPPER BOUNDS                                           MAI01113
LN 1117      8190 PERS=FLOAT(ILSTD+NCELLP-INREG)*100./FLOAT(NCELLP)             MAI01114
LN 1118           PERS=FLOAT(ILSTD+NCELLP-INREG)*100./FLOAT(NCELLP)             MAI01115
LN 1119           PERD=FLOAT(NEXTDT)*100./FLOAT(NCELLD)                         MAI01116
LN 1120           PERST=FLOAT(NSTLST)*100./FLOAT(NSTEND)                        MAI01117
LN 1121           WRITE(IWC,8260) PERS,PERD,PERST                               MAI01118
LN 1122      8260 FORMAT(1H0,20X,33HPERCENT OF AVAILABLE STORAGE USED,12X,F9.3/21X,3***
LN 1123          18HPERCENT OF AVAILABLE DATA STORAGE USED,7X,F9.3/21X,45HPERCENT OF***
LN 1124          1 AVAILABLE NUMBERED STATEMENTS USED,F9.3)                     ***
LN 1125           DO 8200 I=2,27                                                MAI01122
LN 1126           ILOC=DATA(I-1)+0.1                                            MAI01123
LN 1127      8200 DATA(ILOC)=DATA(I)                                            MAI01124
LN 1128      C                                                                  MAI01125
LN 1129      C*****BEGIN EXECUTION                                              MAI01126
LN 1130      C     CALL ZEXEC                                                   MAI01127
LN 1131      C     MCOM=1                                                       MAI01128
LN 1132      C     GOTO 6100                                                    MAI01129
LN 1133      C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   MAI01130
LN 1134      C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*     MAI01131
LN 1135      C---- GEARBEITET WIRD.IN DEN VORAUSGEHENDEN DREI STATEMENTS IST DAS C INMAI01132
LN 1136      C---- DER 1.SPALTE ZU ENTFERNEN.                                    MAI01133
LN 1137           GOTO 8211                                                     MAI01134
LN 1138      8210 STOP                                                          MAI01135
LN 1139      C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   MAI01136
LN 1140      C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*     MAI01137
LN 1141      C---- GEARBEITET WIRD.                                             MAI01138
LN 1142      8211 CONTINUE                                                      MAI01139
LN 1143           END                                                           MAI01140


    USASI FORTRAN DIAGNOSTIC RESULTS FOR COMPIL

 LINE   S   ERRNUM                    MESSAGE

 0006   N    0001     PROGRAM HAS PROGRAM STATEMENT.

 0026   N    0026     NON-USASI OVERLAY OR SEGMENT STATEMENT USED.
```

```
LN 0001          SUBROUTINE COMERR(NERROR,I1,I2,X1,X2,NN)                        COM00001
LN 0002     C**** SUBROUTINE COMERR TO PRINT SYNTAX-ERROR FIND DURING COMPILATION  COM00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,     COM00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOP,NIRET,NSTLST,INEXT, COM00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DOUOTE,MAXFIL, COM00005
LN 0006         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, COM00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZET    COM00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    COM00008
LN 0009         1NIFMAX,INTZEI                                                   COM00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                      COM00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    COM00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                     COM00012
LN 0013          COMMON// ISTLST(340),LISTST(340)                                COM00013
LN 0014          COMMON// DATAN(330)                                             ***
LN 0015          COMMON// DATA(3700)                                             COM00015
LN 0016          DIMENSION IPROG(3700)                                           COM00016
LN 0017          DIMENSION IERR(55),IWAR(7)                                      COM00017
LN 0018          EQUIVALENCE (DATA(1),IPROG(1))                                  COM00018
LN 0019          DATA IERR(1)/8HA  331.2/                                        COM00019
LN 0020          DATA IERR(2)/8HA  326.2/                                        COM00020
LN 0021          DATA IERR(3)/8HA 332.15/                                        COM00021
LN 0022          DATA IERR(4)/8HA  332.3/                                        COM00022
LN 0023          DATA IERR(5)/8HA  332.3/                                        COM00023
LN 0024          DATA IERR(6)/8HA  332.3/                                        COM00024
LN 0025          DATA IERR(7)/8HA  322.2/                                        COM00025
LN 0026          DATA IERR(8)/8HA 332.41/                                        COM00026
LN 0027          DATA IERR(9)/8HA 332.42/                                        COM00027
LN 0028          DATA IERR(10)/8HA 332.42/                                       COM00028
LN 0029          DATA IERR(11)/8HA 332.42/                                       COM00029
LN 0030          DATA IERR(12)/8HA 332.42/                                       COM00030
LN 0031          DATA IERR(13)/8HA 332.43/                                       COM00031
LN 0032          DATA IERR(14)/8HA 332.43/                                       COM00032
LN 0033          DATA IERR(15)/8HA 332.43/                                       COM00033
LN 0034          DATA IERR(16)/8HA 332.43/                                       COM00034
LN 0035          DATA IERR(17)/8HA 332.43/                                       COM00035
LN 0036          DATA IERR(18)/8HA 331.21/                                       COM00036
LN 0037          DATA IERR(19)/8HA 332.11/                                       COM00037
LN 0038          DATA IERR(20)/8HA 332.11/                                       COM00038
LN 0039          DATA IERR(21)/8HA  326.1/                                       COM00039
LN 0040          DATA IERR(22)/8HA  326.2/                                       COM00040
LN 0041          DATA IERR(23)/8HA 332.43/                                       COM00041
LN 0042          DATA IERR(24)/8HA  326.2/                                       COM00042
LN 0043          DATA IERR(25)/8HA 332.23/                                       COM00043
LN 0044          DATA IERR(26)/8HA 332.23/                                       COM00044
LN 0045          DATA IERR(27)/8HA 332.23/                                       COM00045
LN 0046          DATA IERR(28)/8HA  324.1/                                       COM00046
LN 0047          DATA IERR(29)/8HA  324.1/                                       COM00047
LN 0048          DATA IERR(30)/8HA 322.12/                                       COM00048
LN 0049          DATA IERR(31)/8HB    331/                                       COM00049
LN 0050          DATA IERR(32)/8HB331.111/                                       COM00050
LN 0051          DATA IERR(33)/8HB 331.21/                                       COM00051
LN 0052          DATA IERR(34)/8HB331.211/                                       COM00052
LN 0053          DATA IERR(35)/8HB331.211/                                       COM00053
LN 0054          DATA IERR(36)/8HB331.211/                                       COM00054
LN 0055          DATA IERR(37)/8HB331.211/                                       COM00055
LN 0056          DATA IERR(38)/8HB331.211/                                       COM00056
LN 0057          DATA IERR(39)/8HB331.111/                                       COM00057
LN 0058          DATA IERR(40)/8HB 331.21/                                       COM00058
LN 0059          DATA IERR(41)/8HA323/324/                                       COM00059
LN 0060          DATA IERR(42)/8HB    332/                                       COM00060
LN 0061          DATA IERR(43)/8HB 331.21/                                       COM00061
```

```
LN 0062          DATA IERR(44)/8HB 331.21/                                      COM00062
LN 0063          DATA IERR(45)/8HB 331.21/                                      COM00063
LN 0064          DATA IERR(46)/8HB 332.13/                                      COM00064
LN 0065          DATA IERR(47)/8HB 332.11/                                      COM00065
LN 0066          DATA IERR(48)/8HB 332.11/                                      COM00066
LN 0067          DATA IERR(49)/8HB 332.11/                                      COM00067
LN 0068          DATA IERR(50)/8HB 332.11/                                      COM00068
LN 0069          DATA IERR(51)/8HB 331.21/                                      COM00069
LN 0070          DATA IERR(52)/8HA 332.13/                                      COM00070
LN 0071          DATA IERR(53)/8HB331.112/                                      COM00071
LN 0072          DATA IERR(54)/8HB331.211/                                      COM00072
LN 0073          DATA IERR(55)/8HA 332.43/                                      COM00073
LN 0074          DATA IWAR(1)/8HA 332.43/                                       COM00074
LN 0075          DATA IWAR(2)/8HA 331.21/                                       COM00075
LN 0076          DATA IWAR(3)/8HA 332.11/                                       COM00076
LN 0077          DATA IWAR(4)/8HA  323.1/                                       COM00077
LN 0078          DATA IWAR(5)/8HA  323.2/                                       COM00078
LN 0079          DATA IWAR(6)/8HB 332.13/                                       COM00079
LN 0080          DATA IWAR(7)/8HA  332.2/                                       COM00080
LN 0081          IF(NN.EQ.-1) GOTO 997                                          COM00081
LN 0082          IF(NN.EQ.-2) GOTO 998                                          COM00082
LN 0083          IF(NN.EQ.-3) GOTO 999                                          COM00083
LN 0084          IF(NN.EQ.1) GOTO 1999                                          COM00084
LN 0085          IF(NN.EQ.2) GOTO 2999                                          COM00085
LN 0086          GOTO(502,504,506,508,510,512,514,516,518,520,522,524,526,528,530, COM00086
LN 0087         1532,534,536,538,540,542,544,546,548,550,552,554,556,558,560,562   COM00087
LN 0088         2,564,566,568,570,572,574,576,578,580,582,584,586,588,590,592,594  COM00088
LN 0089         3,596,598,600,602,604,606,608,610),NERROR                          COM00089
LN 0090     502  WRITE(IWC,10)                                                    COM00090
LN 0091     10   FORMAT(20X,55HPROGRAM AND DATA EXCEED AVAILABLE STORAGE AT THIS POCOM00091
LN 0092         1INT)                                                           COM00092
LN 0093          GOTO 1000                                                      COM00093
LN 0094     504  WRITE(IWC,20) ISTMAX                                           COM00094
LN 0095     20   FORMAT(20X,48HSORRY - I ACCEPT ONLY LINE NUMBERS BETWEEN 1 AND,I6)COM00095
LN 0096          GOTO 1000                                                      COM00096
LN 0097     506  WRITE(IWC,30)                                                  COM00097
LN 0098     30   FORMAT(20X,25HILLEGAL RESTORE STATEMENT)                       COM00098
LN 0099          GOTO 1000                                                      COM00099
LN 0100     508  WRITE(IWC,40)                                                  COM00100
LN 0101     40   FORMAT(20X,54HI CANNOT FIND AN EQUAL SIGN IN THE LET STATEMENT ABOCOM00101
LN 0102         1VE)                                                           COM00102
LN 0103          GOTO 1000                                                      COM00103
LN 0104     510  WRITE(IWC,50)                                                  COM00104
LN 0105     50   FORMAT(20X,40HTHE EQUAL SIGN ABOVE IS IN A FUNNY PLACE)        COM00105
LN 0106          GOTO 1000                                                      COM00106
LN 0107     512  WRITE(IWC,60)                                                  COM00107
LN 0108     60   FORMAT(20X,53HITEM TO THE LEFT OF THE EQUAL SIGN MUST BE A VARIABLCOM00108
LN 0109         1E)                                                            COM00109
LN 0110          GOTO 1000                                                      COM00110
LN 0111     514  WRITE(IWC,70)                                                  COM00111
LN 0112     70   FORMAT(20X,40HTHE QUOTATION MARKS ABOVE ARE NOT PAIRED)        COM00112
LN 0113          GOTO 1000                                                      COM00113
LN 0114     516  WRITE(IWC,80) ISTMAX                                           COM00114
LN 0115     80   FORMAT(20X,57HSORRY - I CAN ONLY GO TO LINES WITH NUMBERS BETWEEN COM00115
LN 0116         11 AND,I5)                                                     COM00116
LN 0117          GOTO 1000                                                      COM00117
LN 0118     518  WRITE(IWC,90)                                                  COM00118
LN 0119     90   FORMAT(20X,34HTHE IF STATEMENT ABOVE IS HOPELESS)              COM00119
LN 0120          GOTO 1000                                                      COM00120
LN 0121     520  WRITE(IWC,100)                                                 COM00121
LN 0122     100  FORMAT(20X,52HI CANNOT ACCEPT AN IF STATEMENT WITHOUT A COMPARISONCOM00122
```

```
LN 0123          1)                                                          COM00123
LN 0124           GOTO 1000                                                  COM00124
LN 0125      522  WRITE(IWC,110)                                             COM00125
LN 0126      110  FORMAT(20X,26HI CANNOT FIND THEN OR GOTO)                  COM00126
LN 0127           GOTO 1000                                                  COM00127
LN 0128      524  WRITE(IWC,120)                                             COM00128
LN 0129      120  FORMAT(20X,46HTHEN OR GOTO MUST BE FOLLOWED BY A LINE NUMBECOM00129
LN 0130           GOTO 1000                                                  COM00130
LN 0131      526  WRITE(IWC,130)                                             COM00131
LN 0132      130  FORMAT(20X,35HTHE FOR STATEMENT ABOVE IS HOPELESS)         COM00132
LN 0133           GOTO 1000                                                  COM00133
LN 0134      528  WRITE(IWC,140)                                             COM00134
LN 0135      140  FORMAT(20X,54HI CANNOT FIND AN EQUAL SIGN IN THE FOR STATEMENT AROCOM00135
LN 0136          1VE)                                                        COM00136
LN 0137           GOTO 1000                                                  COM00137
LN 0138      530  WRITE(IWC,150)                                             COM00138
LN 0139      150  FORMAT(20X,53HTHE VARIABLE IN A FOR STATEMENT MUST BE UNSUBSCRIPTECOM00139
LN 0140          1D)                                                         COM00140
LN 0141           GOTO 1000                                                  COM00141
LN 0142      532  WRITE(IWC,160)                                             COM00142
LN 0143      160  FORMAT(20X,18HI CANNOT FIND A TO)                          COM00143
LN 0144           GOTO 1000                                                  COM00144
LN 0145      534  WRITE(IWC,170)                                             COM00145
LN 0146      170  FORMAT(20X,50HEITHER YOU FORGOT A FOR STATEMENT OR I REJECTED IT) COM00146
LN 0147           GOTO 1000                                                  COM00147
LN 0148      536  WRITE(IWC,200)                                             COM00148
LN 0149      200  FORMAT(20X,33HDIM STATEMENT INCORRECTLY WRITTEN)           COM00149
LN 0150           GOTO 1000                                                  COM00150
LN 0151      538  WRITE(IWC,210)                                             COM00151
LN 0152      210  FORMAT(20X,27HWRONG CHARACTER WITHIN DATA)                 COM00152
LN 0153           GOTO 1000                                                  COM00153
LN 0154      540  WRITE(IWC,230) I1                                          COM00154
LN 0155      230  FORMAT(20X,11HYOU GAVE ME,I5,47H MORE DATA IN DATA STATEMENTS THANCOM00155
LN 0156          1 I CAN HANDLE)                                             COM00156
LN 0157           GOTO 1000                                                  COM00157
LN 0158      542  WRITE(IWC,240)                                             COM00158
LN 0159      240  FORMAT(20X,49HYOU MAY UNDERSTAND THE COMMAND ABOVE BUT I DO NOT)  COM00159
LN 0160           GOTO 1000                                                  COM00160
LN 0161      544  WRITE(IWC,250) I1                                          COM00161
LN 0162      250  FORMAT(20X,16HSTATEMENT NUMBER,I5,23H APPEARS MORE THAN ONCE)  COM00162
LN 0163           GOTO 1000                                                  COM00163
LN 0164      546  WRITE(IWC,260) I1                                          COM00164
LN 0165      260  FORMAT(20X,9HTHERE ARE,I6,39H FOR STATEMENTS WITHOUT NEXT STATEMENCOM00165
LN 0166          1TS)                                                        COM00166
LN 0167           GOTO 1000                                                  COM00167
LN 0168      548  WRITE(IWC,270) NSTEND                                      COM00168
LN 0169      270  FORMAT(20X,54HCOMPILATION STOPPED AT THIS POINT -- I CAN ONLY HANDCOM00169
LN 0170          1LE,I6,20H NUMBERED STATEMENTS)                             COM00170
LN 0171           GOTO 1000                                                  COM00171
LN 0172      550  WRITE(IWC,280)                                             COM00172
LN 0173      280  FORMAT(20X,47HAN IMAGE STATEMENT MUST HAVE A STATEMENT NUMBER)  COM00173
LN 0174           GOTO 1000                                                  COM00174
LN 0175      552  WRITE(IWC,290) MAXIMA                                      COM00175
LN 0176      290  FORMAT(20X,17HI CAN ONLY HANDLE,I5,17H IMAGE STATEMENTS)   COM00176
LN 0177           GOTO 1000                                                  COM00177
LN 0178      554  WRITE(IWC,300)                                             COM00178
LN 0179      300  FORMAT(20X,66HTHERE IS NO REFERENCE TO AN IMAGE STATEMENT WITHIN TCOM00179
LN 0180          1HE PRINT USING)                                            COM00180
LN 0181           GOTO 1000                                                  COM00181
LN 0182      556  WRITE(IWC,310)                                             COM00182
LN 0183      310  FORMAT(20X,57HI DO NOT LIKE THE EXPRESSION SHOWN BELOW (REASONS FOCOM00183
```

```
LN 0184          1LLOW))                                                    COM00184
LN 0185           GOTO 1000                                                 COM00185
LN 0186      558  WRITE(IWC,320)                                            COM00186
LN 0187      320  FORMAT(20X,65HTHERE IS SOMETHING I DO NOT LIKE ABOUT THE EXPRESSIOCOM00187
LN 0188          1N SHOWN BELOW)                                            COM00188
LN 0189           GOTO 1000                                                 COM00189
LN 0190      560  WRITE(IWC,330) IEXPO,IEXPO                                COM00190
LN 0191      330  FORMAT(20X,38HI CAN ONLY HANDLE NUMBERS BETWEEN 10**,I3,11H AND 10COM00191
LN 0192          1**(-,I3,1H))                                              COM00192
LN 0193           GOTO 1000                                                 COM00193
LN 0194      562  WRITE(IWC,335)                                            COM00194
LN 0195      335  FORMAT(20X,39HI CANNOT ACCEPT THE MAT STATEMENT ABOVE)    COM00195
LN 0196           GOTO 1000                                                 COM00196
LN 0197      564  WRITE(IWC,340)                                            COM00197
LN 0198      340  FORMAT(20X,65HI DO NOT UNDERSTAND THE FOLLOWING PART IN THE MAT STCOM00198
LN 0199          1ATEMENT ABOVE)                                            COM00199
LN 0200           WRITE(IWC,705) (CARDT(J),J=I1,I2)                         COM00200
LN 0201      705  FORMAT(1H ,10X,80A1)                                      COM00201
LN 0202           GOTO 1000                                                 COM00202
LN 0203      566  WRITE(IWC,345)                                            COM00203
LN 0204      345  FORMAT(20X,58HI CANNOT FIND AN EQUAL SIGN IN THE MAT LET STATEMENTCOM00204
LN 0205          1 ABOVE)                                                   COM00205
LN 0206           GOTO 1000                                                 COM00206
LN 0207      568  WRITE(IWC,350)                                            COM00207
LN 0208      350  FORMAT(20X,25HILLEGAL MAT LET STATEMENT)                  COM00208
LN 0209           GOTO 1000                                                 COM00209
LN 0210      570  WRITE(IWC,355) X1                                         COM00210
LN 0211      355  FORMAT(20X,66HI DO NOT LIKE THE FOLLOWING CHARACTER IN THE MAT STACOM00211
LN 0212          1TEMENT ABOVE *,A1,1H*)                                    COM00212
LN 0213           GOTO 1000                                                 COM00213
LN 0214      572  WRITE(IWC,360) I1,I2                                      COM00214
LN 0215      360  FORMAT(20X,10HVARIABLE *,A1,42H* IS DIMENSIONED AS A VECTOR -- VARCOM00215
LN 0216          1IABLE *,A1,28H* IS DIMENSIONED AS A MATRIX)               COM00216
LN 0217           GOTO 562                                                  COM00217
LN 0218      574  WRITE(IWC,365) X1                                         COM00218
LN 0219      365  FORMAT(20X,10HVARIABLE *,A1,36H* IS NOT DIMENSIONED AS A N*N MATRICOM00219
LN 0220          1X)                                                        COM00220
LN 0221           GOTO 562                                                  COM00221
LN 0222      576  WRITE(IWC,370)                                            COM00222
LN 0223      370  FORMAT(20X,35HNO ACCEPTABLE MATRIX MULTIPLICATION)        COM00223
LN 0224           GOTO 562                                                  COM00224
LN 0225      578  WRITE(IWC,375) X1                                         COM00225
LN 0226      375  FORMAT(20X,10HVARIABLE *,A1,20H* IS NOT DIMENSIONED)      COM00226
LN 0227           GOTO 562                                                  COM00227
LN 0228      580  WRITE(IWC,380) X1                                         COM00228
LN 0229      380  FORMAT(20X,10HVARIABLE *,A1,57H* IS DIMENSIONED AS A VECTOR BUT YOCOM00229
LN 0230          1U TREAT IT AS A MATRIX)                                   COM00230
LN 0231           GOTO 562                                                  COM00231
LN 0232      582  WRITE(IWC,385)                                            COM00232
LN 0233      385  FORMAT(20X,75HERROR IN INPUT OR OUTPUT LIST -- UNCORRECT VARIABLE,COM00233
LN 0234          1SUBSCRIPT OR EXPRESSION)                                  COM00234
LN 0235      603  WRITE(IWC,336) (CARDP(J),J=I1,I2)                         COM00235
LN 0236      336  FORMAT(20X,26HTHE PART I DO NOT LIKE IS ,80A1)            COM00236
LN 0237           GOTO 1000                                                 COM00237
LN 0238      584  WRITE(IWC,390)                                            COM00238
LN 0239      390  FORMAT(20X,36HTHE FILE STATEMENT ABOVE IS HOPELESS)       COM00239
LN 0240           GOTO 1000                                                 COM00240
LN 0241      586  WRITE(IWC,395) X1                                         COM00241
LN 0242      395  FORMAT(20X,47HILLEGAL FIELD NAME IN THE MAT STATEMENT ABOVE *,A1,1COM00242
LN 0243          1H*)                                                       COM00243
LN 0244           GOTO 1000                                                 COM00244
```

```
LN 0245     588  WRITE(IWC,400)                                                  COM00245
LN 0246     400  FORMAT(20X,53HFILE NAME ERROR -- THE QUOTATION MARKS ARE NOT PAIRECOM00246
LN 0247         1D)                                                             COM00247
LN 0248          GOTO 1000                                                      COM00248
LN 0249     590  I1=MAXSAT-1000                                                 COM00249
LN 0250          WRITE(IWC,405) I1,I2                                           COM00250
LN 0251     405  FORMAT(20X,49HI AM VERY SORRY BUT I AM ONLY ALLOWED TO GIVE YOU,I5COM00251
LN 0252         1,26H SENTENCES AND YOU ASK FOR,I6)                             COM00252
LN 0253          GOTO 1000                                                      COM00253
LN 0254     592  WRITE(IWC,410)                                                 COM00254
LN 0255     410  FORMAT(20X,56HTHE FIRST FILE STATEMENT MUST BE A COMMON-FILE STATECOM00255
LN 0256         1MENT)                                                          COM00256
LN 0257          GOTO 1000                                                      COM00257
LN 0258     594  WRITE(IWC,415)                                                 COM00258
LN 0259     415  FORMAT(20X,19HFILE SENTENCE ERROR)                             COM00259
LN 0260          GOTO 1000                                                      COM00260
LN 0261     596  WRITE(IWC,420)                                                 COM00261
LN 0262     420  FORMAT(20X,32HTHE FILE NAME ABOVE IS REDEFINED)                COM00262
LN 0263          GOTO 1000                                                      COM00263
LN 0264     598  WRITE(IWC,425) MAXFIL                                          COM00264
LN 0265     425  FORMAT(20X,50HI AM VERY SORRY BUT I CAN ONLY ALLOW YOU TO DEFINE,ICOM00265
LN 0266         13,6H FILES)                                                    COM00266
LN 0267          GOTO 1000                                                      COM00267
LN 0268      600 WRITE(IWC,430)                                                 COM00268
LN 0269      430 FORMAT(20X,74HTHE QUANTITY OF WORDS IN A INDEX SEQUENTIAL OPEN MUSCOM00269
LN 0270         1T BE GREATER THAN ZERO)                                        COM00270
LN 0271          GOTO 1000                                                      COM00271
LN 0272     602  WRITE(IWC,435)                                                 COM00272
LN 0273     435  FORMAT(20X,45HERROR ON THE LEFT SIDE OF A MAT LET STATEMENT)   COM00273
LN 0274          GOTO 564                                                       COM00274
LN 0275     604  WRITE(IWC,440)                                                 COM00275
LN 0276     440  FORMAT(20X,37HDATA CAN ONLY BE READ INTO A VARIABLE)           COM00276
LN 0277          GOTO 603                                                       COM00277
LN 0278     606  WRITE(IWC,445)                                                 COM00278
LN 0279     445  FORMAT(20X,36HMISSING PARENTHESIS IN MAT STATEMENT)            COM00279
LN 0280          GOTO 603                                                       COM00280
LN 0281     608  WRITE(IWC,450)                                                 COM00281
LN 0282     450  FORMAT(20X,37HDIMENSION NOT ALLOWED WITH EXPRESSION)           COM00282
LN 0283          GOTO 564                                                       COM00283
LN 0284     610  WRITE(IWC,455) NIFMAX                                          COM00284
LN 0285     455  FORMAT(20X,17HI CAN ONLY HANDLE,I4,39H FOR STATEMENTS WITHOUT NEXTCOM00285
LN 0286         1 STATMENTS)                                                    COM00286
LN 0287          GOTO 1000                                                      COM00287
LN 0288      999 WRITE(IWC,1002)                                                COM00288
LN 0289     1000 WRITE(IWC,1001) NERROR                                         COM00289
LN 0290     1001 FORMAT(1H+,110X,16H***ERROR NUMBER=,I3)                        COM00290
LN 0291          WRITE(IWC,1003) IERR(NERROR)                                   COM00291
LN 0292     1003 FORMAT(1H ,10X,18HSEE BASIC TEXTBOOK,2X,A8)                    COM00292
LN 0293     9999 WRITE(IWC,1002)                                                COM00293
LN 0294     1002 FORMAT(1H )                                                    COM00294
LN 0295          RETURN                                                         COM00295
LN 0296     1999 GOTO(5000,5002,5004,5006,5008,5010,5012),NERROR                COM00296
LN 0297     5000 WRITE(IWC,180) X1,X2                                           COM00297
LN 0298      180  FORMAT(20X,35HI EXPECTED YOU TO MENTION VARIABLE ,2A1,30H -- I WILCOM00298
LN 0299         1L ASSUME THAT YOU DID)                                         COM00299
LN 0300          GOTO 2000                                                      COM00300
LN 0301     5002 WRITE(IWC,190) X1                                              COM00301
LN 0302      190  FORMAT(20X,9HVARIABLE ,A1,42H REDIMENSIONED -- BUT I TAKE IT AS A COM00302
LN 0303         1 JOKE)                                                         COM00303
LN 0304          GOTO 2000                                                      COM00304
LN 0305     5004 WRITE(IWC,220)                                                 COM00305
```

```
LN 0306     220  FORMAT(20X,40HTHERE ARE NO DATA IN THIS DATA STATEMENT)                COM00306
LN 0307          GOTO 2000                                                             COM00307
LN 0308     5006 WRITE(IWC,185)                                                        COM00308
LN 0309     185  FORMAT(20X,51HYOU TREAT A NUMERIC VARIABLE AS AN ALPHANUMERIC ONE)COM00309
LN 0310          GOTO 2000                                                             COM00310
LN 0311     5008 WRITE(IWC,195)                                                        COM00311
LN 0312     195  FORMAT(20X,51HYOU TREAT AN ALPHANUMERIC VARIABLE AS A NUMERIC ONE)COM00312
LN 0313          GOTO 2000                                                             COM00313
LN 0314     5010 WRITE(IWC,205)                                                        COM00314
LN 0315     205  FORMAT(20X,76HI CANNOT ACCEPT A NUMBER FOR WORDS IN A COMMON-FILE COM00315
LN 0316         1 STATEMENT - I IGNORE IT)                                            COM00316
LN 0317          GOTO 2000                                                             COM00317
LN 0318     5012 WRITE(IWC,215)                                                        COM00318
LN 0319     215  FORMAT(20X,26HMISSING COMMA OR SEMICOLON)                            COM00319
LN 0320          GOTO 2000                                                             COM00320
LN 0321     998  WRITE(IWC,1002)                                                       COM00321
LN 0322     2000 WRITE(IWC,2001) NERROR                                                COM00322
LN 0323     2001 FORMAT(1H+,110X,18H***WARNING NUMBER=,I3)                             COM00323
LN 0324          WRITE(IWC,1003) IWAR(NERROR)                                          COM00324
LN 0325          GOTO 9999                                                             COM00325
LN 0326     2999 GOTO(3002,3004,3006,3008,3010,3012,3014,3016,3018,3020,3022,3024),COM00326
LN 0327         1NERROR                                                             COM00327
LN 0328     3002 WRITE(IWC,3102)                                                       COM00328
LN 0329     3102 FORMAT(30X,28HA -- THIS IS AN ILLEGAL NAME)                           COM00329
LN 0330          GOTO 3200                                                             COM00330
LN 0331     3004 WRITE(IWC,3104)                                                       COM00331
LN 0332     3104 FORMAT(30X,51HB -- NO RIGHT PARENTHESIS FOR THIS LEFT PARENTHESIS)COM00332
LN 0333          GOTO 3200                                                             COM00333
LN 0334     3006 WRITE(IWC,3106)                                                       COM00334
LN 0335     3106 FORMAT(30X,51HC -- NO LEFT PARENTHESIS FOR THIS RIGHT PARENTHESIS)COM00335
LN 0336          GOTO 3200                                                             COM00336
LN 0337     3008 WRITE(IWC,3108)                                                       COM00337
LN 0338     3108 FORMAT(30X,35HD -- THIS COMMA IS IN A FUNNY PLACE)                    COM00338
LN 0339          GOTO 3200                                                             COM00339
LN 0340     3010 WRITE(IWC,3110)                                                       COM00340
LN 0341     3110 FORMAT(30X,38HE -- THIS PAIR OF OPERATORS IS ILLEGAL)                 COM00341
LN 0342          GOTO 3200                                                             COM00342
LN 0343     3012 WRITE(IWC,3112)                                                       COM00343
LN 0344     3112 FORMAT(30X,34HF -- THERE IS SOMETHING FUNNY HERE)                     COM00344
LN 0345          GOTO 3200                                                             COM00345
LN 0346     3014 WRITE(IWC,3114)                                                       COM00346
LN 0347     3114 FORMAT(30X,56HG -- A CONSTANT CANNOT BE FOLLOWED BY A LEFT PARENTHCOM00347
LN 0348         1ESIS)                                                              COM00348
LN 0349          GOTO 3200                                                             COM00349
LN 0350     3016 WRITE(IWC,3116)                                                       COM00350
LN 0351     3116 FORMAT(30X,30HH -- THIS OPERATOR IS DANGLING)                         COM00351
LN 0352          GOTO 3200                                                             COM00352
LN 0353     3018 WRITE(IWC,3118)                                                       COM00353
LN 0354     3118 FORMAT(30X,69HI -- A RIGHT PARENTHESIS CANNOT BE FOLLOWED BY A VARCOM00354
LN 0355         1IABLE OR CONSTANT)                                                 COM00355
LN 0356          GOTO 3200                                                             COM00356
LN 0357     3020 WRITE(IWC,3120)                                                       COM00357
LN 0358     3120 FORMAT(30X,43HJ -- THESE VARIABLES ARE NEXT TO EACH OTHER)            COM00358
LN 0359          GOTO 3200                                                             COM00359
LN 0360     3022 WRITE(IWC,3122)                                                       COM00360
LN 0361     3122 FORMAT(30X,43HK -- THESE CONSTANTS ARE NEXT TO EACH OTHER)            COM00361
LN 0362          GOTO 3200                                                             COM00362
LN 0363     3024 WRITE(IWC,3124)                                                       COM00363
LN 0364     3124 FORMAT(30X,37HL -- A VARIABLE IS NEXT TO A CONSTANT)                  COM00364
LN 0365          GOTO 3200                                                             COM00365
LN 0366     997  WRITE(IWC,1002)                                                       COM00366
```

```
LN 0367      3200 WRITE(IWC,3201) NERROR                                    COM00367
LN 0368      3201 FORMAT(1H+,110X,17H***REASON NUMBER=,I3)                  COM00368
LN 0369           GOTO 9999                                                 COM00369
LN 0370           END                                                       COM00370

     USASI FORTRAN DIAGNOSTIC RESULTS FOR COMERR


         NO ERRORS
```

ZINITL

```
LN 0001          SUBROUTINE ZINITL                                                  ZIN00001
LN 0002    C*****ROUTINE TO INITIALIZE SYSTEM                                       ZIN00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,      ZIN00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZIN00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZIN00005
LN 0006         3IRC,IWC,NSTEND,TEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZIN00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZIN00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,      ZIN00008
LN 0009         1NIFMAX,INTZEI                                                     ZIN00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                        ZIN00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CAROP(80),DIGIT(10),IFOR(20,2),      ZIN00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                       ZIN00012
LN 0013          COMMON// TSTLST(340),LISTST(340)                                  ZIN00013
LN 0014          COMMON// DATAN(330)                                               ***
LN 0015          COMMON// XDATA(3700)                                              ZIN00015
LN 0016          DIMENSION IPROG(3700)                                             ZIN00016
LN 0017          EQUIVALENCE (XDATA(1),IPROG(1))                                   ZIN00017
LN 0018          DIMENSION ATEMP(26),DIGTMP(10),CHARTM(20)                         ZIN00018
LN 0019          DATA ATEMP(1)/1HA/                                                ZIN00019
LN 0020          DATA ATEMP(2)/1HB/                                                ZIN00020
LN 0021          DATA ATEMP(3)/1HC/                                                ZIN00021
LN 0022          DATA ATEMP(4)/1HD/                                                ZIN00022
LN 0023          DATA ATEMP(5)/1HE/                                                ZIN00023
LN 0024          DATA ATEMP(6)/1HF/                                                ZIN00024
LN 0025          DATA ATEMP(7)/1HG/                                                ZIN00025
LN 0026          DATA ATEMP(8)/1HH/                                                ZIN00026
LN 0027          DATA ATEMP(9)/1HI/                                                ZIN00027
LN 0028          DATA ATEMP(10)/1HJ/                                               ZIN00028
LN 0029          DATA ATEMP(11)/1HK/                                               ZIN00029
LN 0030          DATA ATEMP(12)/1HL/                                               ZIN00030
LN 0031          DATA ATEMP(13)/1HM/                                               ZIN00031
LN 0032          DATA ATEMP(14)/1HN/                                               ZIN00032
LN 0033          DATA ATEMP(15)/1HO/                                               ZIN00033
LN 0034          DATA ATEMP(16)/1HP/                                               ZIN00034
LN 0035          DATA ATEMP(17)/1HQ/                                               ZIN00035
LN 0036          DATA ATEMP(18)/1HR/                                               ZIN00036
LN 0037          DATA ATEMP(19)/1HS/                                               ZIN00037
LN 0038          DATA ATEMP(20)/1HT/                                               ZIN00038
LN 0039          DATA ATEMP(21)/1HU/                                               ZIN00039
LN 0040          DATA ATEMP(22)/1HV/                                               ZIN00040
LN 0041          DATA ATEMP(23)/1HW/                                               ZIN00041
LN 0042          DATA ATEMP(24)/1HX/                                               ZIN00042
LN 0043          DATA ATEMP(25)/1HY/                                               ZIN00043
LN 0044          DATA ATEMP(26)/1HZ/                                               ZIN00044
LN 0045          DATA DIGTMP(1)/1H0/                                               ZIN00045
LN 0046          DATA DIGTMP(2)/1H1/                                               ZIN00046
LN 0047          DATA DIGTMP(3)/1H2/                                               ZIN00047
LN 0048          DATA DIGTMP(4)/1H3/                                               ZIN00048
LN 0049          DATA DIGTMP(5)/1H4/                                               ZIN00049
LN 0050          DATA DIGTMP(6)/1H5/                                               ZIN00050
LN 0051          DATA DIGTMP(7)/1H6/                                               ZIN00051
LN 0052          DATA DIGTMP(8)/1H7/                                               ZIN00052
LN 0053          DATA DIGTMP(9)/1H8/                                               ZIN00053
LN 0054          DATA DIGTMP(10)/1H9/                                              ZIN00054
LN 0055          DATA CHARTM(1)/1H*/                                               ZIN00055
LN 0056          DATA CHARTM(2)/1H /                                               ZIN00056
LN 0057          DATA CHARTM(3)/1H,/                                               ZIN00057
LN 0058          DATA CHARTM(4)/1H./                                               ZIN00058
LN 0059          DATA CHARTM(5)/1H=/                                               ZIN00059
LN 0060          DATA CHARTM(6)/1H)/                                               ZIN00060
LN 0061          DATA CHARTM(7)/1H(/                                               ZIN00061
```

```
LN 0062          DATA CHARTM(8)/1H+/                                                ZIN00062
LN 0063          DATA CHARTM(9)/1H#/                                                ZIN00063
LN 0064          DATA CHARTM(10)/1H$/                                               ZIN00064
LN 0065          DATA CHARTM(11)/1H-/                                               ZIN00065
LN 0066          DATA CHARTM(12)/1H//                                               ZIN00066
LN 0067          DATA CHARTM(13)/1H;/                                               ZIN00067
LN 0068          DATA CHARTM(14)/1H</                                               ZIN00068
LN 0069          DATA CHARTM(15)/1H>/                                               ZIN00069
LN 0070          DATA CHARTM(16)/1H&/                                               ZIN00070
LN 0071          DATA CHARTM(17)/1HX/                                               ZIN00071
LN 0072          DATA CHARTM(18)/1H*/                                               ZIN00072
LN 0073          DATA CHARTM(19)/1HV/                                               ZIN00073
LN 0074          DATA CHARTM(20)/1H%/                                               ZIN00074
LN 0075          DATA ICONT/8HBUILD---/                                             ***
LN 0076          DATA ILIST/8HLIST-ERR/                                             ZIN00076
LN 0077          DATA ILINE/8HNO-HEADL/                                             ZIN00077
LN 0078    C----AUF DIE FOLGENDE ZEILE LN=0079 WIRD IM TEXT BEZUG GENOMMEN vv       ***
LN 0079          NSTZEI=1                                                           ZIN00078
LN 0080    C*****INITIALIZE                                                         ZIN00079
LN 0081    C                                                                        ZIN00080
LN 0082    C*****DATA AND IPROG MUST BE DIMENSIONED TO EQUAL NCELLP                 ZIN00081
LN 0083          NCELLP=3700                                                        ZIN00082
LN 0084    C*****DATAN MUST BE DIMENSIONED TO EQUAL NCELLD                          ZIN00083
LN 0085          NCELLD=330                                                         ***
LN 0086    C*****CHANNEL NUMBER FOR READ IS IRC=60 (STANDARD INPUT SYSTEM)          ZIN00085
LN 0087    C******IMIRC RESETS IRC TO THE VALUE OF STANDRAD INPUT SYSTEM IF IRC IS  ZIN00086
LN 0088    C      CHANGED                                                           ZIN00087
LN 0089          IRC=60                                                             ZIN00088
LN 0090          IMIRC=IRC                                                          ZIN00089
LN 0091    C*****CHANNEL NUMBER FOR WRITE IS IWC=61  (STANDARD OUTPUT SYSTEM)       ZIN00090
LN 0092          IWC=61                                                             ZIN00091
LN 0093    C**** BUILT INTERNAL CODE FOR ALL CHARACTERS IF NSTZEI NOT EQUAL 1       ZIN00092
LN 0094    C**** AND IF THE FIRST CARD INCLUDS THE WORD *BUILD* BEGINNING WITH      ZIN00093
LN 0095    C**** COLUMN 3                                                           ZIN00094
LN 0096          REWIND 999                                                         ZIN00095
LN 0097          WRITE(999,2) (CARD(I),I=3,10)                                      ZIN00096
LN 0098          REWIND 999                                                         ZIN00097
LN 0099          READ(999,3) IBUILT                                                 ZIN00098
LN 0100     3    FORMAT(A8)                                                         ZIN00099
LN 0101          IF((IBUILT.NE.ICONT).OR.(NSTZEI.EQ.1)) GOTO 1                      ZIN00100
LN 0102    C**** LETTERS                                                            ZIN00101
LN 0103          READ(IRC,2) (ATEMP(I),I=1,26)                                      ZIN00102
LN 0104     2    FORMAT(80A1)                                                       ZIN00103
LN 0105    C**** NUMBERS                                                            ZIN00104
LN 0106          READ(IRC,2) (DIGTMP(I),I=1,10)                                     ZIN00105
LN 0107    C*****SPECIAL CHARACTERS                                                 ZIN00106
LN 0108          READ(IRC,2) (CHARTM(I),I=1,20)                                     ZIN00107
LN 0109          NSTZEI=1                                                           ZIN00108
LN 0110    C*****MAXIMAL INTERNAL EXPONENT FOR FLOATING POINT NUMBERS IS IEXPO      ZIN00109
LN 0111     1    IEXPO=99                                                          ZIN00110
LN 0112    C**** MAXIMAL INTEGER VALUE FOR 48-BIT-WORD                              ZIN00111
LN 0113          INTMAX=140737488355327                                             ZIN00112
LN 0114    C**** MAXIMAL EXPONENT FOR 48-BIT-WORD                                   ZIN00113
LN 0115          INTNUM=14                                                          ZIN00114
LN 0116    C*****NZIM COUNTS IMAGE STATEMENTS                                       ZIN00115
LN 0117          NZIM=1                                                             ZIN00116
LN 0118    C*****CHANNEL NUMBER FOR IMAGE MEMORY ON DISK                            ZIN00117
LN 0119          NIMAGE=100                                                         ZIN00118
LN 0120          REWIND NIMAGE                                                      ZIN00119
LN 0121    C**** MAXIMAL NUMBER OF IMAGE STATEMENTS                                 ZIN00120
LN 0122          MAXIMA=300                                                         ZIN00121
```

```
LN 0123     C*****CHARACTERS PER PRINT ZONE                                      ZIN00122
LN 0124           IZONE=15                                                       ZIN00123
LN 0125     C**** CHARACTERS PER LINE WITH PRINT USING                           ZIN00124
LN 0126           IIMAGE=135                                                     ZIN00125
LN 0127     C**** SMALLEST ABSOLUT VALUE WHICH CAN BE PRINTED WITH STANDARD PRINT ZIN00126
LN 0128           SMALL=1.E-38                                                   ZIN00127
LN 0129     C**** MAXIMAL STATEMENT NUMBER                                       ZIN00128
LN 0130           ISTMAX=9999                                                    ZIN00129
LN 0131     C**** MAXIMAL NESTED FOR/NEXT LOOPS                                  ZIN00130
LN 0132     C**** IFOR MUST BE DIMENSIONED EQUAL TO NIFMAX                       ZIN00131
LN 0133           NIFMAX=20                                                      ZIN00132
LN 0134     C**** MAXIMAL NESTED GOSUB/RETURN                                    ZIN00133
LN 0135     C**** IRET MUST BE DIMENSIONED EQUAL TO NIRMAX                       ZIN00134
LN 0136           NIRMAX=20                                                      ZIN00135
LN 0137           WRITE(IWC,5)                                                   ZIN00136
LN 0138      5    FORMAT(1H1)                                                    ZIN00137
LN 0139           INREG=NCELLP                                                   ZIN00138
LN 0140           IF(IBUILT.NE.ILIST) GOTO 9                                     ZIN00139
LN 0141           NSTZEI=2                                                       ZIN00140
LN 0142     C**** NM=55 NUMBER OF ERRORS                                         ZIN00141
LN 0143           NM=55                                                          ZIN00142
LN 0144           NN=-1                                                          ZIN00143
LN 0145      8    NN=NN+1                                                        ZIN00144
LN 0146           DO 6 I=1,NM                                                    ZIN00145
LN 0147           NERROR=I                                                       ZIN00146
LN 0148           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                             ZIN00147
LN 0149      6    CONTINUE                                                       ZIN00148
LN 0150           IF(NN.EQ.1) GOTO 7                                             ZIN00149
LN 0151           IF(NN.GE.2) GOTO 4                                             ZIN00150
LN 0152     C**** NM=7 NUMBER OF WARNINGS                                        ZIN00151
LN 0153     C**** WARNING NUMBER 8 AND 9 CONTAINS SUBROUTINE *STRING*            ZIN00152
LN 0154           NM=7                                                           ZIN00153
LN 0155           GOTO 8                                                         ZIN00154
LN 0156     C**** NM=12 NUMBER OF REASONS                                        ZIN00155
LN 0157      7    NM=12                                                          ZIN00156
LN 0158           GOTO 8                                                         ZIN00157
LN 0159      4    WRITE(IWC,5)                                                   ZIN00158
LN 0160      9    IF(IBUILT.EQ.ILINE) NSTZEI=3                                   ZIN00159
LN 0161           DO 60 I=1,10                                                   ZIN00160
LN 0162           IF(CARD(2).EQ.DIGTMP(I)) GOTO 61                               ZIN00161
LN 0163     60    CONTINUE                                                       ZIN00162
LN 0164           IPEND=5                                                        ZIN00163
LN 0165           GOTO 62                                                        ZIN00164
LN 0166     61    IPEND=I-1                                                      ZIN00165
LN 0167           IF(IPEND.EQ.0)IPEND=1                                          ZIN00166
LN 0168           IF(IPEND.GT.8) IPEND=8                                         ZIN00167
LN 0169     62    IWRIT=IZONE*IPEND+1                                            ZIN00168
LN 0170     C*****ISTLST AND LISTST MUST BE DIMENSIONED EQUAL TO NSTEND          ZIN00169
LN 0171           NSTEND=340                                                     ZIN00170
LN 0172     C**** INITIALIZE FILE-COMMANDS                                       ZIN00171
LN 0173           NERRS=-1                                                       ZIN00172
LN 0174     C**** MAXFIL = MAXIMAL NUMBER OF FILE-NAMES                          ZIN00173
LN 0175           MAXFIL=25                                                      ZIN00174
LN 0176     C**** MAXSAT LESS 1000 = MAXIMAL NUMBER OF ALLOCATED SENTENCES       ZIN00175
LN 0177           MAXSAT=1024                                                    ZIN00176
LN 0178     C----AUF DIE FOLGENDE ZEILE LN=0179 WIRD IM TEXT BEZUG GENOMMENvv    ***
LN 0179           CALL ZFILE(NERRS)                                              ZIN00177
LN 0180           INEXT=324                                                      ZIN00178
LN 0181           NERRS=0                                                        ZIN00179
LN 0182           NEXTDT=0                                                       ZIN00180
LN 0183           NIFOR=0                                                        ZIN00181
```

```
LN 0184           NSTLST=0                                                  ZIN00182
LN 0185           DO 20 I=1,26                                              ZIN00183
LN 0186      20   IPROG(I)=2                                                ZIN00184
LN 0187           DO 30 I=28,53                                             ZIN00185
LN 0188      30   XDATA(I)=11.                                              ZIN00186
LN 0189           DO 35 I=54,313                                            ZIN00187
LN 0190      35   XDATA(I)=0.                                               ZIN00188
LN 0191           DO 36 I=314,323                                           ZIN00189
LN 0192      36   XDATA(I)=I-314                                            ZIN00190
LN 0193           DO 38 I=324,NCELLP                                        ZIN00191
LN 0194      38   XDATA(I)=0.                                               ZIN00192
LN 0195      C                                                              ZIN00193
LN 0196      C*****SET UP VOCABULARY                                        ZIN00194
LN 0197      C*****LOAD ALPHABETIC CHARACTERS                               ZIN00195
LN 0198           DO 40 I=1,26                                              ZIN00196
LN 0199      40   ALPH(I)=ATEMP(I)                                          ZIN00197
LN 0200      C                                                              ZIN00198
LN 0201      C*****LOAD DIGITS                                              ZIN00199
LN 0202           DO 41 I=1,10                                              ZIN00200
LN 0203      41   DIGIT(I)=DIGTMP(I)                                        ZIN00201
LN 0204      C                                                              ZIN00202
LN 0205      C*****LOAD SPECIAL CHARACTERS                                  ZIN00203
LN 0206           ASTRSK=CHARTM(1)                                          ZIN00204
LN 0207           BLANK=CHARTM(2)                                           ZIN00205
LN 0208           COMMA=CHARTM(3)                                           ZIN00206
LN 0209           DECMAL=CHARTM(4)                                          ZIN00207
LN 0210           EQUALS=CHARTM(5)                                          ZIN00208
LN 0211           PARRT=CHARTM(6)                                           ZIN00209
LN 0212           PARLFT=CHARTM(7)                                          ZIN00210
LN 0213           PLUS=CHARTM(8)                                            ZIN00211
LN 0214           QUOTE=CHARTM(9)                                           ZIN00212
LN 0215           DOLSGN=CHARTM(10)                                         ZIN00213
LN 0216           CMINUS=CHARTM(11)                                         ZIN00214
LN 0217           SLASH=CHARTM(12)                                          ZIN00215
LN 0218           PUCO=CHARTM(13)                                           ZIN00216
LN 0219           VLESS=CHARTM(14)                                          ZIN00217
LN 0220           VGREAT=CHARTM(15)                                         ZIN00218
LN 0221           DQUOTF=CHARTM(16)                                         ZIN00219
LN 0222           DOPU=CHARTM(17)                                           ZIN00220
LN 0223           EXSIGN=CHARTM(18)                                         ZIN00221
LN 0224           XNULL=CHARTM(19)                                          ZIN00222
LN 0225           DDOPU=CHARTM(20)                                          ZIN00223
LN 0226      C                                                              ZIN00224
LN 0227      C**** CLEAR MERKER(I,J)                                        ZIN00225
LN 0228           DO 44 I=1,2                                               ZIN00226
LN 0229           DO 44 J=1,26                                              ZIN00227
LN 0230           MERKER(J,I)=0                                             ZIN00228
LN 0231      44   CONTINUE                                                  ZIN00229
LN 0232      C                                                              ZIN00230
LN 0233      C*****LOAD DIGITS INTO POSITIONS 27 THROUGH 36 OF ALPH         ZIN00231
LN 0234           LOC=26                                                    ZIN00232
LN 0235           DO 42 I=1,10                                              ZIN00233
LN 0236           LOC=LOC+1                                                 ZIN00234
LN 0237      42   ALPH(LOC)=DIGIT(I)                                        ZIN00235
LN 0238      C                                                              ZIN00236
LN 0239      C*****LOAD OTHER CHARACTERS INTO POSITIONS 37 THROUGH 48 OF ALPH ZIN00237
LN 0240           DO 43 I=1,12                                              ZIN00238
LN 0241           LOC=LOC+1                                                 ZIN00239
LN 0242      43   ALPH(LOC)=CHARTM(I)                                       ZIN00240
LN 0243           CALL ZHOPPR(VALUE,NSTOP,1,-1)                             ZIN00241
LN 0244           RETURN                                                    ZIN00242
```

```
LN 0245          END                                                ZIN00243

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZINITL

          NO ERRORS
```

```
LN 0001          SUBROUTINE ZTRANX(IFR,ITO)                                    ZTR00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,  ZTR00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZTR00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZTR00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZTR00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZTR00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    ZTR00007
LN 0008         1NIFMAX,INTZEI                                                    ZTR00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                       ZTR00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),     ZTR00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                      ZTR00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                 ZTR00012
LN 0013          COMMON// DATAN(330)                                              ***
LN 0014          COMMON// DATA(3700)                                              ZTR00014
LN 0015          DIMENSION IPROG(3700)                                            ZTR00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                   ZTR00016
LN 0017          DIMENSION ITRAN(80)                                              ZTR00017
LN 0018          DIMENSION IHOLD(80)                                              ZTR00018
LN 0019          DIMENSION ZMK(80)                                                ZTR00019
LN 0020          DIMENSION NTYPER(12)                                             ZTR00020
LN 0021     C*****SUBROUTINE TO TRANSLATE EXPRESSIONS INTO PSEUDO-MACHINE CODE    ZTR00021
LN 0022     C*****CLEAR ITRAN                                                     ZTR00022
LN 0023     C*****DELETE ALL DOLLAR SIGNS                                         ZTR00023
LN 0024          IDOL=IFR                                                         ZTR00024
LN 0025          IF((CARDP(IFR+1).NE.DOLSGN).AND.(CARDP(IFR+2).NE.DOLSGN)) GOTO 95 ZTR00025
LN 0026          CALL ZDIGIT(CARDP(IFR),I)                                        ZTR00026
LN 0027          IF(I.LE.10) GOTO 1100                                            ZTR00027
LN 0028      95  DO 116 I=IFR,ITO                                                ZTR00028
LN 0029          IF(CARDP(I).EQ.DOLSGN) GOTO 116                                  ZTR00029
LN 0030          ZMK(IDOL)=CARDP(I)                                               ZTR00030
LN 0031          IDOL=IDOL+1                                                      ZTR00031
LN 0032     116  CONTINUE                                                        ZTR00032
LN 0033          ITO=IDOL-1                                                       ZTR00033
LN 0034          DO 117 I=IFR,ITO                                                 ZTR00034
LN 0035          CARDP(I)=ZMK(I)                                                  ZTR00035
LN 0036     117  CONTINUE                                                        ZTR00036
LN 0037          DO 110 I=IFR,ITO                                                 ZTR00037
LN 0038     110  ITRAN(I)=0                                                      ZTR00038
LN 0039          NERR=0                                                           ZTR00039
LN 0040     C                                                                     ZTR00040
LN 0041     C*****CLEAR NTYPER                                                    ZTR00041
LN 0042          DO 115 I=1,12                                                    ZTR00042
LN 0043     115  NTYPER(I)=0                                                     ZTR00043
LN 0044     C                                                                     ZTR00044
LN 0045     C*****CLEAR ZMK                                                       ZTR00045
LN 0046          DO 120 I=1,80                                                    ZTR00046
LN 0047     120  ZMK(I)=BLANK                                                    ZTR00047
LN 0048     C                                                                     ZTR00048
LN 0049     C*****PROCESS ALL PARENTHESES                                         ZTR00049
LN 0050          DO 300 I=IFR,ITO                                                 ZTR00050
LN 0051          IF(CARDP(I).NE.PARLFT) GOTO 300                                  ZTR00051
LN 0052     C*****LEFT PARENTHESIS FOUND -- MARK IT                               ZTR00052
LN 0053          ITRAN(I)=-20                                                     ZTR00053
LN 0054          IF(I.EQ.ITO) GOTO 290                                            ZTR00054
LN 0055     C*****FIND RIGHT PARENTHESIS                                          ZTR00055
LN 0056          ITOT=0                                                           ZTR00056
LN 0057          DO 150 IRT=I,ITO                                                 ZTR00057
LN 0058          IF(CARDP(IRT).EQ.PARLFT) ITOT=ITOT+1                             ZTR00058
LN 0059          IF(CARDP(IRT).EQ.PARRT) ITOT=ITOT-1                              ZTR00059
LN 0060          IF(ITOT.EQ.0) GOTO 155                                           ZTR00060
LN 0061     150  CONTINUE                                                        ZTR00061
```

```
LN 0062          GOTO 290                                                    ZTR00062
LN 0063     C*****RIGHT PARENTHESIS IS IN POSITIO  IRT                        ZTR00063
LN 0064      155  IF(I.EQ.IFR) GOTO 200                                      ZTR00064
LN 0065     C*****SEE IF PRECEEDING CHARACTER IS ALPHABETIC                   ZTR00065
LN 0066          CALL ZALPH(CARDP(I-1),IM1)                                  ZTR00066
LN 0067          IF(IM1.LE.26) GOTO 210                                      ZTR00067
LN 0068     C*****THESE PARENTHESES ARE NOT FOR A SUBSCRIPT -- MARK RIGHT ONE ZTR00068
LN 0069      200  ITRAN(IRT)=-21                                            ZTR00069
LN 0070          GOTO 300                                                    ZTR00070
LN 0071     C                                                                ZTR00071
LN 0072     C*****PRECEEDING CHARACTERS IS ALPHABETIC -- CHECK PREVIOUS ONE   ZTR00072
LN 0073      210  IF(I.EQ.IFR+1) GOTO 220                                    ZTR00073
LN 0074          CALL ZALPH(CARDP(I-2),IM2)                                  ZTR00074
LN 0075          IF(IM2.LE.26) GOTO 224                                      ZTR00075
LN 0076     C*****THIS IS A SUPSCRIPT EXPRESSION - MARK RIGHT PARENTHESIS AND INSERTZTR00076
LN 0077      220  ITRAN(IRT)=-8                                             ZTR00077
LN 0078          ITRAN(I-1)=IM1                                              ZTR00078
LN 0079     C*****RESERVE SPACE IF NOT ALREADY RESERVED                       ZTR00079
LN 0080          IF(IPROG(IM1).NE.2) GOTO 221                                ZTR00080
LN 0081          IF(MERKER(IM1,1).NE.0) GOTO 221                             ZTR00081
LN 0082          IPROG(IM1)=122                                              ZTR00082
LN 0083          MERKER(IM1,1)=10                                            ZTR00083
LN 0084          MERKER(IM1,2)=10                                            ZTR00084
LN 0085     C*****FIND AND MARK COMMA IF PRESENT                              ZTR00085
LN 0086      221  ITOT=0                                                     ZTR00086
LN 0087          DO 222 ICOM=I,IPT                                           ZTR00087
LN 0088          IF(CARDP(ICOM).EQ.PARLFT) ITOT=ITOT+1                       ZTR00088
LN 0089          IF(CARDP(ICOM).EQ.PARRT) ITOT=ITOT-1                        ZTR00089
LN 0090          IF((CARDP(ICOM).EQ.COMMA).AND.(ITOT.EQ.1)) GOTO 223         ZTR00090
LN 0091      222  CONTINUE                                                   ZTR00091
LN 0092     C*****NONE FOUND                                                 ZTR00092
LN 0093          GOTO 300                                                    ZTR00093
LN 0094     C*****MARK COMMA WITH VARIABLE NUMBER                            ZTR00094
LN 0095      223  ITRAN(ICOM)=IM1                                           ZTR00095
LN 0096          GOTO 300                                                    ZTR00096
LN 0097     C                                                                ZTR00097
LN 0098     C*****THIS IS A FUNCTION -- CHECK FIRST CHARACTER                 ZTR00098
LN 0099      224  IF(I.EQ.IFR+2) GOTO 230                                    ZTR00099
LN 0100          CALL ZALPH(CARDP(I-3),IM3)                                  ZTR00100
LN 0101          IF(IM3.LE.26) GOTO 240                                      ZTR00101
LN 0102     C*****TWO-CHARACTER NAME FOUND                                   ZTR00102
LN 0103      230  NTYPER(1)=1                                               ZTR00103
LN 0104          ZMK(I-2)=ALPH(1)                                            ZTR00104
LN 0105          ZMK(I-1)=ALPH(1)                                            ZTR00105
LN 0106          ITRAN(I-1)=-50                                              ZTR00106
LN 0107          ITRAN(I-2)=-50                                              ZTR00107
LN 0108          ITRAN(IRT)=-21                                              ZTR00108
LN 0109          GOTO 300                                                    ZTR00109
LN 0110     C                                                                ZTR00110
LN 0111     C*****FUNCTION FOUND -- MARK NAME                                ZTR00111
LN 0112      240  ITRAN(I-1)=-50                                            ZTR00112
LN 0113          ITRAN(I-2)=-50                                              ZTR00113
LN 0114          ITRAN(I-3)=-50                                              ZTR00114
LN 0115     C*****FIND FUNCTION                                              ZTR00115
LN 0116          IF(IM3.NE.3) GOTO 241                                       ZTR00116
LN 0117          ITRAN(IPT)=-10                                              ZTR00117
LN 0118          GOTO 300                                                    ZTR00118
LN 0119      241  IF(IM3.NE.20) GOTO 242                                     ZTR00119
LN 0120          ITRAN(IRT)=-11                                              ZTR00120
LN 0121          GOTO 300                                                    ZTR00121
LN 0122      242  IF(IM3.NE.5) GOTO 243                                      ZTR00122
```

```
LN 0123           ITRAN(IRT)=-13                                          ZTR00123
LN 0124           GOTO 300                                                ZTR00124
LN 0125      243  IF(IM3.NE.12) GOTO 244                                  ZTR00125
LN 0126           ITRAN(IRT)=-15                                          ZTR00126
LN 0127           GOTO 300                                                ZTR00127
LN 0128      244  IF(IM3.NE. 9) GOTO 245                                  ZTR00128
LN 0129           ITRAN(IRT)=-17                                          ZTR00129
LN 0130           GOTO 300                                                ZTR00130
LN 0131      245  IF(IM3.NE.18) GOTO 246                                  ZTR00131
LN 0132           ITRAN(IRT)=-18                                          ZTR00132
LN 0133           GOTO 300                                                ZTR00133
LN 0134      246  IF((IM3.NE.19).OR.(IM2.NE.9)) GOTO 247                  ZTR00134
LN 0135           ITRAN(IRT)=-9                                           ZTR00135
LN 0136           GOTO 300                                                ZTR00136
LN 0137      247  IF((IM3.NE.1).OR.(IM2.NE.20)) GOTO 248                  ZTR00137
LN 0138           ITRAN(IRT)=-12                                          ZTR00138
LN 0139           GOTO 300                                                ZTR00139
LN 0140      248  IF((IM3.NE.1).OR.(IM2.NE.2)) GOTO 249                   ZTR00140
LN 0141           ITRAN(IRT)=-14                                          ZTR00141
LN 0142           GOTO 300                                                ZTR00142
LN 0143      249  IF((IM3.NE.19).OR.(IM2.NE.17)) GOTO 250                 ZTR00143
LN 0144           ITRAN(IRT)=-16                                          ZTR00144
LN 0145           GOTO 300                                                ZTR00145
LN 0146     C                                                             ZTR00146
LN 0147     C*****ILLEGAL FUNCTION NAME                                   ZTR00147
LN 0148      250  NTYPEP(1)=1                                             ZTR00148
LN 0149           ZMK(I-3)=ALPH(1)                                        ZTR00149
LN 0150           ZMK(I-2)=ALPH(1)                                        ZTR00150
LN 0151           ZMK(I-1)=ALPH(1)                                        ZTR00151
LN 0152           ITRAN(IRT)=-21                                          ZTR00152
LN 0153           GOTO 300                                                ZTR00153
LN 0154     C                                                             ZTR00154
LN 0155     C*****MISSING RIGHT PARENTHESIS                               ZTR00155
LN 0156      290  NTYPER(2)=1                                             ZTR00156
LN 0157           ZMK(I)=ALPH(2)                                          ZTR00157
LN 0158      300  CONTINUE                                                ZTR00158
LN 0159     C                                                             ZTR00159
LN 0160     C*****CHECK TO SEE IF ALL RIGHT PARENTHESES HAVE BEEN MARKED  ZTR00160
LN 0161           DO 310 I=IFR,ITO                                        ZTR00161
LN 0162           IF((CARDP(I).NE.PARRT).OR.(ITRAN(I).NE.0)) GOTO 310     ZTR00162
LN 0163           NTYPER(3)=1                                             ZTR00163
LN 0164           ZMK(I)=ALPH(3)                                          ZTR00164
LN 0165           ITRAN(I)=-21                                            ZTR00165
LN 0166      310  CONTINUE                                                ZTR00166
LN 0167     C                                                             ZTR00167
LN 0168     C*****CHECK TO SEE IF ALL COMMAS HAVE BEEN MARKED             ZTR00168
LN 0169           DO 315 I=IFR,ITO                                        ZTR00169
LN 0170           IF((CARDP(I).NE.COMMA).OR.(ITRAN(I).NE.0)) GOTO 315     ZTR00170
LN 0171           NTYPER(4)=1                                             ZTR00171
LN 0172           ZMK(I)=ALPH(4)                                          ZTR00172
LN 0173      315  CONTINUE                                                ZTR00173
LN 0174     C                                                             ZTR00174
LN 0175     C*****CHECK FOR NON-SUBSCRIPTED VARIABLE NAMES                ZTR00175
LN 0176           DO 400 I=IFR,ITO                                        ZTR00176
LN 0177           IF(ITRAN(I).NE.0) GOTO 400                              ZTR00177
LN 0178           CALL ZALPH(CARDP(I),J)                                  ZTR00178
LN 0179           IF(J.GT.26) GOTO 400                                    ZTR00179
LN 0180     C*****CHARACTER IS ALPHABETIC                                 ZTR00180
LN 0181           IF(I.EQ.ITO) GOTO 380                                   ZTR00181
LN 0182           IF(ITRAN(I+1).NE.0) GOTO 380                            ZTR00182
LN 0183           CALL ZDIGIT(CARDP(I+1),K)                               ZTR00183
```

```
LN 0184           IF(K.GT.10) GOTO 380                                    ZTR00184
LN 0185     C*****TWO-CHARACTER NAME FOUND                                ZTR00185
LN 0186           ITRAN(I)=J+(26*(K-1))+53                                ZTR00186
LN 0187           ITRAN(I+1)=-50                                          ZTR00187
LN 0188           GOTO 400                                                ZTR00188
LN 0189     C                                                             ZTR00189
LN 0190     C*****ONE-CHARACTER NAME FOUND                                ZTR00190
LN 0191      380  ITRAN(I)=J                                             ZTR00191
LN 0192      400  CONTINUE                                               ZTR00192
LN 0193     C                                                             ZTR00193
LN 0194     C*****TRANSLATE OPERATORS                                     ZTR00194
LN 0195           LAST=-20                                                ZTR00195
LN 0196           ILAST=0                                                 ZTR00196
LN 0197           DO 500 I=IFR,ITO                                        ZTR00197
LN 0198           IF(ITRAN(I).EQ.-50) GOTO 500                            ZTR00198
LN 0199           IF(ITRAN(I).EQ.0) GOTO 410                              ZTR00199
LN 0200     C*****CHARACTER HAS BEEN TRANSLATED -- RECORD IT              ZTR00200
LN 0201           LAST=ITRAN(I)                                           ZTR00201
LN 0202           ILAST=I                                                 ZTR00202
LN 0203           GOTO 500                                                ZTR00203
LN 0204     C                                                             ZTR00204
LN 0205     C*****CHARACTER NOT TRANSLATED -- CHECK FOR OPERATORS         ZTR00205
LN 0206      410  IF(CARDP(I).EQ.PLUS) GOTO 420                          ZTR00206
LN 0207           IF(CARDP(I).EQ.CMINUS) GOTO 430                         ZTR00207
LN 0208           IF(CARDP(I).EQ.ASTRSK) GOTO 440                         ZTR00208
LN 0209           IF(CARDP(I).EQ.SLASH) GOTO 450                          ZTR00209
LN 0210           IF(CARDP(I).EQ.EXSIGN) GOTO 460                         ZTR00210
LN 0211     C*****CHARACTER NOT AN OPERATOR -- RECORD IT                   ZTR00211
LN 0212           LAST=ITRAN(I)                                           ZTR00212
LN 0213           ILAST=I                                                 ZTR00213
LN 0214           GOTO 500                                                ZTR00214
LN 0215     C                                                             ZTR00215
LN 0216     C*****CHARACTER IS PLUS                                       ZTR00216
LN 0217      420  IF((LAST.NE.-20).AND.(CARDP(I-1).NE.COMMA)) GOTO 422    ZTR00217
LN 0218     C*****FORM IS (+ OR ,+ DELETE +                               ZTR00218
LN 0219           ITRAN(I)=-50                                            ZTR00219
LN 0220           GOTO500                                                 ZTR00220
LN 0221     C*****INSERT + OPERATOR                                       ZTR00221
LN 0222      422  ITRAN(I)=-1                                            ZTR00222
LN 0223           GOTO 490                                                ZTR00223
LN 0224     C                                                             ZTR00224
LN 0225     C*****CHARACTER IS MINUS                                      ZTR00225
LN 0226      430  IF((LAST.NE.-20).AND.(CARDP(I-1).NE.COMMA)) GOTO 432    ZTR00226
LN 0227     C*****FORM IS (- OR ,- INSERT UNARY MINUS                     ZTR00227
LN 0228           ITRAN(I)=-6                                             ZTR00228
LN 0229           GOTO 500                                                ZTR00229
LN 0230     C*****INSERT - OPERATOR                                       ZTR00230
LN 0231      432  ITRAN(I)=-2                                            ZTR00231
LN 0232           GOTO 490                                                ZTR00232
LN 0233     C                                                             ZTR00233
LN 0234     C*****CHARACTER IS *                                          ZTR00234
LN 0235      440  IF(LAST.NE.-3) GOTO 442                                ZTR00235
LN 0236     C*****PREVIOUS CHARACTER WAS *                                ZTR00236
LN 0237           ITRAN(TLAST)=-50                                        ZTR00237
LN 0238           ITRAN(I)=-5                                             ZTR00238
LN 0239           GOTO 495                                                ZTR00239
LN 0240     C*****PREVIOUS CHARACTER WAS NOT *                            ZTR00240
LN 0241      442  ITRAN(I)=-3                                            ZTR00241
LN 0242           GOTO 490                                                ZTR00242
LN 0243     C*****CHARACTER IS EXSIGN (+)                                 ZTR00243
LN 0244      460  ITRAN(I)=-5                                            ZTR00244
```

```
LN 0245          GOTO 490                                                ZTR00245
LN 0246     C                                                            ZTR00246
LN 0247     C*****CHARACTER IS /                                         ZTR00247
LN 0248      450  ITRAN(I)=-4                                           ZTR00248
LN 0249     C                                                            ZTR00249
LN 0250     C*****CHECK FOR DOUBLE OPERATOR                              ZTR00250
LN 0251      490  IF(LAST.GT.-1) GOTO 495                               ZTR00251
LN 0252          IF((LAST.LT.-6).AND.(LAST.NE.-20)) GOTO 495             ZTR00252
LN 0253     C*****DOUBLE OPERATOR FOUND                                  ZTR00253
LN 0254          NTYPER(5)=1                                             ZTR00254
LN 0255          DO 492 IZQ=ILAST,I                                      ZTR00255
LN 0256      492  ZMK(IZQ)=ALPH(5)                                       ZTR00256
LN 0257     C*****POST THIS IN LAST                                      ZTR00257
LN 0258      495  LAST=ITRAN(I)                                         ZTR00258
LN 0259          ILAST=I                                                 ZTR00259
LN 0260      500  CONTINUE                                              ZTR00260
LN 0261     C                                                            ZTR00261
LN 0262     C*****CHECK FOR OPERATOR FOLLOWED BY A RIGHT PARENTHESIS     ZTR00262
LN 0263          ITM=ITO-1                                               ZTR00263
LN 0264          IF(ITM.LT.IFR) GOTO 508                                 ZTR00264
LN 0265          DO 505 I=IFR,ITM                                        ZTR00265
LN 0266          IF((ITRAN(I).GT.-1).OR.(ITRAN(I).LT.-6)) GOTO 505       ZTR00266
LN 0267          IF(CARDP(I+1).NE.PARRT) GOTO 505                        ZTR00267
LN 0268     C*****OPERATOR IS FOLLOWED BY A RIGHT PARENTHESIS            ZTR00268
LN 0269          NTYPER(5)=1                                             ZTR00269
LN 0270          ZMK(I)=ALPH(5)                                          ZTR00270
LN 0271          ZMK(I+1)=ALPH(5)                                        ZTR00271
LN 0272      505  CONTINUE                                              ZTR00272
LN 0273     C                                                            ZTR00273
LN 0274     C*****INSERT NUMBERS                                         ZTR00274
LN 0275      508  DO 600 I=IFR,ITO                                       ZTR00275
LN 0276          IF(ITRAN(I).NE.0) GOTO 600                              ZTR00276
LN 0277     C*****CHARACTER NOT TRANSLATED                               ZTR00277
LN 0278          DO 510 J=I,ITO                                          ZTR00278
LN 0279          IF(ITRAN(J).NE.0) GOTO 520                              ZTR00279
LN 0280      510  CONTINUE                                              ZTR00280
LN 0281          J=ITO+1                                                 ZTR00281
LN 0282      520  JM=J-1                                                 ZTR00282
LN 0283          IF(I.NE.JM) GOTO 523                                    ZTR00283
LN 0284     C*****SINGLE DIGIT FOUND                                     ZTR00284
LN 0285          CALL ZDIGIT(CARDP(I),JXYZ)                              ZTR00285
LN 0286          IF(JXYZ.GT.10) GOTO 524                                 ZTR00286
LN 0287          ITRAN(I)=314+(JXYZ-1)                                   ZTR00287
LN 0288          GOTO 551                                                ZTR00288
LN 0289      523  IXM=I                                                  ZTR00289
LN 0290          CALL ZCONVN(IXM,JM,FNUM)                                ZTR00290
LN 0291          IF(FNUM.GE.0.) GOTO 550                                 ZTR00291
LN 0292     C*****NON-NUMERIC CHARACTER FOUND                            ZTR00292
LN 0293      524  DO 525 IZQ=I,JM                                        ZTR00293
LN 0294          IF(ZMK(IZQ).NE.BLANK) GOTO 525                          ZTR00294
LN 0295          NTYPER(6)=1                                             ZTR00295
LN 0296          ZMK(IZQ)=ALPH(6)                                        ZTR00296
LN 0297      525  CONTINUE                                              ZTR00297
LN 0298          GOTO 570                                                ZTR00298
LN 0299     C                                                            ZTR00299
LN 0300     C*****LOOK FOR NUMBER                                        ZTR00300
LN 0301      550  INM=INEXT-1                                           ZTR00301
LN 0302          IF(FNUM.EQ.0.) GOTO 5508                                ZTR00302
LN 0303          IF(INM.LT.324) GOTO 5508                                ZTR00303
LN 0304          DO 5505 IQ=324,INM                                      ZTR00304
LN 0305          IF(DATA(IQ).EQ.FNUM) GOTO 5510                          ZTR00305
```

```
LN 0306     5505 CONTINUE                                                    ZTR00306
LN 0307    C*****NUMBER FOUND -- INSERT IN NEXT POSITION                     ZTR00307
LN 0308     5508 DATA(INEXT)=FNUM                                           ZTR00308
LN 0309          ITRAN(I)=INEXT                                             ZTR00309
LN 0310          INEXT=INEXT+1                                              ZTR00310
LN 0311          GOTO 551                                                   ZTR00311
LN 0312     5510 ITRAN(I)=IQ                                                ZTR00312
LN 0313    C                                                                ZTR00313
LN 0314    C*****CHECK FOR PARENTHESES FOLLOWING                            ZTR00314
LN 0315      551  IF(JM.EQ.ITO) GOTO 570                                    ZTR00315
LN 0316           IF(CARDP(J).NE.PARLFT) GOTO 570                           ZTR00316
LN 0317    C*****NUMBER IS FOLLOWED BY A LEFT PARENTHESIS                   ZTR00317
LN 0318           NTYPER(7)=1                                               ZTR00318
LN 0319           DO 552 IZQ=I,J                                            ZTR00319
LN 0320      552  ZMK(IZQ)=ALPH(7)                                          ZTR00320
LN 0321    C                                                                ZTR00321
LN 0322    C*****DELETE REMAINING POSITIONS                                 ZTR00322
LN 0323      570  IP=I+1                                                    ZTR00323
LN 0324           IF(IP.GT.JM) GOTO 600                                     ZTR00324
LN 0325           DO 580 K=IP,JM                                            ZTR00325
LN 0326      580  ITRAN(K)=-50                                              ZTR00326
LN 0327    C                                                                ZTR00327
LN 0328      600  CONTINUE                                                  ZTR00328
LN 0329    C                                                                ZTR00329
LN 0330    C*****CHECK FOR DANGLING OPERATOR                                ZTR00330
LN 0331           IF((ITRAN(ITO).GT.-1).OR.(ITRAN(ITO).LT.-6)) GOTO 700     ZTR00331
LN 0332    C*****DANGLING OPERATOR FOUND                                    ZTR00332
LN 0333           NTYPER(8)=1                                               ZTR00333
LN 0334           ZMK(ITO)=ALPH(8)                                          ZTR00334
LN 0335    C                                                                ZTR00335
LN 0336    C*****CHECK FOR ADJACENT ADDRESSES                               ZTR00336
LN 0337      700  DO 790 I=IFR,ITO                                          ZTR00337
LN 0338           IF(ITRAN(I).EQ.-50) GOTO 790                              ZTR00338
LN 0339           IF(I.EQ.IFR) GOTO 780                                     ZTR00339
LN 0340           IF(ITRAN(I).LT.0) GOTO 780                                ZTR00340
LN 0341    C*****ADDRESS FOUND                                              ZTR00341
LN 0342           IF(CARDP(ILAST).NE.PARRT) GOTO 750                        ZTR00342
LN 0343           IF(CARDP(I).EQ.COMMA) GOTO 750                            ZTR00343
LN 0344           NTYPER(9)=1                                               ZTR00344
LN 0345           DO 741 IZQ=ILAST,I                                        ZTR00345
LN 0346      741  ZMK(IZQ)=ALPH(9)                                          ZTR00346
LN 0347           GOTO 780                                                  ZTR00347
LN 0348      750  IF(LAST.LE.-1) GOTO 780                                   ZTR00348
LN 0349           IF(CARDP(ILAST).EQ.COMMA) GOTO 780                        ZTR00349
LN 0350           IF(CARDP(I).EQ.COMMA) GOTO 780                            ZTR00350
LN 0351    C*****TWO ADDRESSES FOUND                                        ZTR00351
LN 0352           IF((LAST.GT.313).OR.(ITRAN(I).GT.313)) GOTO 760           ZTR00352
LN 0353           NTYPER(10)=1                                              ZTR00353
LN 0354           DO 755 IZQ=ILAST,I                                        ZTR00354
LN 0355      755  ZMK(IZQ)=ALPH(10)                                         ZTR00355
LN 0356           GOTO 780                                                  ZTR00356
LN 0357      760  IF((LAST.LE.313).OR.(ITRAN(I).LE.313)) GOTO 770           ZTR00357
LN 0358           NTYPER(11)=1                                              ZTR00358
LN 0359           DO 765 IZQ=ILAST,I                                        ZTR00359
LN 0360      765  ZMK(IZQ)=ALPH(11)                                         ZTR00360
LN 0361           GOTO 780                                                  ZTR00361
LN 0362      770  NTYPER(12)=1                                              ZTR00362
LN 0363           DO 775 IZQ=ILAST,I                                        ZTR00363
LN 0364      775  ZMK(IZQ)=ALPH(12)                                         ZTR00364
LN 0365    C                                                                ZTR00365
LN 0366      780  LAST=ITRAN(I)                                             ZTR00366
```

```
LN 0367           ILAST=I                                                    ZTR00367
LN 0368     790   CONTINUE                                                   ZTR00368
LN 0369     C                                                                ZTR00369
LN 0370     C*****SEE IF EXPRESSION IS ACCEPTABLE                            ZTR00370
LN 0371           NERTOT=0                                                   ZTR00371
LN 0372           DO 810 I=1,12                                              ZTR00372
LN 0373     810   NERTOT=NERTOT+NTYPER(I)                                    ZTR00373
LN 0374           IF(NERTOT.EQ.0) GOTO 900                                   ZTR00374
LN 0375     C                                                                ZTR00375
LN 0376     C*****EXPRESSION CONTAINS ERRORS                                 ZTR00376
LN 0377           NERRS=NERRS+1                                              ZTR00377
LN 0378           NN=0                                                       ZTR00378
LN 0379           NERROR=28                                                  ZTR00379
LN 0380           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                         ZTR00380
LN 0381           WRITE(IWC,825) (CARDP(I),I=IFR,ITO)                        ZTR00381
LN 0382     825   FORMAT(25X,8041)                                           ZTR00382
LN 0383           WRITE(IWC,825) (ZMK(I),I=IFR,ITO)                          ZTR00383
LN 0384           DO 831 I=1,12                                              ZTR00384
LN 0385           IF(NTYPER(I).NE.1) GOTO 831                                ZTR00385
LN 0386           NERROR=I                                                   ZTR00386
LN 0387           NN=2                                                       ZTR00387
LN 0388           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                         ZTR00388
LN 0389     831   CONTINUE                                                   ZTR00389
LN 0390           RETURN                                                     ZTR00390
LN 0391     C                                                                ZTR00391
LN 0392     C*****EXPRESSION ACCEPTED -- SET UP POLISH STRING                ZTR00392
LN 0393     900   IPROG(INREG)=-1                                            ZTR00393
LN 0394           NSTK=INREG-2                                               ZTR00394
LN 0395           NHLD=1                                                     ZTR00395
LN 0396           DO 1000 I=IFR,ITO                                          ZTR00396
LN 0397           IF(ITRAN(I).EQ.-50) GOTO 1000                              ZTR00397
LN 0398           IF(CARDP(I).EQ.COMMA) GOTO 930                             ZTR00398
LN 0399           IF(ITRAN(I).LT.0) GOTO 910                                 ZTR00399
LN 0400     C*****ADDRESS FOUND -- INSERT IN STRING                          ZTR00400
LN 0401           IPROG(NSTK)=ITRAN(I)                                       ZTR00401
LN 0402           NSTK=NSTK-1                                                ZTR00402
LN 0403     C*****CHECK FOR UNSUBSCRIPTED REFERENCES TO ONE-CHARACTER NAMES  ZTR00403
LN 0404           IF(ITRAN(I).GT.26) GOTO 1000                               ZTR00404
LN 0405           IF(I.EQ.ITO) GOTO 909                                      ZTR00405
LN 0406           IF(CARDP(I+1).EQ.PARLFT) GOTO 1000                         ZTR00406
LN 0407     909   IPROG(NSTK)=-7                                             ZTR00407
LN 0408           NSTK=NSTK-1                                                ZTR00408
LN 0409           GOTO 1000                                                  ZTR00409
LN 0410     C                                                                ZTR00410
LN 0411     910   IF(CARDP(I).NE.PARRT) GOTO 920                             ZTR00411
LN 0412     C*****RIGHT PARENTHESIS FOUND                                    ZTR00412
LN 0413           IPR=0                                                      ZTR00413
LN 0414           IGT=1                                                      ZTR00414
LN 0415           GOTO 970                                                   ZTR00415
LN 0416     915   IF(ITRAN(I).EQ.-21) GOTO 1000                              ZTR00416
LN 0417     C******PARENTHESIS CARRIES OPERATORS -- INSERT                   ZTR00417
LN 0418           IPROG(NSTK)=ITRAN(I)                                       ZTR00418
LN 0419           NSTK=NSTK-1                                                ZTR00419
LN 0420           GOTO 1000                                                  ZTR00420
LN 0421     C                                                                ZTR00421
LN 0422     920   IF(ITRAN(I).EQ.-20) GOTO 925                               ZTR00422
LN 0423     C*****OPERATOR FOUND                                             ZTR00423
LN 0424           IF((ITRAN(I).EQ.-1).OR.(ITRAN(I).EQ.-2)) IPR=1             ZTR00424
LN 0425     C---- WENN GEWUENSCHT WIRD, DASS DIE VORZEICHEN (UNARY OPERATORS) ZTR00425
LN 0426     C---- VOR DER EXPONENTIATION UND NACH DEN FUNKTIONEN VERARBEITET  ZTR00426
LN 0427     C----, SIND DIE FOLGENDEN BEIDEN KARTEN DURCH                    ZTR00427
```

```
LN 0428     C      IF((ITRAN(I).EQ.-3).OR.(ITRAN(I).EQ.-4)) IPR=2                    ZTR00428
LN 0429     C      IF((ITRAN(I).EQ.-5).OR.(ITRAN(I).EQ.-6)) IPR=3                    ZTR00429
LN 0430     C---- ZU ERSETZEN, DAS C IN DER ERSTEN SPALTE MUSS DAZU ENTFERNT WERDEN.ZTR00430
LN 0431           IF((ITRAN(I).EQ.-3).OR.(ITRAN(I).EQ.-4).OR.(ITRAN(I).EQ.-6)) IPR=2ZTR00431
LN 0432           IF(ITRAN(I).EQ.-5) IPR=3                                          ZTR00432
LN 0433           IGT=2                                                             ZTR00433
LN 0434           GOTO 970                                                          ZTR00434
LN 0435      925  IHOLD(NHLD)=ITRAN(I)                                             ZTR00435
LN 0436           NHLD=NHLD+1                                                       ZTR00436
LN 0437           GOTO 1000                                                         ZTR00437
LN 0438     C                                                                       ZTR00438
LN 0439     C*****COMMA FOUND -- UNSTACK AND INSERT ROW MULTIPLIER                  ZTR00439
LN 0440      930  IPR=1                                                             ZTR00440
LN 0441           IGT=3                                                             ZTR00441
LN 0442           GOTO 970                                                          ZTR00442
LN 0443      935  IPROG(NSTK)=ITRAN(I)+27                                           ZTR00443
LN 0444           IPROG(NSTK-1)=-3                                                  ZTR00444
LN 0445           NSTK=NSTK-2                                                       ZTR00445
LN 0446           IHOLD(NHLD)=-1                                                    ZTR00446
LN 0447           NHLD=NHLD+1                                                       ZTR00447
LN 0448           GOTO 1000                                                         ZTR00448
LN 0449     C                                                                       ZTR00449
LN 0450     C*****ROUTINE TO UNSTACK DOWN TO FIRST LEFT PARENTHESIS                 ZTR00450
LN 0451     C*****IPR=0 -- UNSTACK ALL AND THROW AWAY LEFT PARENTHESIS              ZTR00451
LN 0452     C*****IPR=1 -- UNSTACK ALL BUT RETAIN LEFT PARENTHESIS                  ZTR00452
LN 0453     C*****IPR=2 -- UNSTACK * / AND **                                       ZTR00453
LN 0454     C*****IPR=3 -- UNSTACK ** ONLY                                          ZTR00454
LN 0455      970  NO=NHLD-1                                                        ZTR00455
LN 0456           IF(NO.EQ.0) GOTO 980                                              ZTR00456
LN 0457           DO 975 IX=1,NO                                                    ZTR00457
LN 0458           LOC=NHLD-IX                                                       ZTR00458
LN 0459           IF(IHOLD(LOC).NE.-20) GOTO 971                                    ZTR00459
LN 0460     C*****LEFT PARENTHESIS FOUND                                            ZTR00460
LN 0461           IF(IPR.EQ.0) IHOLD(LOC)=+1                                        ZTR00461
LN 0462           GOTO 980                                                          ZTR00462
LN 0463     C---- WENN GEWUENSCHT WIRD, DASS DIE VORZEICHEN (UNARY OPERATORS)        ZTR00463
LN 0464     C---- VOR DER EXPONENTIATION UND NACH DEN FUNKTIONEN VERARBEITET         ZTR00464
LN 0465     C---- WERDEN, IST DIE DIESEM TEXT FOLGENDE KARTE DURCH                   ZTR00465
LN 0466     C 971 IF((IHOLD(LOC).LE.-5).AND.(IPR.LE.3)) GOTO 972                    ZTR00466
LN 0467     C---- ZU ERSETZEN, DAS C IN DER ERSTEN SPALTE MUSS DAZU ENTFERNT WERDEN.ZTR00467
LN 0468      971  IF(IHOLD(LOC).EQ.-5) GOTO 972                                     ZTR00468
LN 0469           IF((IHOLD(LOC).LE.-3).AND.(IPR.LE.2)) GOTO 972                    ZTR00469
LN 0470           IF((IHOLD(LOC).LE.-1).AND.(IPR.LE.1)) GOTO 972                    ZTR00470
LN 0471     C*****DO NOT UNSTACK ITEM                                               ZTR00471
LN 0472           GOTO 975                                                          ZTR00472
LN 0473     C*****UNSTACK ITEM INTO STRING                                          ZTR00473
LN 0474      972  IPROG(NSTK)=IHOLD(LOC)                                           ZTR00474
LN 0475           NSTK=NSTK-1                                                       ZTR00475
LN 0476     C*****INSERT POSITIVE NUMBER IN POSITION VACATED                        ZTR00476
LN 0477           IHOLD(LOC)=+1                                                     ZTR00477
LN 0478      975  CONTINUE                                                          ZTR00478
LN 0479      980  GOTO(915,925,935), IGT                                            ZTR00479
LN 0480      1000 CONTINUE                                                         ZTR00480
LN 0481     C                                                                       ZTR00481
LN 0482     C*****DUMP REMAINING OPERATORS                                          ZTR00482
LN 0483           NO=NHLD-1                                                         ZTR00483
LN 0484           IF(NO.EQ.0) GOTO 1040                                             ZTR00484
LN 0485           DO 1010 IX=1,NO                                                   ZTR00485
LN 0486           LOC=NHLD-IX                                                       ZTR00486
LN 0487           IF(IHOLD(LOC).GE.0) GOTO 1010                                     ZTR00487
LN 0488           IPROG(NSTK)=IHOLD(LOC)                                            ZTR00488
```

```
LN 0489           NSTK=NSTK-1                                                ZTR00489
LN 0490      1010 CONTINUE                                                   ZTR00490
LN 0491     C*****FIND AND INSERT LENGTH OF EXPRESSION                       ZTR00491
LN 0492      1040 LNGTH=INREG-NSTK-2                                         ZTR00492
LN 0493           IPROG(INREG-1)=LNGTH                                       ZTR00493
LN 0494     C                                                                ZTR00494
LN 0495     C*****CHECK EXPRESSION                                           ZTR00495
LN 0496           NOP=0                                                      ZTR00496
LN 0497           DO 1090 IO=1,LNGTH                                         ZTR00497
LN 0498           LOC=INREG-1-IO                                             ZTR00498
LN 0499           IF(IPROG(LOC).LT.0) GOTO 1060                              ZTR00499
LN 0500     C*****OPERAND FOUND                                              ZTR00500
LN 0501           NOP=NOP+1                                                  ZTR00501
LN 0502           GOTO 1090                                                  ZTR00502
LN 0503      1060 IF((IPROG(LOC).GE.-5).OR.(IPROG(LOC).EQ.-8)) GOTO 1070     ZTR00503
LN 0504     C*****UNARY OPERATOR FOUND                                       ZTR00504
LN 0505           IF(NOP.LT.1) GOTO 1100                                     ZTR00505
LN 0506           GOTO 1090                                                  ZTR00506
LN 0507     C*****BINARY OPERATOR FOUND                                      ZTR00507
LN 0508      1070 IF(NOP.LT.2) GOTO 1100                                     ZTR00508
LN 0509           NOP=NOP-1                                                  ZTR00509
LN 0510      1090 CONTINUE                                                   ZTR00510
LN 0511           IF(NOP.NE.1) GOTO 1100                                     ZTR00511
LN 0512     C*****EXPRESSION IS OK                                           ZTR00512
LN 0513           INREG=NSTK                                                 ZTR00513
LN 0514           RETURN                                                     ZTR00514
LN 0515     C*****EXPRESSION IS NOT OK                                       ZTR00515
LN 0516      1100 NERRS=NERRS+1                                              ZTR00516
LN 0517           NN=0                                                       ZTR00517
LN 0518           NERROR=29                                                  ZTR00518
LN 0519           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                         ZTR00519
LN 0520           WRITE(IWC,825) (CARDP(I),I=IFR,ITO)                        ZTR00520
LN 0521           RETURN                                                     ZTR00521
LN 0522           END                                                        ZTR00522

     USAST FORTRAN DIAGNOSTIC RESULTS FOR ZTRANX

          NO ERRORS
```

ZCONVN

```
LN 0001           SUBROUTINE ZCONVN(IFR,ITO,FNUM)                                ZC000001
LN 0002     C*****SUBROUTINE TO CONVERT A POSITIVE NUMBER IN CARDP(IFR) THROUGH   ZC000002
LN 0003     C*****CARDP(ITO) TO A REAL NUMBER FNUM                               ZC000003
LN 0004     C                                                                    ZC000004
LN 0005           COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   ZC000005
LN 0006          1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZC000006
LN 0007          2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZC000007
LN 0008          3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,TIMAGE,NPRI,NIMAGE, ZC000008
LN 0009          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZFI     ZC000009
LN 0010           COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     ZC000010
LN 0011          1NIFMAX,INTZEI                                                     ZC000011
LN 0012           COMMON// CARDT(80),MERKER(26,2),CARP(140),                        ZC000012
LN 0013          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),      ZC000013
LN 0014          1IRET(20),XXX(4),NFILE(25,3)                                       ZC000014
LN 0015           COMMON// ISTLST(340),LISTST(340)                                  ZC000015
LN 0016           COMMON// DATAN(330)                                               ***
LN 0017           COMMON// DATA(3700)                                               ZC000017
LN 0018           DIMENSION IPROG(3700)                                             ZC000018
LN 0019           EQUIVALENCE (DATA(1),IPROG(1))                                    ZC000019
LN 0020     C                                                                       ZC000020
LN 0021           IF(ITO.LT.IFR) GOTO 190                                           ZC000021
LN 0022           FNUM=0.                                                           ZC000022
LN 0023           IDEC=0                                                            ZC000023
LN 0024           DO 120 I=IFR,ITO                                                  ZC000024
LN 0025           CALL ZDTGIT(CARDP(I),J)                                           ZC000025
LN 0026           IF(J.LE.10) GOTO 118                                              ZC000026
LN 0027           IF(CARDP(I).NE.DECMAL) GOTO 190                                   ZC000027
LN 0028           IDEC=I                                                            ZC000028
LN 0029           GOTO 120                                                          ZC000029
LN 0030      118  DIG=J-1                                                          ZC000030
LN 0031           FNUM=(FNUM*10.)+DIG                                               ZC000031
LN 0032      120  CONTINUE                                                         ZC000032
LN 0033           IF(IDEC.EQ.0) RETURN                                              ZC000033
LN 0034           IDIFF=ITO-IDEC                                                    ZC000034
LN 0035           FNUM=FNUM/(10.**IDIFF)                                            ZC000035
LN 0036           RETURN                                                            ZC000036
LN 0037     C                                                                       ZC000037
LN 0038     C*****ILLEGAL NUMBER                                                    ZC000038
LN 0039      190  FNUM=-1.                                                         ZC000039
LN 0040           RETURN                                                            ZC000040
LN 0041           END                                                               ZC000041

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZCONVN

         NO ERRORS
```

```
LN 0001          SUBROUTINE MATTRA                                                  MAT00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,       MAT00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NEPRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,MAT00003
LN 0004         2NUMBUF,PAPLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  MAT00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NORI,NIMAGE, MAT00005
LN 0006         4NERUS,NCARD,MAXTMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     MAT00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,NIPMAX,      MAT00007
LN 0008         1NIFMAX,INTZEI                                                     MAT00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                        MAT00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),      MAT00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                       MAT00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                  MAT00012
LN 0013          COMMON// DATAN(330)                                               ***
LN 0014          COMMON// DATA(3700)                                               MAT00014
LN 0015          DIMENSION IPROG(3700)                                             MAT00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                    MAT00016
LN 0017    C**** MAT A=B+C       CODE=-23                                          MAT00017
LN 0018    C**** MAT A=B*C       CODE=-24                                          MAT00018
LN 0019    C**** MAT A=B-C       CODE=-25                                          MAT00019
LN 0020    C**** MAT A=TRN(X)    CODE=-26                                          MAT00020
LN 0021    C**** MAT READ NO REDIMENSION CODE=-28                                  MAT00021
LN 0022    C**** MAT READ DIMENSION CODE=-28                                       MAT00022
LN 0023    C**** MAT READ REDIMENSION CODE=-29                                     MAT00023
LN 0024    C**** MAT INPUT NO REDIMENSION CODE=-46                                 MAT00024
LN 0025    C**** MAT INPUT DIMENSION CODE=-46                                      MAT00025
LN 0026    C**** MAT INPUT REDIMENSION CODE=-47                                    MAT00026
LN 0027    C**** MAT PRINT NUMERIC CODE=-31                                        MAT00027
LN 0028    C**** MAT PRINT NUMERIC REDIMENSION CODE=-32                            MAT00028
LN 0029    C**** MAT PRINT ALPHANUMERIC CODE=-33                                   MAT00029
LN 0030    C**** MAT PRINT ALPHANUMERIC REDIMENSION CODE=-34                       MAT00030
LN 0031    C**** MAT A=IDN       CODE=-35                                          MAT00031
LN 0032    C**** MAT A=IDN       CODE=-36  REDIMENSION                             MAT00032
LN 0033    C**** MAT A=CON       CODE=-37                                          MAT00033
LN 0034    C**** MAT A=CON       CODE=-38  REDIMENSION                             MAT00034
LN 0035    C**** MAT A=ZER       CODE=-39                                          MAT00035
LN 0036    C**** MAT A=ZER       CODE=-40  REDIMENSION                             MAT00036
LN 0037    C**** MAT A=(EXPRESSION)     CODE=-41                                   MAT00037
LN 0038    C**** MAT A=B  CODE=-42                                                 MAT00038
LN 0039    C**** MAT A=B-BINEARY OPERATOR-(EXPRESSION   CODE=-43                   MAT00039
LN 0040    C**** MAT A=(EXPRESSION)-BINEARY OPERATOR-B CODE=-43/-43                MAT00040
LN 0041    C**** MAT A=INV(B)     CODE=-71                                         MAT00041
LN 0042          IBEGST=IBEGST+3                                                   MAT00042
LN 0043    C*****MOVE STATEMENT FROM CARDP TO CARDT                                MAT00043
LN 0044          DO 9 IUM=IBEGST,LNGCRP                                            MAT00044
LN 0045       9  CARDT(IUM)=CARDP(IUM)                                            MAT00045
LN 0046          MCOM=0                                                            MAT00046
LN 0047          IF((CARDP(IBEGST).EQ.ALPH(7)).AND.(CARDP(IBEGST+1).EQ.ALPH(5)))   MAT00047
LN 0048         1 MCOM=7                                                           MAT00048
LN 0049          IF((CARDP(IBEGST).EQ.ALPH(16)).AND.(CARDP(IBEGST+1).EQ.ALPH(21))) MAT00049
LN 0050         1 MCOM=8                                                           MAT00050
LN 0051          IF(MCOM.EQ.0) GOTO 6                                              MAT00051
LN 0052    C----AUF DIE FOLGENDEN ZEILEN LN=0053-0054 WIRD IM TEXT BEZUG GENOMMENvv*** 
LN 0053          CALL ZFILE(MCOM)                                                  MAT00052
LN 0054          RETURN                                                            MAT00053
LN 0055       6  IF((CARDP(IBEGST).EQ.ALPH(18)).AND.(CARDP(IBEGST+1).EQ.ALPH(5)).  MAT00054
LN 0056         1AND.(CARDP(IBEGST+3).EQ.ALPH(4))) GOTO 11                         MAT00055
LN 0057          IF((CARDT(IBEGST).EQ.ALPH(9)).AND.(CARDT(IBEGST+1).EQ.ALPH(14)))  MAT00056
LN 0058         1 GOTO 13                                                          MAT00057
LN 0059          IF((CARDT(IBEGST).EQ.ALPH(12)).AND.(CARDT(IBEGST+1).EQ.ALPH(5)).  MAT00058
LN 0060         1AND.(CARDT(IBEGST+2).EQ.ALPH(20))) GOTO 111                       MAT00059
LN 0061          IF((CARDT(IBEGST).EQ.ALPH(16)).AND.(CARDT(IBEGST+1).EQ.ALPH(18))  MAT00060
```

```
LN 0062          1.AND.(CARDT(IBEGST+2).EQ.ALPH(9))) GOTO 1111                  MAT00061
LN 0063           IBEGST=IBEGST-3                                               MAT00062
LN 0064           GOTO 111                                                      MAT00063
LN 0065     1000  NERRS=NERRS+1                                                 MAT00064
LN 0066           NN=0                                                          MAT00065
LN 0067           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                            MAT00066
LN 0068           RETURN                                                        MAT00067
LN 0069     C     COMMAND IS READ                                               MAT00068
LN 0070      11   NDOL=4                                                        MAT00069
LN 0071           MOP=-28                                                       MAT00070
LN 0072           GOTO 12                                                       MAT00071
LN 0073     C     COMMAND IS INPUT   CODE IS -46                                MAT00072
LN 0074      13   NDOL=5                                                        MAT00073
LN 0075           MOP=-46                                                       MAT00074
LN 0076      12   ITANF=IBEGST+NDOL                                             MAT00075
LN 0077           IF(LNGCRP.LT.ITANF) GOTO 1000                                 MAT00076
LN 0078     706   ITOT=0                                                        MAT00077
LN 0079           DO 699 J=ITANF,LNGCRP                                         MAT00078
LN 0080           IF(CARDT(J).EQ.DOLSGN) GOTO 698                               MAT00079
LN 0081           IF(CARDT(J).EQ.PARLFT) ITOT=ITOT+1                            MAT00080
LN 0082           IF(CARDT(J).EQ.PARRT)ITOT=ITOT-1                              MAT00081
LN 0083           IF((CARDT(J).EQ.COMMA).AND.(ITOT.EQ.0)) GOTO 707              MAT00082
LN 0084     699   CONTINUE                                                      MAT00083
LN 0085           J=LNGCRP+1                                                    MAT00084
LN 0086           GOTO 707                                                      MAT00085
LN 0087     698   CARDT(J)=CARDT(J-1)                                           MAT00086
LN 0088           ITANF=J                                                       MAT00087
LN 0089           GOTO 706                                                      MAT00088
LN 0090     707   CALL ZALPH(CARDT(ITANF),I)                                    MAT00089
LN 0091           IF(ITOT.NE.0) GOTO 318                                        MAT00090
LN 0092           IF(I.LE.26) GOTO 704                                          MAT00091
LN 0093           LOCN=ITANF                                                    MAT00092
LN 0094           GOTO 2100                                                     MAT00093
LN 0095     657   NERROR=32                                                     MAT00094
LN 0096           I1=ITANF                                                      MAT00095
LN 0097           I2=IC                                                         MAT00096
LN 0098           GOTO 1000                                                     MAT00097
LN 0099     704   IF(J-2.GT.ITANF) GOTO 702                                     MAT00098
LN 0100     C**** MAT READ OR INPUT -- NO REDIMENSION AND NO DIMENSION          MAT00099
LN 0101      740  IPROG(INREG)=MOP                                              MAT00100
LN 0102           IPROG(INREG-1)=I                                              MAT00101
LN 0103           CALL 7KLAM(ITANF,ITANF,I)                                     MAT00102
LN 0104     701   INREG=INREG-2                                                 MAT00103
LN 0105           ITANF=J+1                                                     MAT00104
LN 0106           IF(J.LT.LNGCRP) GOTO 706                                      MAT00105
LN 0107           RETURN                                                        MAT00106
LN 0108     C**** MAT READ OR INPUT WITH REDIMENSION                            MAT00107
LN 0109      702  IF(MEMKER(I,1).EQ.0) GOTO 720                                MAT00108
LN 0110           CALL 7KLAM(ITANF+1,J-1,I)                                     MAT00109
LN 0111           IPROG(INREG)=MOP-1                                            MAT00110
LN 0112           IPROG(INREG-1)=I                                              MAT00111
LN 0113           GOTO 701                                                      MAT00112
LN 0114     C**** MAT READ OR INPUT WITH DIMENSION                              MAT00113
LN 0115      720  MEMCRP=LNGCRP                                                 MAT00114
LN 0116           LOC=ITANF+1                                                   MAT00115
LN 0117           LNGCRP=J-1                                                    MAT00116
LN 0118           GOTO 658                                                      MAT00117
LN 0119      721  LNGCRP=MEMCRP                                                 MAT00118
LN 0120           GOTO 740                                                      MAT00119
LN 0121     C                                                                   MAT00120
LN 0122     C**** COMMAND IS MAT PRINT                                          MAT00121
```

```
LN 0123    1111  IF(LNGCRP.LT.IBEGST+5) GOTO 1000                                  MAT00122
LN 0124          ITANF=IBEGST+5                                                    MAT00123
LN 0125     310  NDOL=0                                                           MAT00124
LN 0126     312  ITOT=0                                                           MAT00125
LN 0127          DO 311 J=ITANF,LNGCRP                                             MAT00126
LN 0128          IF(CARDP(J).EQ.DOLSGN) GOTO 314                                   MAT00127
LN 0129          IF(CARDP(J).EQ.PARRT) ITOT=ITOT+1                                 MAT00128
LN 0130          IF(CARDP(J).EQ.PARLFT) ITOT=ITOT-1                                MAT00129
LN 0131          IF((CARDP(J).EQ.COMMA).AND.(ITOT.EQ.0)) GOTO 3135                 MAT00130
LN 0132          IF((CARDP(J).EQ.PUCO).AND.(ITOT.EQ.0)) GOTO 317                   MAT00131
LN 0133     311  CONTINUE                                                         MAT00132
LN 0134          J=LNGCRP+1                                                        MAT00133
LN 0135          GOTO 3135                                                         MAT00134
LN 0136     313  CALL ZALPH(CARDP(ITANF),IA)                                      MAT00135
LN 0137          MOP=-31                                                           MAT00136
LN 0138          IF(IA.GT.26) GOTO 319                                             MAT00137
LN 0139          IF(NDOL.EQ.1) MOP=-33                                             MAT00138
LN 0140    C*****NUMERIC MAT PRINT =-31                                            MAT00139
LN 0141    C*****ALPHANUMERIC MAT PRINT =-33                                       MAT00140
LN 0142          IF(J.EQ.ITANF+1) GOTO 315                                         MAT00141
LN 0143          IF(J.EQ.ITANF+3) GOTO 316                                         MAT00142
LN 0144    C*****MAT PRINT WITH REDIMENSION                                        MAT00143
LN 0145          MOP=MOP-1                                                         MAT00144
LN 0146          CALL ZKLAM(ITANF+1,J-1,IA)                                        MAT00145
LN 0147     315  IPROG(INREG)=MOP                                                  MAT00146
LN 0148          IPROG(INREG-1)=IA                                                 MAT00147
LN 0149          INREG=INREG-2                                                     MAT00148
LN 0150          ITANF=J+1                                                         MAT00149
LN 0151          CALL ZKLAM(ITANF,ITANF,IA)                                        MAT00150
LN 0152          IF(ITANF.GT.LNGCRP) RETURN                                        MAT00151
LN 0153          GOTO 310                                                          MAT00152
LN 0154     317  IPROG(INREG)=-69                                                  MAT00153
LN 0155    3136  INREG=INREG-1                                                     MAT00154
LN 0156          IF(ITOT.NE.0) GOTO 318                                            MAT00155
LN 0157          GOTO 313                                                          MAT00156
LN 0158    3135  IPROG(INREG)=-67                                                  MAT00157
LN 0159          GOTO 3136                                                         MAT00158
LN 0160     316  IC=J-1                                                           MAT00159
LN 0161          GOTO 657                                                          MAT00160
LN 0162     314  CARDP(J)=CARDP(J-1)                                              MAT00161
LN 0163          ITANF=J                                                           MAT00162
LN 0164          NDOL=1                                                            MAT00163
LN 0165          GOTO 312                                                          MAT00164
LN 0166     318  NERROR=53                                                         MAT00165
LN 0167          I1=ITANF                                                          MAT00166
LN 0168          I2=J                                                              MAT00167
LN 0169          GOTO 1000                                                         MAT00168
LN 0170     319  LOCN=ITANF                                                       MAT00169
LN 0171          GOTO 2100                                                         MAT00170
LN 0172    C                                                                       MAT00171
LN 0173    C*****COMMAND IS MAT LET                                                MAT00172
LN 0174     111  IBEGST=IBEGST+3                                                  MAT00173
LN 0175          DO 2000 LOC=IBEGST,LNGCRP                                         MAT00174
LN 0176          IF(CARDT(LOC).EQ.EQUALS) GOTO 2001                                MAT00175
LN 0177     2000 CONTINUE                                                         MAT00176
LN 0178          NERROR=33                                                         MAT00177
LN 0179          GOTO 1000                                                         MAT00178
LN 0180     2001 IF((CARDT(LOC+1).EQ.ALPH(9)).AND.(CARDT(LOC+2).EQ.ALPH(4)))      MAT00179
LN 0181         1 GOTO 500                                                         MAT00180
LN 0182          IF((CARDP(LOC+1).EQ.ALPH(9)).AND.(CARDP(LOC+3).EQ.ALPH(22)))     MAT00181
LN 0183         1 GOTO 8000                                                        MAT00182
```

```
LN 0184           IF((CARDT(LOC+1).EQ.ALPH(20)).AND.(CARDT(LOC+2).EQ.ALPH(18)))     MAT00183
LN 0185          1 GOTO 600                                                         MAT00184
LN 0186           IF((CARDT(LOC+1).EQ.ALPH(3)).AND.(CARDT(LOC+2).EQ.ALPH(15)))      MAT00185
LN 0187          1 GOTO 610                                                         MAT00186
LN 0188           IF((CARDT(LOC+1).EQ.ALPH(26)).AND.(CARDT(LOC+2).EQ.ALPH(5)))      MAT00187
LN 0189          1 GOTO 650                                                         MAT00188
LN 0190           IF(CARDT(LOC+1).EQ.PARLFT) GOTO 2003                               MAT00189
LN 0191           IF(CARDT(LNGCRP).NE.PARRT) GOTO 2030                               MAT00190
LN 0192     C**** COMMAND IS A=B+(EXPRESSION)                                        MAT00191
LN 0193           IANF=1                                                             MAT00192
LN 0194           I=LOC+2                                                            MAT00193
LN 0195           LOC1=LOC+3                                                         MAT00194
LN 0196           LOC2=LNGCRP                                                        MAT00195
LN 0197           IAS=LOC+1                                                          MAT00196
LN 0198           GOTO 2025                                                          MAT00197
LN 0199      2030 IF(CARDT(LOC+1).NE.QUOTE) GOTO 2040                                MAT00198
LN 0200     C**** COMMAND IS MAT A=STRING                                            MAT00199
LN 0201           IF(CARDT(LOC-1).EQ.DOLSGN) GOTO 2041                               MAT00200
LN 0202           NERROR=4                                                           MAT00201
LN 0203           NN=1                                                               MAT00202
LN 0204           CALL COMERR(NERROR,I1,I2,CARDT(LOC-1),X2,NN)                       MAT00203
LN 0205      2041 CALL STRING(LOC+2,LNGCRP-1,IX)                                     MAT00204
LN 0206           IPROG(INREG)=-1                                                    MAT00205
LN 0207           IPROG(INREG-1)=2                                                   MAT00206
LN 0208           IPROG(INREG-2)=-20                                                 MAT00207
LN 0209           IPROG(INREG-3)=IX                                                  MAT00208
LN 0210           IPROG(INREG-4)=-41                                                 MAT00209
LN 0211           IB=INREG-5                                                         MAT00210
LN 0212      5555 LOCN=LOC-1                                                         MAT00211
LN 0213           IF(CARDT(LOCN).EQ.DOLSGN) LOCN=LOC-2                               MAT00212
LN 0214           IF((LOCN-IBEGST+1).EQ.1) GOTO 5455                                 MAT00213
LN 0215           NERROR=51                                                          MAT00214
LN 0216           I1=IBEGST                                                          MAT00215
LN 0217           I2=LOC-1                                                           MAT00216
LN 0218           GOTO 1000                                                          MAT00217
LN 0219      5455 CALL ZALPH(CARDT(LOCN),IA)                                         MAT00218
LN 0220           IF(IA.GT.26) GOTO 2100                                             MAT00219
LN 0221           CALL ZKLAM(LOC,LOC,IA)                                             MAT00220
LN 0222           IPROG(IB)=IA                                                       MAT00221
LN 0223           INREG=IB-1                                                         MAT00222
LN 0224           RETURN                                                             MAT00223
LN 0225     C**** NO EXPRESSION ON THE RIGHT                                         MAT00224
LN 0226      2040 IF(CARDT(LOC+2).EQ.ASTRSK) GOTO 700                                MAT00225
LN 0227           IF(LNGCRP.LE.LOC+2) GOTO 2102                                      MAT00226
LN 0228           IF(LNGCRP.GT.LOC+3) GOTO 2112                                      MAT00227
LN 0229     C**** COMMAND IS MAT A=B+C OR MAT A=B-C                                  MAT00228
LN 0230           IF(CARDP(LOC+2).EQ.PLUS) MOP=-23                                   MAT00229
LN 0231           IF(CARDP(LOC+2).EQ.CMINUS) MOP=-25                                 MAT00230
LN 0232           IPROG(INREG)=MOP                                                   MAT00231
LN 0233           IANF=1                                                             MAT00232
LN 0234           IB=1                                                               MAT00233
LN 0235           LOCN=LOC-1                                                         MAT00234
LN 0236           GOTO 2221                                                          MAT00235
LN 0237      2112 NERROR=34                                                          MAT00236
LN 0238           GOTO 1000                                                          MAT00237
LN 0239      2101 CALL ZTRANX(LOC+1,LNGCRP)                                          MAT00238
LN 0240           IPROG(INREG)=-41                                                   MAT00239
LN 0241           IB=INREG-1                                                         MAT00240
LN 0242           GOTO 5555                                                          MAT00241
LN 0243      2003 IF(CARDT(LNGCRP).EQ.PARRT) GOTO 2101                               MAT00242
LN 0244           IANF=-1                                                            MAT00243
```

```
LN 0245          I=LNGCRP-1                                                    MAT00244
LN 0246          LOC1=LOC+1                                                    MAT00245
LN 0247          LOC2=LNGCRP-2                                                 MAT00246
LN 0248          IAS=LNGCRP                                                    MAT00247
LN 0249     2025 IF(CARDT(I).NE.PLUS) GOTO 2012                                MAT00248
LN 0250          MOP=1                                                         MAT00249
LN 0251          GOTO 2020                                                     MAT00250
LN 0252     2012 IF(CARDT(I).NE.CMINUS) GOTO 2013                              MAT00251
LN 0253          MOP=2                                                         MAT00252
LN 0254          GOTO 2020                                                     MAT00253
LN 0255     2013 IF(CARDT(I).NE.SLASH) GOTO 2014                               MAT00254
LN 0256          MOP=3                                                         MAT00255
LN 0257          GOTO 2020                                                     MAT00256
LN 0258     2014 IF(CARDT(I).NE.ASTRSK) GOTO 2040                              MAT00257
LN 0259          IF((I.EQ.LNGCRP-1).AND.(CARDT(I-1).EQ.ASTRSK)) GOTO 2015      MAT00258
LN 0260          IF((I.EQ.LOC+2).AND.(CARDT(I+1).EQ.ASTRSK)) GOTO 2017         MAT00259
LN 0261          MOP=4                                                         MAT00260
LN 0262          GOTO 2020                                                     MAT00261
LN 0263     2015 LOC2=LOC2-1                                                   MAT00262
LN 0264          GOTO 2018                                                     MAT00263
LN 0265     2017 LOC1=LOC1+1                                                   MAT00264
LN 0266     2018 MOP=5                                                         MAT00265
LN 0267          GOTO 2020                                                     MAT00266
LN 0268    C*****IDENTITY MATRIX WITH REDIMENSION CODE=-36                     MAT00267
LN 0269      500 MOP=-35                                                       MAT00268
LN 0270          GOTO 660                                                      MAT00269
LN 0271    C**** MATRIX=1 (CON) REDIMENSION=-38                                MAT00270
LN 0272      610 MOP=-37                                                       MAT00271
LN 0273          GOTO 660                                                      MAT00272
LN 0274    C**** MATRIX=0 (ZERO) REDIMENSION=-40                               MAT00273
LN 0275      650 MOP=-39                                                       MAT00274
LN 0276          GOTO 660                                                      MAT00275
LN 0277    C*****MAT A=B                                                       MAT00276
LN 0278     2102 IF((CARDT(LOC-1).NE.DOLSGN).AND.(CARDT(LOC+2).NE.DOLSGN))GOTO 2136MAT00277
LN 0279          IF((CARDT(LOC-1).EQ.DOLSGN).AND.(CARDT(LOC+2).EQ.DOLSGN))GOTO 2136MAT00278
LN 0280          IF(CARDT(LOC-1).EQ.DOLSGN) NERROR=5                           MAT00279
LN 0281          IF(CARDT(LOC+2).EQ.DOLSGN) NERROR=4                           MAT00280
LN 0282          NN=1                                                          MAT00281
LN 0283          CALL COMERR(NERROR,I1,I2,CARDT(LOC-2),X2,NN)                  MAT00282
LN 0284     2136 CALL ZALPH(CARDP(LOC+1),IA)                                   MAT00283
LN 0285          IF(IA.GT.26) GOTO 2100                                        MAT00284
LN 0286          CALL ZKLAM(LOC,LOC,IA)                                        MAT00285
LN 0287          IPROG(INREG)=-42                                              MAT00286
LN 0288          IPROG(INREG-1)=IA                                             MAT00287
LN 0289          IB=INREG-2                                                    MAT00288
LN 0290          GOTO 5555                                                     MAT00289
LN 0291    C**** NO ACCEPTABLE MAT LET COMMAND                                 MAT00290
LN 0292     2100 X1=CARDT(LOCN)                                                MAT00291
LN 0293          NERROR=43                                                     MAT00292
LN 0294          GOTO 1000                                                     MAT00293
LN 0295    C*****WITH EXPRESSION                                               MAT00294
LN 0296     2020 CALL ZTRANX(LOC1,LOC2)                                        MAT00295
LN 0297          IB=2                                                          MAT00296
LN 0298          IPROG(INREG)=-43                                              MAT00297
LN 0299          IPROG(INREG-1)=MOP                                            MAT00298
LN 0300          IF(IANF.EQ.1) GOTO 2072                                       MAT00299
LN 0301          IPROG(INREG-2)=-43                                            MAT00300
LN 0302          INREG=INREG-1                                                 MAT00301
LN 0303     2072 LOCN=LOC-1                                                    MAT00302
LN 0304     2221 IF(CARDT(LOCN).NE.DOLSGN) GOTO 2021                           MAT00303
LN 0305          LOCN=LOCN-1                                                   MAT00304
```

```
LN 0306          NERROR=5                                                    MAT00305
LN 0307          NN=1                                                        MAT00306
LN 0308          CALL COMERR(NERROR,I1,I2,CARDT(LOCN),X2,NN)                 MAT00307
LN 0309     2021 CALL ZALPH(CARDT(LOCN),IA)                                  MAT00308
LN 0310          IF(IA.GT.26) GOTO 2100                                      MAT00309
LN 0311          CALL ZKLAM(LOCN,LOCN,IA)                                    MAT00310
LN 0312          LOC1=INREG-IB                                               MAT00311
LN 0313          IPROG(LOC1)=IA                                              MAT00312
LN 0314          IF(IB.EQ.3) GOTO 2022                                       MAT00313
LN 0315          IB=IB+1                                                     MAT00314
LN 0316          IC=IA                                                       MAT00315
LN 0317          IF(IANF.EQ.-1) GOTO 2076                                    MAT00316
LN 0318          LOCN=LOCN+2                                                 MAT00317
LN 0319          GOTO 2021                                                   MAT00318
LN 0320     2076 LOCN=IAS                                                    MAT00319
LN 0321          GOTO 2221                                                   MAT00320
LN 0322     2022 INREG=INREG-4                                               MAT00321
LN 0323          IF((MERKER(IA,2).NE.0).AND.(MERKER(IC,2).EQ.0)) GOTO 2023   MAT00322
LN 0324          IF((MERKER(IA,2).EQ.0).AND.(MERKER(IC,2).NE.0)) GOTO 2026   MAT00323
LN 0325          RETURN                                                      MAT00324
LN 0326     2026 IB=IC                                                       MAT00325
LN 0327          IC=IA                                                       MAT00326
LN 0328          IA=IB                                                       MAT00327
LN 0329     2023 I1=IC                                                       MAT00328
LN 0330          I2=IA                                                       MAT00329
LN 0331          NERROR=36                                                   MAT00330
LN 0332          GOTO 1000                                                   MAT00331
LN 0333    C**** MAT LET WITH CON OR IDN OR ZER                              MAT00332
LN 0334     660  LOCN=LOC-1                                                  MAT00333
LN 0335          IF(CARDT(LOCN).NE.DOLSGN) GOTO 669                          MAT00334
LN 0336          LOCN=LOC-2                                                  MAT00335
LN 0337          NERROR=5                                                    MAT00336
LN 0338          NN=1                                                        MAT00337
LN 0339          CALL COMERR(NERROR,I1,I2,CARDT(LOCN),X2,NN)                 MAT00338
LN 0340     669  CALL ZALPH(CARDT(LOCN),I)                                   MAT00339
LN 0341          IF(I.GT.26) GOTO 2100                                       MAT00340
LN 0342          IF(LNGCRP.GT.LOC+3) GOTO 652                                MAT00341
LN 0343    C**** NO REDIMENSION                                              MAT00342
LN 0344     656  IPROG(INREG)=MOP                                            MAT00343
LN 0345          IPROG(INREG-1)=I                                            MAT00344
LN 0346          CALL ZKLAM(LOC-1,LOC-1,I)                                   MAT00345
LN 0347     659  INREG=INREG-2                                               MAT00346
LN 0348          RETURN                                                      MAT00347
LN 0349    C**** DIMENSION OR REDIMENSION                                    MAT00348
LN 0350     652  IF(CARDT(LNGCRP).EQ.PARRT) GOTO 658                        MAT00349
LN 0351          ITANF=LOC+1                                                 MAT00350
LN 0352          IC=LNGCRP                                                   MAT00351
LN 0353          GOTO 657                                                    MAT00352
LN 0354    C**** REDIMENSION                                                 MAT00353
LN 0355     655  CALL ZKLAM(LOC+4,LNGCRP,I)                                  MAT00354
LN 0356          IPROG(INREG)=MOP-1                                          MAT00355
LN 0357          IPROG(INREG-1)=I                                            MAT00356
LN 0358          GOTO 659                                                    MAT00357
LN 0359    C**** DIMENSION                                                   MAT00358
LN 0360     658  IF(MERKER(I,1).NE.0) GOTO 655                              MAT00359
LN 0361          DO 661 IA=LOC,LNGCRP                                        MAT00360
LN 0362          IF(CARDT(IA).EQ.COMMA) GOTO 662                             MAT00361
LN 0363          IF(CARDT(IA).EQ.PARLFT) LOCN=IA+1                           MAT00362
LN 0364     661  CONTINUE                                                    MAT00363
LN 0365          IA=LNGCRP                                                   MAT00364
LN 0366    C**** SINGLE SUBSCRIPT                                            MAT00365
```

```
LN 0367     662  CALL ZCONVN(LOCN,TA-1,FNUM)                                    MAT00366
LN 0368          IF(FNUM.GE.1.) GOTO 663                                        MAT00367
LN 0369          I1=LOCN                                                        MAT00368
LN 0370          I2=TA-1                                                        MAT00369
LN 0371     6651 NERROR=54                                                      MAT00370
LN 0372          GOTO 1000                                                      MAT00371
LN 0373     663  MERKER(I,1)=FNUM                                               MAT00372
LN 0374          IF(IA.LT.LNGCRP) GOTO 664                                      MAT00373
LN 0375          MERKER(I,2)=0                                                  MAT00374
LN 0376          IPROG(I)=FNUM+2.                                               MAT00375
LN 0377          GOTO 6561                                                      MAT00376
LN 0378     C**** DOUBLE SUBSCRIPTED                                            MAT00377
LN 0379      664 CALL ZCONVN(IA+1,LNGCRP-1,FNUM)                                 MAT00378
LN 0380          IF(FNUM.GE.1.) GOTO 665                                        MAT00379
LN 0381          I1=IA+1                                                        MAT00380
LN 0382          I2=LNGCRP-1                                                    MAT00381
LN 0383          GOTO 6651                                                      MAT00382
LN 0384     665  MERKER(I,2)=FNUM                                               MAT00383
LN 0385          IPROG(I)=(MERKER(I,1)+1)*(MERKER(I,2)+1)+1                     MAT00384
LN 0386          DATA(I+27)=FNUM+1.                                             MAT00385
LN 0387     6561 IF((MOP.EQ.-28).OR.(MOP.EQ.-46)) GOTO 721                      MAT00386
LN 0388          IF((MOP.EQ.-35).AND.(MERKER(I,1).NE.MERKER(I,2))) GOTO 666     MAT00387
LN 0389          GOTO 656                                                       MAT00388
LN 0390     666  X1=ALPH(I)                                                     MAT00389
LN 0391          NERROR=37                                                      MAT00390
LN 0392          GOTO 1000                                                      MAT00391
LN 0393     C*****MATRIX MULTIPLICATION                                         MAT00392
LN 0394     C*****CODE=-24                                                      MAT00393
LN 0395     C*****FIRST ADDRESS IS IN INREG-1                                   MAT00394
LN 0396     C*****SECOND ADDRESS IS IN INREG-2                                  MAT00395
LN 0397     C*****THIRD ADDRESS IS IN INREG-3                                   MAT00396
LN 0398     C*****A(I)=ROW VECTOR                                               MAT00397
LN 0399     C*****A(1,I)=ROW VECTOR                                             MAT00398
LN 0400     C*****A(I,1)=COLUMN VECTOR                                          MAT00399
LN 0401      700 IF(LNGCRP.GT.LOC+3) GOTO 7777                                  MAT00400
LN 0402          IANF=IBEGST                                                    MAT00401
LN 0403      300 CALL ZALPH(CARDT(IANF),I)                                      MAT00402
LN 0404          LOCN=IANF                                                      MAT00403
LN 0405          IF(I.GT.26) GOTO 2100                                          MAT00404
LN 0406          IF(IANF.EQ.LNGCRP) GOTO 173                                    MAT00405
LN 0407          IF(IANF.EQ.IBEGST) GOTO 172                                    MAT00406
LN 0408          IF(IANF.EQ.LOC+1) GOTO 102                                     MAT00407
LN 0409          GOTO 7777                                                      MAT00408
LN 0410     172  CALL ZDIGIT(CARDT(IANF+1),J)                                   MAT00409
LN 0411          IF(J.GT.10) GOTO 178                                           MAT00410
LN 0412          IPROG(INREG-1)=I+(26*(J-1))+53                                 MAT00411
LN 0413          IANF=LOC+1                                                     MAT00412
LN 0414          GOTO 300                                                       MAT00413
LN 0415     178  IPROG(INREG-1)=I                                               MAT00414
LN 0416          IANF=LOC+1                                                     MAT00415
LN 0417          IA=I                                                           MAT00416
LN 0418          GOTO 300                                                       MAT00417
LN 0419     102  IANF=LNGCRP                                                    MAT00418
LN 0420          IPROG(INREG-2)=I                                               MAT00419
LN 0421          IB=I                                                           MAT00420
LN 0422          GOTO 300                                                       MAT00421
LN 0423     173  IPROG(INREG-3)=I                                               MAT00422
LN 0424          IC=I                                                           MAT00423
LN 0425          IF((IA.EQ.IB).AND.(IA.EQ.IC)) GOTO 7777                        MAT00424
LN 0426          IPROG(INREG)=-24                                               MAT00425
LN 0427          INREG=INREG-4                                                  MAT00426
```

```
LN 0428           RETURN                                                MAT00427
LN 0429      7777 NERROR=38                                             MAT00428
LN 0430           GOTO 1000                                             MAT00429
LN 0431     C***** TRANSPONATION CODE IS -26                            MAT00430
LN 0432       600 MOP=-26                                              MAT00431
LN 0433       601 CALL ZALPH(CARDT(LNGCRP-1),I)                        MAT00432
LN 0434           IF(I.GT.26) GOTO 2100                                 MAT00433
LN 0435           CALL ZKLAM(LOC,LOC,I)                                 MAT00434
LN 0436           IPROG(INREG-2)=I                                      MAT00435
LN 0437           CALL ZALPH(CARDT(LOC-1),I)                            MAT00436
LN 0438           IF(I.GT.26) GOTO 2100                                 MAT00437
LN 0439           CALL ZKLAM(LOC,LOC,I)                                 MAT00438
LN 0440           IPROG(INREG-1)=I                                      MAT00439
LN 0441           IPROG(INREG)=MOP                                      MAT00440
LN 0442           INREG=INREG-3                                         MAT00441
LN 0443           RETURN                                                MAT00442
LN 0444     C**** INVERSION                                             MAT00443
LN 0445       8000 MOP=-71                                             MAT00444
LN 0446           GOTO 601                                              MAT00445
LN 0447           END                                                   MAT00446
```

```
      USASI FORTRAN DIAGNOSTIC RESULTS FOR MATTRA

          NO ERRORS
```

```
LN 0001          SUBROUTINE ZKLAM(IFR,ITO,MI)                                   ZKL00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMTNUS,COMMA,DECMAL,DOLSGN,EQUALS,   ZKL00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT,  ZKL00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZKL00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRT,NIMAGE,  ZKL00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIN,NSTZEI     ZKL00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,MIRMAX,     ZKL00007
LN 0008         1NIFMAX,INTZEI                                                     ZKL00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                       ZKL00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),     ZKL00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                       ZKL00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                 ZKL00012
LN 0013          COMMON// DATAN(330)                                              ***
LN 0014          COMMON// DATA(3700)                                              ZKL00014
LN 0015          DIMENSION IPROG(3700)                                            ZKL00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                   ZKL00016
LN 0017          NZ=NERRS                                                         ZKL00017
LN 0018          IF(MERKER(MI,1) .NE.0).GOTO 707                                  ZKL00018
LN 0019          NERROR=39                                                        ZKL00019
LN 0020     706  NERRS=NERRS+1                                                    ZKL00020
LN 0021          NN=0                                                             ZKL00021
LN 0022          CALL COMERR(NERROR,I1,I2,ALPH(MI),X2,NN)                         ZKL00022
LN 0023          GOTO 703                                                         ZKL00023
LN 0024     707  IF(IFR.EQ.ITO) RETURN                                            ZKL00024
LN 0025          DO 700 I=IFR,ITO                                                 ZKL00025
LN 0026          IF(CARDT(I).EQ.COMMA) GOTO 709                                   ZKL00026
LN 0027     700  CONTINUE                                                        ZKL00027
LN 0028          I=ITO                                                            ZKL00028
LN 0029          GOTO 710                                                         ZKL00029
LN 0030     709  CALL ZTRANX(I+1,ITO-1)                                           ZKL00030
LN 0031          IPROG(INREG)=-14                                                 ZKL00031
LN 0032          IPROG(INREG-1)=4                                                 ZKL00032
LN 0033          IF(MERKER(MI,2).EQ.0) GOTO 704                                   ZKL00033
LN 0034          INREG=INREG-2                                                    ZKL00034
LN 0035     710  CALL ZTRANX(IFR+1,I-1)                                           ZKL00035
LN 0036          IPROG(INREG)=-14                                                 ZKL00036
LN 0037          IPROG(INREG-1)=3                                                 ZKL00037
LN 0038          INREG=INREG-2                                                    ZKL00038
LN 0039     703  IF(NZ.NE.NERRS) MI=-1                                            ZKL00039
LN 0040          RETURN                                                           ZKL00040
LN 0041     704  NERROR=40                                                        ZKL00041
LN 0042          GOTO 706                                                         ZKL00042
LN 0043          END                                                              ZKL00043

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZKLAM

          NO ERRORS
```

ZLISTE

```
LN 0001          SUBROUTINE ZLISTE(IANF,MOP,IX,MCOM)                                ZLI00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,       ZLI00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST,INEXT,  ZLI00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZLI00004
LN 0005         3IRC,INC,NSTEND,IEXPO,IBEGST,INRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,  ZLI00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZLI00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,ODOPU,IMIRC,SMALL,ISTMAX,NIRMAX,      ZLI00007
LN 0008         1NIFMAX,INTZEI                                                      ZLI00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                         ZLI00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),       ZLI00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                        ZLI00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                   ZLI00012
LN 0013          COMMON// DATAN(330)                                                ***
LN 0014          COMMON// DATA(3700)                                                ZLI00014
LN 0015          DIMENSION IPROG(3700)                                              ZLI00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                     ZLI00016
LN 0017          IF(MCOM.EQ.2) GOTO 311                                             ZLI00017
LN 0018          IB=1                                                               ZLI00018
LN 0019      310 IF(MCOM.LE.3) IB=2                                                 ZLI00019
LN 0020          IPROG(INREG)=MOP                                                   ZLI00020
LN 0021          IPROG(INREG-1)=IX                                                  ZLI00021
LN 0022          INREG=INREG-IB                                                     ZLI00022
LN 0023      311 NTOT=0                                                             ZLI00023
LN 0024          DO 315 LOC=IANF,LNGCRP                                             ZLI00024
LN 0025          IF(CARDP(LOC).EQ.PARLFT)NTOT=NTOT+1                                ZLI00025
LN 0026          IF(CARDP(LOC).EQ.PARRT) NTOT=NTOT-1                                ZLI00026
LN 0027          IF((CARDP(LOC).EQ.COMMA).AND.(NTOT.EQ.0)) GOTO 320                 ZLI00027
LN 0028      315 CONTINUE                                                           ZLI00028
LN 0029          LOC=LNGCRP+1                                                       ZLI00029
LN 0030      320 IEND=LOC-1                                                         ZLI00030
LN 0031          IF(MCOM.EQ.2) GOTO 321                                             ZLI00031
LN 0032          IF(CARDP(IEND).EQ.DOLSGN) IEND=IEND-1                              ZLI00032
LN 0033          IF(IEND.EQ.IANF) GOTO 370                                          ZLI00033
LN 0034          IF(IEND.EQ.IANF+1) GOTO 345                                        ZLI00034
LN 0035     C**** SUBSCRIPTED VARIABLE FOUND                                        ZLI00035
LN 0036      321 IF(CARDP(IANF).EQ.QUOTE) GOTO 322                                  ZLI00036
LN 0037          NZ=NERRS                                                           ZLI00037
LN 0038          CALL ZTRANX(IANF,IEND)                                             ZLI00038
LN 0039          IF(NZ.NE.NERRS) GOTO 330                                           ZLI00039
LN 0040          IF(MCOM.EQ.2) GOTO 385                                             ZLI00040
LN 0041          IF(IPROG(INREG+1).EQ.-8) GOTO 340                                  ZLI00041
LN 0042      330 NERROR=41                                                          ZLI00042
LN 0043      338 NN=0                                                               ZLI00043
LN 0044          CALL COMERR(NERROR,IANF,IEND,X1,X2,NN)                             ZLI00044
LN 0045          NERRS=NERRS+1                                                      ZLI00045
LN 0046          GOTO 385                                                           ZLI00046
LN 0047     C**** SUBSCRIPTED VARIABLE FOUND                                        ZLI00047
LN 0048      340 IPROG(INREG+1)=-19                                                 ZLI00048
LN 0049          GOTO 385                                                           ZLI00049
LN 0050     C**** UNSUBSCRIPTED VARIABLE FOUND -- DOUBLE CHARACTER NAME             ZLI00050
LN 0051      345 CALL ZALPH(CARDP(IANF),K)                                          ZLI00051
LN 0052          IF(K.GT.26) GOTO 330                                               ZLI00052
LN 0053          CALL ZDIGIT(CARDP(IANF+1),L)                                       ZLI00053
LN 0054          IF(L.GT.10) GOTO 330                                               ZLI00054
LN 0055          IPROG(INREG)=-8                                                    ZLI00055
LN 0056          IPROG(INREG-1)=K+(26*(L-1))+53                                     ZLI00056
LN 0057          INREG=INREG-2                                                      ZLI00057
LN 0058          GOTO 385                                                           ZLI00058
LN 0059     C                                                                       ZLI00059
LN 0060     C**** SINGLE CHARACTER NAME FOUND                                       ZLI00060
LN 0061      370 CALL ZALPH(CARDP(IANF),K)                                          ZLI00061
```

```
LN 0062          IF(K.GT.26) GOTO 330                                    ZLI00062
LN 0063          IPROG(INREG)=-9                                         ZLI00063
LN 0064          IPROG(INREG-1)=K                                        ZLI00064
LN 0065          INREG=INREG-2                                           ZLI00065
LN 0066      385 IANF=LOC+1                                              ZLI00066
LN 0067          IF(IANF.GT.LNGCRP) GOTO 386                             ZLI00067
LN 0068          GOTO 310                                                ZLI00068
LN 0069      386 IF(MCOM.NE.2) GOTO 387                                  ZLI00069
LN 0070          IPROG(INREG)=MOP                                        ZLI00070
LN 0071          IPROG(INREG-1)=IX                                       ZLI00071
LN 0072          INREG=INREG-2                                           ZLI00072
LN 0073      387 RETURN                                                  ZLI00073
LN 0074     C*****ALPHANUMERIC CONSTANT FOUND IN A PUT COMMAND           ZLI00074
LN 0075      322 ICT=0                                                   ZLI00075
LN 0076          IF(MCOM.EQ.2) GOTO 405                                  ZLI00076
LN 0077          IF((MCOM.GE.7).OR.(MCOM.LE.9)) GOTO 400                 ZLI00077
LN 0078      405 IF(CARDP(IEND).NE.QUOTE) GOTO 323                       ZLI00078
LN 0079          LOC=IANF                                                ZLI00079
LN 0080          K=INREG                                                 ZLI00080
LN 0081          INREG=INREG-4                                           ZLI00081
LN 0082      324 IANF=LOC+1                                              ZLI00082
LN 0083          LOC=IANF+4                                              ZLI00083
LN 0084          IF(LOC.GE.IEND) LOC=IEND-1                              ZLI00084
LN 0085          CALL STRING(IANF,LOC,IA)                                ZLI00085
LN 0086          L=INREG-ICT                                             ZLI00086
LN 0087          IPROG(L)=IA                                             ZLI00087
LN 0088          ICT=ICT+1                                               ZLI00088
LN 0089          IF(LOC.LT.IEND-1) GOTO 324                              ZLI00089
LN 0090          IPPOG(K)=MOP                                            ZLI00090
LN 0091          IPROG(K-1)=IX                                           ZLI00091
LN 0092          IPROG(K-2)=-16                                          ZLI00092
LN 0093          IPROG(K-3)=ICT                                          ZLI00093
LN 0094          INREG=INREG-ICT                                         ZLI00094
LN 0095          IANF=IEND+2                                             ZLI00095
LN 0096          IF(IANF.GT.LNGCRP) RETURN                               ZLI00096
LN 0097          GOTO 311                                                ZLI00097
LN 0098      323 NERROR=7                                                ZLI00098
LN 0099          GOTO 338                                                ZLI00099
LN 0100      400 NERROR=52                                               ZLI00100
LN 0101          GOTO 338                                                ZLI00101
LN 0102          END                                                     ZLI00102

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZLISTE

          NO ERRORS
```

ZFILE

```
LN 0001          SUBROUTINE ZFILE(MCOM)                                          ZFI00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMTNUS,COMMA,DECMAL,DOLSGN,EQUALS,    ZFI00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT, ZFI00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZFI00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,ITMAGE,NPRI,NIMAGE, ZFI00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZFI00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,NIPMAX,     ZFI00007
LN 0008         1NIFMAX,INTZEI                                                     ZFI00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                      ZFI00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),     ZFI00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                       ZFI00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                ZFI00012
LN 0013          COMMON// DATAN(330)                                             ***
LN 0014          COMMON// DATA(3700)                                             ZFI00014
LN 0015          DIMENSION IPROG(3700)                                           ZFI00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                  ZFI00016
LN 0017          IF(MCOM.NE.-1) GOTO 1                                           ZFI00017
LN 0018          MSTAT=0                                                         ZFI00018
LN 0019     C**** MAXFIL MUST BE EQUAL TO THE FIRST SUBSCRIPT IN NFILE           ZFI00019
LN 0020          NUMFIL=0                                                        ZFI00020
LN 0021          DO 100 I=1,MAXFIL                                               ZFI00021
LN 0022          NFILE(I,1)=-1                                                   ZFI00022
LN 0023          NFILE(I,2)=0                                                    ZFI00023
LN 0024          NFILE(I,3)=0                                                    ZFI00024
LN 0025      100 CONTINUE                                                        ZFI00025
LN 0026          RETURN                                                          ZFI00026
LN 0027     C***  MCOM=1 COMMAND IS OPEN        CODE IS -48                     ZFI00027
LN 0028     C***  MCOM=2 COMMAND IS PUT         CODE IS -49/-50                 ZFI00028
LN 0029     C***  MCOM=3 COMMAND IS GET         CODE IS -51/-52                 ZFI00029
LN 0030     C***  MCOM=4 COMMAND IS RESET       CODE IS -53/-54                 ZFI00030
LN 0031     C***  MCOM=5 COMMAND IS CLOSE       CODE IS -55                     ZFI00031
LN 0032     C***  MCOM=6 COMMAND IS COMMON-FILE  NO OPERATION CODE              ZFI00032
LN 0033     C***  MCOM=7 COMMAND IS MAT GET     CODE IS -58/-59                 ZFI00033
LN 0034     C***  MCOM=8 COMMAND IS MAT PUT     CODE IS -60/-61                 ZFI00034
LN 0035     C***  COMPILATION OF A POINTER POSITION IN A RESET COMMAND  CODE IS -56 ZFI00035
LN 0036     C***  COMPILATION OF A SENTENCE NUMBER IN A STATEMENT CODE IS -57     ZFI00036
LN 0037      1    GOTO(1000,2000,3000,4000,5000,6000,7500,8000),MCOM              ZFI00037
LN 0038     C**** COMMAND IS OPEN                                                ZFI00038
LN 0039      1000 ITANF=IBEGST+4                                                 ZFI00039
LN 0040           MOP=-48                                                        ZFI00040
LN 0041      1200 MSTAT=MSTAT+1                                                  ZFI00041
LN 0042      1100 IF(ITANF.GE.LNGCRP) GOTO 7000                                  ZFI00042
LN 0043      1004 IF(CARDP(ITANF).NE.QUOTE) GOTO 7001                            ZFI00043
LN 0044           DO 1001 I=ITANF,LNGCRP                                         ZFI00044
LN 0045           IF(CARDP(I).EQ.COMMA) GOTO 1002                                ZFI00045
LN 0046      1001 CONTINUE                                                       ZFI00046
LN 0047           I=LNGCRP+1                                                     ZFI00047
LN 0048     C**** SEQUENTIAL TEST                                                ZFI00048
LN 0049      1002 IF(CARDP(I-1).NE.QUOTE) GOTO 1003                              ZFI00049
LN 0050           FNUM=1.1                                                       ZFI00050
LN 0051           FFNUM=0.                                                       ZFI00051
LN 0052           K=I                                                            ZFI00052
LN 0053      1007 CALL STRING(ITANF+1,K-2,IX)                                    ZFI00053
LN 0054           IF(MCOM.NE.6) GOTO 1011                                        ZFI00054
LN 0055     C**** COMMAND IS COMMON-FILE                                         ZFI00055
LN 0056           IF(FFNUM.EQ.0.) GOTO 1028                                      ZFI00056
LN 0057           NN=1                                                           ZFI00057
LN 0058           NERROR=6                                                       ZFI00058
LN 0059           CALL COMERR(NERROR,I1,I2,X1,X2,NN)                             ZFI00059
LN 0060      1028 CALL FINDFI(IX,K)                                              ZFI00060
LN 0061           IF(K.LE.NUMFIL) GOTO 7005                                      ZFI00061
```

```
LN 0062          NUMFIL=NUMFIL+1                                                ZFI00062
LN 0063          IF(NUMFIL.GT.MAXFIL) GOTO 7006                                 ZFI00063
LN 0064          NFILE(NUMFIL,1)=IX                                             ZFI00064
LN 0065          NFILE(NUMFIL,2)=1000                                           ZFI00065
LN 0066          NFILE(NUMFIL,3)=FNUM                                           ZFI00066
LN 0067          IF(NUMFIL.GT.1) NFILE(NUMFIL,2)=NFILE(NUMFIL-1,2)+NFILE(NUMFIL-1,3ZFI00067
LN 0068         1)                                                              ZFI00068
LN 0069          IF(NFILE(NUMFIL,2)+NFILE(NUMFIL,3).GT.MAXSAT) GOTO 7002        ZFI00069
LN 0070          GOTO 1030                                                      ZFI00070
LN 0071     1011 IPROG(INREG)=MOP                                               ZFI00071
LN 0072          IPROG(INREG-1)=IX                                              ZFI00072
LN 0073          IPROG(INREG-2)=FNUM                                            ZFI00073
LN 0074          IPROG(INREG-3)=FFNUM                                           ZFI00074
LN 0075          INREG=INREG-4                                                  ZFI00075
LN 0076     1030 ITANF=I+1                                                      ZFI00076
LN 0077          IF(ITANF.LT.LNGCRP) GOTO 1004                                  ZFI00077
LN 0078          RETURN                                                         ZFI00078
LN 0079     C**** INDEX-SEQUENTIAL                                              ZFI00079
LN 0080      1003 DO 1005 K=ITANF,I                                             ZFI00080
LN 0081           IF(CARDP(K).EQ.PUCO) GOTO 1006                                ZFI00081
LN 0082      1005 CONTINUE                                                      ZFI00082
LN 0083           K=I                                                           ZFI00083
LN 0084      1006 IF(CARDP(K-1).NE.QUOTE) GOTO 7001                             ZFI00084
LN 0085           IF(MCOM.EQ.4) GOTO 1500                                       ZFI00085
LN 0086           CALL ZCONVN(K+1,I-1,FNUM)                                     ZFI00086
LN 0087           IF(FNUM.GT.FLOAT(MAXSAT)) GOTO 7002                           ZFI00087
LN 0088           IF(FNUM.LT.0.) GOTO 7004                                      ZFI00088
LN 0089           IX=I+1                                                        ZFI00089
LN 0090           IF((CARDP(IX).NE.QUOTE).AND.(I.LT.LNGCRP)) GOTO 1027          ZFI00090
LN 0091           FFNUM=0.                                                      ZFI00091
LN 0092           GOTO 1007                                                     ZFI00092
LN 0093      1027 DO 1025 I=IX,LNGCPP                                           ZFI00093
LN 0094           IF(CARDP(I).EQ.COMMA) GOTO 1026                               ZFI00094
LN 0095      1025 CONTINUE                                                      ZFI00095
LN 0096           I=LNGCRP+1                                                    ZFI00096
LN 0097      1026 CALL ZCONVN(IX,I-1,FFNUM)                                     ZFI00097
LN 0098           IF(FFNUM.LT.0.) GOTO 7007                                     ZFI00098
LN 0099           GOTO 1007                                                     ZFI00099
LN 0100     C**** RESET -- INDEX-SEQUENTIAL                                     ZFI00100
LN 0101      1500 IX=K+1                                                        ZFI00101
LN 0102           MOP=-54                                                       ZFI00102
LN 0103           NZ=NERRS                                                      ZFI00103
LN 0104           DO 1501 KK=IX,I                                               ZFI00104
LN 0105           IF(CARDP(KK).EQ.PUCO) GOTO 1502                               ZFI00105
LN 0106      1501 CONTINUE                                                      ZFI00106
LN 0107           KK=I                                                          ZFI00107
LN 0108           FNUM=2.1                                                      ZFI00108
LN 0109           GOTO 1503                                                     ZFI00109
LN 0110     C**** POINTER POSITION                                              ZFI00110
LN 0111      1502 CALL ZTRANX(KK+1,I-1)                                         ZFI00111
LN 0112           IF(NZ.NE.NERRS) GOTO 7004                                     ZFI00112
LN 0113           FNUM=3.1                                                      ZFI00113
LN 0114           IPROG(INREG)=-56                                              ZFI00114
LN 0115           INREG=INREG-1                                                 ZFI00115
LN 0116     C**** SENTENCE NUMBER                                               ZFI00116
LN 0117      1503 CALL ZTRANX(K+1,KK-1)                                         ZFI00117
LN 0118           IF(NZ.NE.NERRS) GOTO 7004                                     ZFI00118
LN 0119           IPROG(INREG)=-57                                              ZFI00119
LN 0120           INREG=INREG-1                                                 ZFI00120
LN 0121           GOTO 1007                                                     ZFI00121
LN 0122     C**** COMMAND IS COMMON-FILE                                        ZFI00122
```

```
LN 0123     6000 ITANF=IBEGST+11                                                ZFI00123
LN 0124          IF(MSTAT.NE.0) GOTO 7003                                       ZFI00124
LN 0125          GOTO 1100                                                      ZFI00125
LN 0126     C**** COMMAND IS CLOSE                                              ZFI00126
LN 0127      5000 MOP=-55                                                       ZFI00127
LN 0128          ITANF=IBEGST+5                                                 ZFI00128
LN 0129          GOTO 1200                                                      ZFI00129
LN 0130     C**** COMMAND IS PUT                                                ZFI00130
LN 0131      2000 MOP=-49                                                       ZFI00131
LN 0132      3001 MSTAT=MSTAT+1                                                 ZFI00132
LN 0133          IBEGST=IBEGST+3                                                ZFI00133
LN 0134          IF(CARDP(IBEGST).NE.QUOTE) GOTO 7001                           ZFI00134
LN 0135          IF(IBEGST.GT.LNGCRP-3) GOTO 7000                               ZFI00135
LN 0136          DO 2001 I=IBEGST,LNGCRP                                        ZFI00136
LN 0137          IF(CARDP(I).EQ.COMMA) GOTO 2003                                ZFI00137
LN 0138      2001 CONTINUE                                                      ZFI00138
LN 0139      7000 NERROR=42                                                     ZFI00139
LN 0140      2222 NN=0                                                          ZFI00140
LN 0141          CALL COMERR(NERROR,I1,I2,X1,X2,NN)                             ZFI00141
LN 0142          NERRS=NERRS+1                                                  ZFI00142
LN 0143          RETURN                                                         ZFI00143
LN 0144      2003 IF(CARDP(I-1).NE.QUOTE) GOTO 2004                             ZFI00144
LN 0145          K=I                                                            ZFI00145
LN 0146      2007 CALL STRING(IBEGST+1,K-2,IX)                                  ZFI00146
LN 0147          IF(MOP.LT.-57) GOTO 7010                                       ZFI00147
LN 0148          CALL ZLISTE(I+1,MOP,IX,MCOM)                                   ZFI00148
LN 0149          RETURN                                                         ZFI00149
LN 0150      2004 DO 2005 K=IBEGST,I                                            ZFI00150
LN 0151          IF(CARDP(K).EQ.PUCO) GOTO 2006                                 ZFI00151
LN 0152      2005 CONTINUE                                                      ZFI00152
LN 0153      2006 IF(CARDP(K-1).NE.QUOTE) GOTO 7001                             ZFI00153
LN 0154          CALL ZTRANX(K+1,T-1)                                           ZFI00154
LN 0155          IPROG(INREG)=-57                                               ZFI00155
LN 0156          INREG=INREG-1                                                  ZFI00156
LN 0157          MOP=MOP-1                                                      ZFI00157
LN 0158          GOTO 2007                                                      ZFI00158
LN 0159     C**** COMMAND IS GET                                                ZFI00159
LN 0160      3000 MOP=-51                                                       ZFI00160
LN 0161          GOTO 3001                                                      ZFI00161
LN 0162     C**** COMMAND IS RESET                                              ZFI00162
LN 0163      4000 ITANF=IBEGST+5                                                ZFI00163
LN 0164          MOP=-53                                                        ZFI00164
LN 0165          GOTO 1200                                                      ZFI00165
LN 0166     C**** COMMAND IS MAT GET                                            ZFI00166
LN 0167      7500 MOP=-58                                                       ZFI00167
LN 0168          GOTO 3001                                                      ZFI00168
LN 0169      7010 ITANF=I+1                                                     ZFI00169
LN 0170      7013 DO 7011 K=ITANF,LNGCRP                                        ZFI00170
LN 0171          IF(CARDP(K).EQ.COMMA) GOTO 7012                                ZFI00171
LN 0172      7011 CONTINUE                                                      ZFI00172
LN 0173          K=LNGCRP+1                                                     ZFI00173
LN 0174      7012 I=K-1                                                         ZFI00174
LN 0175          IF(CARDP(I).EQ.DOLSGN) I=I-1                                   ZFI00175
LN 0176          CALL ZALPH(CARDP(I),KK)                                        ZFI00176
LN 0177          IF(KK.GT.26) GOTO 7050                                         ZFI00177
LN 0178          CALL ZKLAM(I,I,KK)                                             ZFI00178
LN 0179          IPROG(INREG)=MOP                                               ZFI00179
LN 0180          IPROG(INREG-1)=IX                                              ZFI00180
LN 0181          IPROG(INREG-2)=KK                                              ZFI00181
LN 0182          INREG=INREG-3                                                  ZFI00182
LN 0183          IF(K.GE.LNGCRP) RETURN                                         ZFI00183
```

```
LN 0184           ITANF=K+1                                          ZFI00184
LN 0185           GOTO 7013                                          ZFI00185
LN 0186     C**** COMMAND IS MAT PUT                                 ZFI00186
LN 0187      8000 MOP=-60                                            ZFI00187
LN 0188           GOTO 3001                                          ZFI00188
LN 0189      7050 NERROR=43                                          ZFI00189
LN 0190           X1=CARDP(K)                                        ZFI00190
LN 0191           GOTO 2222                                          ZFI00191
LN 0192      7001 NERROR=44                                          ZFI00192
LN 0193           GOTO 2222                                          ZFI00193
LN 0194      7002 NERROR=45                                          ZFI00194
LN 0195           I2=FNUM                                            ZFI00195
LN 0196           GOTO 2222                                          ZFI00196
LN 0197      7003 NERROR=46                                          ZFI00197
LN 0198           GOTO 2222                                          ZFI00198
LN 0199      7004 NERROR=47                                          ZFI00199
LN 0200           GOTO 2222                                          ZFI00200
LN 0201      7005 NERROR=48                                          ZFI00201
LN 0202           GOTO 2222                                          ZFI00202
LN 0203      7006 NERROR=49                                          ZFI00203
LN 0204           GOTO 2222                                          ZFI00204
LN 0205      7007 NERROR=50                                          ZFI00205
LN 0206           GOTO 2222                                          ZFI00206
LN 0207           END                                                ZFI00207

      USASI FORTRAN DIAGNOSTIC RESULTS FOR ZFILE

           NO ERRORS
```

53 Programme des Ausführungssystems

```
LN 0001     C      SUBROUTINE ZEXEC                                                ZEX00001
LN 0002     C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   ZEX00002
LN 0003     C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*      ZEX00003
LN 0004     C---- GEARBEITET WIRD.IN DEM VORAUSGEHENDEN STATEMENT IST DAS C IN DER  ZEX00004
LN 0005     C---- 1.SPALTE ZU ENTFERNEN.                                           ZEX00005
LN 0006           PROGRAM EXECUT                                                  ZEX00006
LN 0007     C---- (SIEHE BEMERKUNGEN IM HAUPTELEMENT DER *OVERLAY-TECHNIK*)        ZEX00007
LN 0008           COMMON// ACC,ASTPSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    ZEX00008
LN 0009          1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZEX00009
LN 0010          2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL, ZEX00010
LN 0011          3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,ZEX00011
LN 0012          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI    ZEX00012
LN 0013           COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,   ZEX00013
LN 0014          1NIFMAX,INTZEI                                                   ZEX00014
LN 0015           COMMON// CARDT(80),MERKER(26,2),CARP(140),                      ZEX00015
LN 0016          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    ZEX00016
LN 0017          1IPET(20),XXX(4),NFILE(25,3)                                     ZEX00017
LN 0018           COMMON// ISTLST(340),LISTST(340)                                ZEX00018
LN 0019           COMMON// DATAN(330)                                             ***
LN 0020           COMMON// DATA(3700)                                             ZEX00020
LN 0021           DIMENSION IPROG(3700)                                           ZEX00021
LN 0022           EQUIVALENCE (DATA(1),IPROG(1))                                  ZEX00022
LN 0023           DIMENSION IPRT(10)                                              ZEX00023
LN 0024     C                                                                     ZEX00024
LN 0025     C---- DAS FOLGENDE STATEMENT IST ZU ENTFERNEN ODER DURCH EIN C IN DER   ZEX00025
LN 0026     C---- 1.SPALTE UNWIRKSAM ZU MACHEN,WENN NICHT IN *OVERLAY-TECHNIK*     ZEX00026
LN 0027     C---- GEARBEITET WIRD.                                                ZEX00027
LN 0028           OVERLAY(2)                                                      ZEX00028
LN 0029           INEXT=1                                                         ZEX00029
LN 0030           NPRUS=1                                                         ZEX00030
LN 0031           IF(NSTZEI.NE.2) GOTO 1                                          ZEX00031
LN 0032     C**** NUMBER OF FOSBIC-CODE ERRORS = 51                               ***
LN 0033           IA=51                                                           ***
LN 0034           I1=12                                                           ***
LN 0035           I2=25                                                           ***
LN 0036           X1=ALPH(1)                                                      ***
LN 0037           X2=12345                                                        ***
LN 0038           DO 2 I=1,IA                                                     ZEX00034
LN 0039           NERROR=I                                                        ZEX00035
LN 0040           CALL EXERR(NERROR,I1,I2,X1,X2)                                  ZEX00036
LN 0041         2 CONTINUE                                                        ZEX00037
LN 0042     C**** MAXIMAL INTERNAL FOSBIC CODE NUMBER UNTIL APRIL 1976            ZEX00038
LN 0043         1 ICODE=71                                                        ZEX00039
LN 0044     C**** INITIALIZE RANDOM NUMBER GENERATOR                              ZEX00040
LN 0045           XRN=RNG(2.)                                                     ZEX00041
LN 0046           NOLNP=NCELLP-INREG+1                                            ZEX00042
LN 0047     C**** SET INTERNAL ADDRESS IN COMPUTED GOTO AND GOSUR ****            ZEX00043
LN 0048           DO 200 I=1,NOLNP                                                ZEX00044
LN 0049           INREG=NCELLP-I                                                  ZEX00045
LN 0050           IF(IPROG(INREG).NE.-27) GOTO 200                                ZEX00046
LN 0051           IF(NSTLST.EQ.0) GOTO 204                                        ZEX00047
LN 0052           MIC=IPROG(INREG-1)                                              ZEX00048
LN 0053           DO 201 J=1,MIC                                                  ZEX00049
LN 0054           DO 202 K=1,NSTLST                                               ZEX00050
LN 0055           IA=INREG-J-1                                                    ZEX00051
LN 0056           IF(IPROG(IA).EQ.LISTST(K)) GOTO 203                             ZEX00052
LN 0057       202 CONTINUE                                                        ZEX00053
LN 0058       204 NERROR=2                                                        ZEX00054
LN 0059           CALL EXERR(NERROR,IPROG(IA),I2,X1,X2)                           ZEX00055
LN 0060           NERRS=NERRS+1                                                   ZEX00056
LN 0061           GOTO 201                                                        ZEX00057
```

```
LN 0062     203  IPROG(IA)=ISTLST(K)                                                    ZEX00058
LN 0063     201  CONTINUE                                                               ZEX00059
LN 0064     200  CONTINUE                                                               ZEX00060
LN 0065          IF(NERRS.NE.0) GOTO 21000                                             ZEX00061
LN 0066          INREG=NCELLP                                                          ZEX00062
LN 0067          MOP=0                                                                 ZEX00063
LN 0068          NIRET=0                                                               ZEX00064
LN 0069          NPRI=1                                                                ZEX00065
LN 0070          CALL CLEAR(1,140)                                                     ZEX00066
LN 0071          DO 110 I=1,4                                                          ZEX00067
LN 0072     110  XXX(I)=0.                                                            ZEX00068
LN 0073          NIT=0                                                                 ZEX00069
LN 0074          WRITE(IWC,603)                                                        ZEX00070
LN 0075     603  FORMAT(1H1)                                                          ZEX00071
LN 0076    C**** RESET COMMON-FILES                                                    ZEX00072
LN 0077          NPR=0                                                                 ZEX00073
LN 0078    C----AUF DIE FOLGENDE ZEILE LN=0079 WIRD IM TEXT BEZUG GENOMMEN++          ***
LN 0079          CALL ZEXFIL(NPR)                                                      ZEX00074
LN 0080    C                                                                           ZEX00075
LN 0081     900  IOP=IPROG(INREG)                                                      ZEX00076
LN 0082          NPR=0                                                                 ZEX00077
LN 0083          NSTOP=0                                                               ZEX00078
LN 0084          MIC=0                                                                 ZEX00079
LN 0085          IF(IOP.LT.0) GOTO 950                                                 ZEX00080
LN 0086    C*****ILLEGAL OPERAND                                                       ZEX00081
LN 0087     909  NERROR=9                                                            ZEX00082
LN 0088          IF(IOP.EQ.27) GOTO  29008                                             ***
LN 0089          IF(IOP.EQ.30) GOTO  29008                                             ***
LN 0090          IF(IOP.EQ.44) GOTO  29008                                             ***
LN 0091          IF(IOP.EQ.45) GOTO  29008                                             ***
LN 0092          IF(IOP.EQ.71) GOTO  906                                               ***
LN 0093          IF(IOP.GE.48) GOTO 907                                                ***
LN 0094     906  NERROR=11                                                           ***
LN 0095          GOTO 29008                                                            ***
LN 0096     907  IF(IOP.GT.61) GOTO 908                                              ***
LN 0097          NERROR=12                                                             ***
LN 0098          GOTO 29008                                                            ***
LN 0099     908  NERROR=9                                                            ***
LN 0100          GOTO 29008                                                          ZEX00083
LN 0101    C                                                                           ZEX00084
LN 0102     950  IOP=-IOP                                                           ZEX00085
LN 0103          IF((IOP.LT.1).OR.(IOP.GT.ICODE)) GOTO 19001                        ZEX00086
LN 0104          IF(IOP.GE.23) GOTO 960                                             ZEX00087
LN 0105          GOTO(1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,           ZEX00088
LN 0106         1 11000,12000,13000,14000,15000,16000,17000,18000,19000,20000,21000ZEX00089
LN 0107         2,22000),IOP                                                        ZEX00090
LN 0108     960  IF(IOP.EQ.27) GOTO 27000                                           ZEX00091
LN 0109          IF(IOP.EQ.30) GOTO 30000                                           ZEX00092
LN 0110          IF(IOP.EQ.44) GOTO 44000                                           ZEX00093
LN 0111          IF(IOP.EQ.45) GOTO 45000                                           ZEX00094
LN 0112          IF(IOP.EQ.71) GOTO 44400                                           ZEX00095
LN 0113          IF(IOP.GE.48) GOTO 970                                             ZEX00096
LN 0114    44400 IOP=IOP-22                                                         ZEX00097
LN 0115    C**** MAT CODE EXECUTION IN ZEXMAT                                       ZEX00098
LN 0116    C----AUF DIE FOLGENDE ZEILE LN=0117 WIRD IM TEXT BEZUG GENOMMEN++          ***
LN 0117          CALL ZEXMAT(IOP)                                                   ZEX00099
LN 0118          IF(IOP.GT.0) GOTO 900                                              ZEX00100
LN 0119          GOTO 21000                                                         ZEX00101
LN 0120    C**** FILE CODE EXECUTION IN ZEXFIL                                      ZEX00102
LN 0121     970  IF(IOP.GT.61) GOTO 990                                            ZEX00103
LN 0122          IOP=IOP-47                                                         ZEX00104
```

```
LN 0123    C----AUF DIE FOLGENDE ZEILE LN=0124 WIRD IM TEXT BEZUG GENOMMEN**      ***
LN 0124          CALL ZEXFIL(ICP)                                                ZEX00105
LN 0125          IF(IOP.GT.9) GOTO 900                                           ZEX00106
LN 0126          GOTO 21000                                                      ZEX00107
LN 0127     900  IOP=IOP-61                                                      ZEX00108
LN 0128          GOTO(62000,63000,64000,65000,66000,67000,68000,69000,70000),IOP ZEX00109
LN 0129    C*****EVALUATE EXPRESSION                                             ZEX00110
LN 0130     1000 CALL ZEVAL(NSTOP)                                               ZEX00111
LN 0131          IF(NSTOP.EQ.1) GOTO 21000                                       ZEX00112
LN 0132          GOTO 900                                                        ZEX00113
LN 0133    C                                                                     ZEX00114
LN 0134    C*****STORE FOR RETURN                                                ZEX00115
LN 0135     2000 NIRET=NIRET+1                                                   ZEX00116
LN 0136          IF(NIRET.LE.NIRMAX) GOTO 2100                                   ZEX00117
LN 0137          NERROR=1                                                        ZEX00118
LN 0138          GOTO 29008                                                      ZEX00119
LN 0139     2100 IRET(NIRET)=INREG-3                                             ZEX00120
LN 0140          INREG=INREG-1                                                   ZEX00121
LN 0141          GOTO 900                                                        ZEX00122
LN 0142    C                                                                     ZEX00123
LN 0143    C*****GO TO EXTERNAL                                                  ZEX00124
LN 0144     3000 IF(NSTLST.EQ.0) GOTO 3005                                       ZEX00125
LN 0145          DO 3004 I=1,NSTLST                                              ZEX00126
LN 0146          IF(IPROG(INREG-1).EQ.LISTST(I)) GOTO 3010                       ZEX00127
LN 0147     3004 CONTINUE                                                        ZEX00128
LN 0148     3005 NERROR=2                                                        ZEX00129
LN 0149          I1=IPROG(INREG-1)                                               ZEX00130
LN 0150          GOTO 29008                                                      ZEX00131
LN 0151    C                                                                     ZEX00132
LN 0152     3010 IPROG(INREG-1)=ISTLST(I)                                        ZEX00133
LN 0153          IPROG(INREG)=-4                                                 ZEX00134
LN 0154          INREG=IPROG(INREG-1)                                            ZEX00135
LN 0155          GOTO 900                                                        ZEX00136
LN 0156    C                                                                     ZEX00137
LN 0157    C*****GO TO INTERNAL                                                  ZEX00138
LN 0158     4000 INREG=IPROG(INREG-1)                                            ZEX00139
LN 0159          GOTO 900                                                        ZEX00140
LN 0160    C                                                                     ZEX00141
LN 0161    C*****RETURN                                                          ZEX00142
LN 0162     5000 IF(NIRET.GT.0) GOTO 5010                                        ZEX00143
LN 0163          NERROR=3                                                        ZEX00144
LN 0164          GOTO 29008                                                      ZEX00145
LN 0165    C                                                                     ZEX00146
LN 0166     5010 INREG=IRET(NIRET)                                               ZEX00147
LN 0167          NIRET=NIRET-1                                                   ZEX00148
LN 0168          GOTO 900                                                        ZEX00149
LN 0169    C                                                                     ZEX00150
LN 0170    C*****CONDITIONAL TRANSFER                                            ZEX00151
LN 0171     6000 ICOMP=IPROG(INREG-1)                                            ZEX00152
LN 0172          ITF=0                                                           ZEX00153
LN 0173          GOTO(6010,6020,6030,6040,6050,6060),ICOMP                       ZEX00154
LN 0174     6010 IF(XXX(1).GT.XXX(2)) ITF=1                                      ZEX00155
LN 0175          GOTO 6100                                                       ZEX00156
LN 0176     6020 IF(XXX(1).GE.XXX(2)) ITF=1                                      ZEX00157
LN 0177          GOTO 6100                                                       ZEX00158
LN 0178     6030 IF(XXX(1).LT.XXX(2)) ITF=1                                      ZEX00159
LN 0179          GOTO 6100                                                       ZEX00160
LN 0180     6040 IF(XXX(1).LE.XXX(2)) ITF=1                                      ZEX00161
LN 0181          GOTO 6100                                                       ZEX00162
LN 0182     6050 IF(XXX(1).EQ.XXX(2)) ITF=1                                      ZEX00163
LN 0183          GOTO 6100                                                       ZEX00164
```

```
LN 0134     6060 IF(XXX(1).NE.YYY(2)) ITF=1                                  ZEX00165
LN 0135     6100 IF(ITF.EQ.0) GOTO 6200                                      ZEX00166
LN 0136          INREG=INREG-2                                               ZEX00167
LN 0137          GOTO 900                                                    ZEX00168
LN 0138     6200 INREG=INREG-4                                               ZEX00169
LN 0139          GOTO 900                                                    ZEX00170
LN 0140     C                                                                ZEX00171
LN 0141     C*****READ INTO ACCUMULATOR                                      ZEX00172
LN 0142     7000 NSW=1                                                       ZEX00173
LN 0143     7001 IFR=1                                                       ZEX00174
LN 0144          CALL ZHOPPR(VALUE,NSTOP,IFR,NSW)                            ZEX00175
LN 0145          IF(NSTOP.EQ.0) GOTO 7010                                    ZEX00176
LN 0146    22400 NERROR=NSTOP+3                                              ZEX00177
LN 0147          I1=VALUE                                                    ZEX00178
LN 0148    29008 CALL EXERR(NERROR,I1,I2,X1,X2)                              ZEX00179
LN 0149          GOTO 21000                                                  ZEX00180
LN 0150     7010 ACC=VALUE                                                   ZEX00181
LN 0151          INREG=INREG-1                                               ZEX00182
LN 0152          GOTO 900                                                    ZEX00183
LN 0153     C                                                                ZEX00184
LN 0154     C*****STORE FROM ACCUMULATOR                                     ZEX00185
LN 0155     8000 IPOS=IPROG(INREG-1)                                         ZEX00186
LN 0156          DATA(IPOS)=ACC                                              ZEX00187
LN 0157          INREG=INREG-2                                               ZEX00188
LN 0158          GOTO 900                                                    ZEX00189
LN 0159     C                                                                ZEX00190
LN 0160     C*****STORE FROM ACCUMULATOR INDIRECT                            ZEX00191
LN 0161     9000 IPOS=IPROG(INREG-1)                                         ZEX00192
LN 0162          LOC=DATA(IPOS)+1.5                                          ZEX00193
LN 0163          DATA(LOC)=ACC                                               ZEX00194
LN 0164          INREG=INREG-2                                               ZEX00195
LN 0165          GOTO 900                                                    ZEX00196
LN 0166     C                                                                ZEX00197
LN 0167     C*****SKIP A LINE                                                ZEX00198
LN 0168    10000 INREG=INREG-1                                               ZEX00199
LN 0169          IF((NPRUS.EQ.1).AND.(INEXT.EQ.1)) GOTO 10200                ZEX00200
LN 0170          NX=3                                                        ***
LN 0171          CALL PRILIN(NX)                                             ZEX00202
LN 0172          GOTO 10300                                                  ZEX00203
LN 0173    10200 WRITE(IWC,10100)                                            ZEX00204
LN 0174    10100 FORMAT(1H )                                                 ZEX00205
LN 0175    10300 NPRI=1                                                      ZEX00206
LN 0176          GOTO 900                                                    ZEX00207
LN 0177     C                                                                ZEX00208
LN 0178     C*****STOP                                                       ZEX00209
LN 0179    11000 NX=3                                                        ZEX00210
LN 0180          CALL PRILIN(NX)                                             ZEX00211
LN 0181          GOTO 21000                                                  ZEX00212
LN 0182     C                                                                ZEX00213
LN 0183     C*****NO OPERATION                                               ZEX00214
LN 0184    12000 INREG=INREG-1                                               ZEX00215
LN 0185          GOTO 900                                                    ZEX00216
LN 0186     C                                                                ZEX00217
LN 0187     C*****PAGE                                                       ZEX00218
LN 0188    13000 NX=3                                                        ZEX00219
LN 0189          CALL PRILIN(NX)                                             ZEX00220
LN 0190          WRITE(IWC,13005)                                            ZEX00221
LN 0191    13005 FORMAT(1H1)                                                 ZEX00222
LN 0192          INREG=INREG-1                                               ZEX00223
LN 0193          GOTO 900                                                    ZEX00224
LN 0194     C                                                                ZEX00225
```

```
LN 0245     C*****LOAD ACC INTO EXP                                          ZEX00226
LN 0246     14000 LOC=IPROG(INREG-1)                                         ZEX00227
LN 0247           XXX(LOC)=ACC                                               ZEX00228
LN 0248           INREG=INREG-2                                              ZEX00229
LN 0249           GOTO 900                                                   ZEX00230
LN 0250     C                                                                ZEX00231
LN 0251     C*****FOR STEP AND TEST                                          ZEX00232
LN 0252     C*****FIND LOCATION OF VARIABLE                                  ZEX00233
LN 0253     15000 LOCV=IPROG(INREG-1)                                        ZEX00234
LN 0254           IF(LOCV.GT.26) GOTO 15100                                  ZEX00235
LN 0255     C*****ONE-CHARACTER NAME                                         ZEX00236
LN 0256           LOCV=DATA(LOCV)+1.5                                        ZEX00237
LN 0257     15100 LOCE1=IPROG(INREG-2)                                       ZEX00238
LN 0258     C*****INCREMENT                                                  ZEX00239
LN 0259           DATA(LOCV)=DATA(LOCV)+DATA(LOCE1+2)                        ZEX00240
LN 0260     C*****CHECK STEP                                                 ZEX00241
LN 0261           IF(DATA(LOCE1+2).GT.0.) GOTO 15200                         ZEX00242
LN 0262           IF(DATA(LOCE1+2).LT.0.) GOTO 15300                         ZEX00243
LN 0263     C*****ZERO STEP                                                  ZEX00244
LN 0264           NERROR=10                                                  ZEX00245
LN 0265           GOTO 29008                                                 ZEX00246
LN 0266     C*****TEST FOR TRANSFER OUT OF LOOP -- E3 GT 0                   ZEX00247
LN 0267     15200 IF(DATA(LOCV).GT.DATA(LOCE1+1)) GOTO 15500                 ZEX00248
LN 0268     C*****CONTINUE LOOP                                              ZEX00249
LN 0269           INREG=INREG-4                                              ZEX00250
LN 0270           GOTO 900                                                   ZEX00251
LN 0271     C*****TEST FOR TRANSFER OUT OF LOOP -- E3 LT 0                   ZEX00252
LN 0272     15300 IF(DATA(LOCV).LT.DATA(LOCE1+1)) GOTO 15500                 ZEX00253
LN 0273     C*****CONTINUE LOOP                                              ZEX00254
LN 0274           INREG=INREG-4                                              ZEX00255
LN 0275           GOTO 900                                                   ZEX00256
LN 0276     C*****TRANSFER OUT OF LOOP                                       ZEX00257
LN 0277     15500 INREG=IPROG(INREG-3)                                       ZEX00258
LN 0278           GOTO 900                                                   ZEX00259
LN 0279     C                                                                ZEX00260
LN 0280     C*****INSERT ALPHA                                               ZEX00261
LN 0281     16000 IF(NPRI.EQ.1) GOTO 16012                                   ZEX00262
LN 0282           IF(NPRUS.LE.INEXT) GOTO 16013                              ZEX00263
LN 0283           INEXT=NPRUS                                                ZEX00264
LN 0284           GOTO 16013                                                 ZEX00265
LN 0285     16012 I1=INEXT                                                   ZEX00266
LN 0286           I2=NPRUS                                                   ZEX00267
LN 0287           CALL CHECK(I1,I2)                                          ZEX00268
LN 0288     16013 IF(INEXT.LT.IWRIT) GOTO 16014                              ZEX00269
LN 0289           NX=1                                                       ZEX00270
LN 0290           CALL PRILIN(NX)                                            ZEX00271
LN 0291     16014 IF(NPR.EQ.1) GOTO 30100                                    ZEX00272
LN 0292           MOP=IPROG(INREG-1)                                         ZEX00273
LN 0293           INPEG=INREG-1                                              ZEX00274
LN 0294     30100 IPOS=INEXT                                                 ZEX00275
LN 0295           DO 16500 I=1,MOP                                           ZEX00276
LN 0296           IF(NPR.EQ.1) GOTO 16002                                    ZEX00277
LN 0297           LOC=INREG-I                                                ZEX00278
LN 0298           IX=IPROG(LOC)                                              ZEX00279
LN 0299     16002 DO 16400 K=1,5                                             ZEX00280
LN 0300           IZ=IX-((IX/INTZET)*INTZET)                                 ***
LN 0301           CARP(IPOS)=ALPH(IZ+1)                                      ZEX00282
LN 0302           IPOS=IPOS+1                                                ZEX00283
LN 0303     16400 IX=IX/INTZET                                               ZEX00284
LN 0304     16500 CONTINUE                                                   ZEX00285
LN 0305           INREG=INREG-MOP-1                                          ZEX00286
```

```
LN 0306          IF(NPRI.EQ.1) INEXT=INEXT+IZONE                                ZEX00287
LN 0307          IF(NPRI.EQ.-1) INEXT=INEXT+5*MOP                               ZEX00288
LN 0308          IF(INEXT.GE.IWRIT) GOTO 18100                                  ZEX00289
LN 0309          GOTO 900                                                       ZEX00290
LN 0310     C                                                                   ZEX00291
LN 0311     C*****INSERT ACC                                                    ZEX00292
LN 0312     17000 INREG=INREG-1                                                 ZEX00293
LN 0313          IF(NPRI.EQ.1) GOTO 17010                                       ZEX00294
LN 0314          IF(NPRUS.LE.INEXT) GOTO 17020                                  ZEX00295
LN 0315          INEXT=NPRUS                                                    ZEX00296
LN 0316          GOTO 17020                                                     ZEX00297
LN 0317     17010 I1=INEXT                                                      ZEX00298
LN 0318          I2=NPRUS                                                       ZEX00299
LN 0319          CALL CHECK(I1,I2)                                              ZEX00300
LN 0320     17020 IF(INEXT.LT.IWRIT) GOTO 17205                                 ZEX00301
LN 0321          NX=1                                                           ZEX00302
LN 0322          CALL PRILIN(NX)                                                ZEX00303
LN 0323     17205 INEXT=INEXT+IZONE                                             ZEX00304
LN 0324          CALL ZINSNO                                                    ZEX00305
LN 0325          IF(INEXT.GE.IWRIT) GOTO 18100                                  ZEX00306
LN 0326          GOTO 900                                                       ZEX00307
LN 0327     C                                                                   ZEX00308
LN 0328     C*****PRINT AND RESTORE                                             ZEX00309
LN 0329     18000 INREG=INREG-1                                                 ZEX00310
LN 0330          NPRI=1                                                         ZEX00311
LN 0331     18100 NX=3                                                          ZEX00312
LN 0332          CALL PRILIN(NX)                                                ZEX00313
LN 0333          GOTO 900                                                       ZEX00314
LN 0334     C                                                                   ZEX00315
LN 0335     C*****DUMP PROGRAM AND DATA                                         ZEX00316
LN 0336     19000 INREG=INREG-1                                                 ZEX00317
LN 0337     19001 WRITE(IWC,19100)                                              ZEX00318
LN 0338     19100 FORMAT(15H1 PROGRAM DUMP                                      ZEX00319
LN 0339          IF(NSTLST.EQ.0) GOTO 19600                                     ZEX00320
LN 0340          DO 19500 I=1,NSTLST                                            ZEX00321
LN 0341          WRITE(IWC,19200) LISTST(I),ISTLST(I)                           ZEX00322
LN 0342     19200 FORMAT(18H0 STATEMENT NUMBER,I8,21H  INTERNAL LOCATION =,I8)  ZEX00323
LN 0343          IF(ISTLST(I).GT.NCELLP) GOTO 19500                             ZEX00324
LN 0344          ITOP=ISTLST(I)                                                 ZEX00325
LN 0345          IF(I.EQ.NSTLST) GOTO 19300                                     ZEX00326
LN 0346          NX=I                                                           ZEX00327
LN 0347     19355 NX=NX+1                                                       ZEX00328
LN 0348          IBOT=ISTLST(NX)+1                                              ZEX00329
LN 0349          IF(ISTLST(NX).GT.NCELLP) GOTO 19355                            ZEX00330
LN 0350          GOTO 19350                                                     ZEX00331
LN 0351     19300 IBOT=NCELLP+1-NOLNP                                           ZEX00332
LN 0352     19350 IF(ITOP.GT.(IBOT+9)) GOTO 19400                               ZEX00333
LN 0353          ITEMS=ITOP-IBOT+1                                              ZEX00334
LN 0354          DO 19360 K=1,ITEMS                                             ZEX00335
LN 0355          IXYZ=ITOP-K+1                                                  ZEX00336
LN 0356     19360 IPRT(K)=IPROG(IXYZ)                                           ZEX00337
LN 0357          WRITE(IWC,19370) (IPRT(K),K=1,ITEMS)                           ZEX00338
LN 0358     19370 FORMAT(1H ,5X,10I11)                                          ZEX00339
LN 0359          GOTO 19500                                                     ZEX00340
LN 0360     19400 DO 19450 K=1,10                                               ZEX00341
LN 0361          IXYZ=ITOP-K+1                                                  ZEX00342
LN 0362     19450 IPRT(K)=IPROG(IXYZ)                                           ZEX00343
LN 0363          WRITE(IWC,19370) (IPRT(K),K=1,10)                              ZEX00344
LN 0364          ITOP=ITOP-10                                                   ZEX00345
LN 0365          GOTO 19350                                                     ZEX00346
LN 0366     19500 CONTINUE                                                      ZEX00347
```

```
LN 0367    19600 ITOP=DATA(1)+.1-1.                                                ZEX00348
LN 0368          WRITE(IWC,19650)                                                  ZEX00349
LN 0369    19650 FORMAT(12H0 CONSTANTS    )                                       ZEX00350
LN 0370          WRITE(IWC,19655)                                                  ZEX00351
LN 0371    19655 FORMAT(24H0LOCATION         VALUE          )                      ZEX00352
LN 0372          DO 19700 I=314,ITOP                                               ZEX00353
LN 0373    19700 WRITE(IWC,19750) I,DATA(I)                                        ZEX00354
LN 0374    19750 FORMAT(1H ,I8,F20.5)                                              ZEX00355
LN 0375          WRITE(IWC,603)                                                    ZEX00356
LN 0376          IF((IOP.LT.1).OR.(IOP.GT.IGODE)) GOTO 20001                       ZEX00357
LN 0377          GOTO 900                                                          ZEX00358
LN 0378    C                                                                       ZEX00359
LN 0379    C*****PRINT ALL                                                         ZEX00360
LN 0380    C*****DUMP OUTPUT BUFFER                                                ZEX00361
LN 0381    20000 INREG=INREG-1                                                     ZEX00362
LN 0382    20001 NX=3                                                              ZEX00363
LN 0383          CALL PRILIN(NX)                                                   ZEX00364
LN 0384          WRITE(IWC,10100)                                                  ZEX00365
LN 0385    C*****UNSUBSCRIPTED SINGLE-LETTER VARIABLES                             ZEX00366
LN 0386          DO 20100 I=1,26                                                   ZEX00367
LN 0387          LOCN=DATA(I)+1.1                                                  ZEX00368
LN 0388          VAL=DATA(LOCN)                                                    ZEX00369
LN 0389          IF((VAL.LT.SMALL).AND.(VAL.GT.-SMALL)) GOTO 20100                 ZEX00370
LN 0390          INEXT=16                                                          ZEX00371
LN 0391          ACC=VAL                                                           ZEX00372
LN 0392          CALL CLEAR(1,45)                                                  ZEX00373
LN 0393          CALL ZINSNO                                                       ZEX00374
LN 0394          WRITE(IWC,20050) ALPH(I),(CARP(KZ),KZ=1,15)                       ZEX00375
LN 0395    20050 FORMAT(1H , 41X, A1, 4H  =  , 15A1)                               ZEX00376
LN 0396    20100 CONTINUE                                                          ZEX00377
LN 0397    C                                                                       ZEX00378
LN 0398    C*****UNSUBSCRIPTED DOUBLE-LETTER VARIABLES                             ZEX00379
LN 0399          DO 20300 I=1,26                                                   ZEX00380
LN 0400          DO 20200 J=1,10                                                   ZEX00381
LN 0401          JM=J-1                                                            ZEX00382
LN 0402          LOCN=I+(26*JM)+53                                                 ZEX00383
LN 0403          VAL=DATA(LOCN)                                                    ZEX00384
LN 0404          IF((VAL.LT.SMALL).AND.(VAL.GT.-SMALL)) GOTO 20200                 ZEX00385
LN 0405          INEXT=16                                                          ZEX00386
LN 0406          ACC=VAL                                                           ZEX00387
LN 0407          CALL CLEAR(1,45)                                                  ZEX00388
LN 0408          CALL ZINSNO                                                       ZEX00389
LN 0409          WRITE(IWC,20150) ALPH(I),JM,(CARP(KZ),KZ=1,15)                    ZEX00390
LN 0410    20150 FORMAT(1H ,41X,A1,I1,3H = ,15A1)                                  ZEX00391
LN 0411    20200 CONTINUE                                                          ZEX00392
LN 0412    20300 CONTINUE                                                          ZEX00393
LN 0413    C                                                                       ZEX00394
LN 0414    C*****SUBSCRIPTED VARIABLES                                             ZEX00395
LN 0415          DO 20500 I=1,26                                                   ZEX00396
LN 0416          LOCS=DATA(I)+.1                                                   ZEX00397
LN 0417          LOCU=DATA(LOCS)+.1                                                ZEX00398
LN 0418          LOCUM=0                                                           ZEX00399
LN 0419          IF(DATA(I+27).NE.0.) LOCUM=(LOCU-LOCS-1)/INT(DATA(I+27))          ZEX00400
LN 0420          WRITE(IWC,350) LOCS,LOCU,LOCUM,DATA(I+27),ALPH(I)                 ZEX00401
LN 0421     350  FORMAT(6H LOCS=,I6,7H  LOCU=,I6,7H  ROWS=,I6,10H  COLUMNS=,F6.0,11ZEX00402
LN 0422         1H  VARIABLE ,A1)                                                  ZEX00403
LN 0423          IF(LOCU.EQ.(LOCS+2)) GOTO 20500                                   ZEX00404
LN 0424          NUMCOL=DATA(I+27)+.1                                              ZEX00405
LN 0425          LOCUM=LOCU-1                                                      ZEX00406
LN 0426          LOCSP=LOCS+1                                                      ZEX00407
LN 0427          DO 20400 K=LOCSP,LOCUM                                            ZEX00408
```

```
LN 0428          VAL=DATA(K)                                                    ZEX00409
LN 0429          IF((VAL.LT.SMALL).AND.(VAL.GT.-SMALL)) GOTO 20400              ZEX00410
LN 0430          ITEM=K-LOCSP                                                   ZEX00411
LN 0431          NR=ITEM/NUMCOL                                                 ZEX00412
LN 0432          NC=ITEM-(NR*NUMCOL)                                            ZEX00413
LN 0433          INEXT=16                                                       ZEX00414
LN 0434          ACC=VAL                                                        ZEX00415
LN 0435          CALL CLEAR(1,45)                                               ZEX00416
LN 0436          CALL ZINSNO                                                    ZEX00417
LN 0437          WRITE(IWC,20350) ALPH(I),ITEM,ALPH(I),NR,NC,(CARP(KZ),KZ=1,15),K  ZEX00418
LN 0438    20350 FORMAT(1H ,20X,A1,1H(,I4,5H) OP ,A1,1H(,I4,1H,,I4,3H) =,15A1,5X,5HZEX00419
LN 0439         1DATA(,I5,1H))                                                 ZEX00420
LN 0440    20400 CONTINUE                                                       ZEX00421
LN 0441    20500 CONTINUE                                                       ZEX00422
LN 0442          WRITE(IWC,10100)                                               ZEX00423
LN 0443          INEXT=1                                                        ZEX00424
LN 0444          WRITE(IWC,20701)                                               ZEX00425
LN 0445    20701 FORMAT(20X,31HALLOCATED FILES UNTIL PRINT ALL//20X,9HFILE-NAME,5X,ZEX00426
LN 0446         120HINTERNAL FILE NUMBER,5X,5HBEGIN,5X,3HEND,10X,9HSENTENCES/)    ZEX00427
LN 0447          DO 20702 I=1,MAXFIL                                            ZEX00428
LN 0448          IF(NFILE(I,1).EQ.-1) GOTO 20702                                ZEX00429
LN 0449          IPOS=0                                                         ZEX00430
LN 0450          IX=NFILE(I,1)                                                  ZEX00431
LN 0451          DO 20703 K=1,5                                                 ZEX00432
LN 0452          IZ=IX-((IX/INTZEI)*INTZEI)                                     ZEX00433
LN 0453          IPOS=IPOS+1                                                    ZEX00434
LN 0454          CARP(IPOS)=ALPH(IZ+1)                                          ZEX00435
LN 0455          IX=IX/INTZEI                                                   ZEX00436
LN 0456    20703 CONTINUE                                                       ZEX00437
LN 0457          LOCS=NFILE(I,2)+NFILE(I,3)-1                                   ZEX00438
LN 0458          WRITE(IWC,20704) I,(CARP(K),K=1,5),NFILE(I,1),NFILE(I,2),LOCS,NFILZEX00439
LN 0459         1E(I,3)                                                         ZEX00440
LN 0460    20704 FORMAT(15X,I2,5X,5A1,8X,I12,11X,I5,5X,I5,5X,I5)               ZEX00441
LN 0461    20702 CONTINUE                                                       ZEX00442
LN 0462          CALL CLEAR(1,140)                                              ZEX00443
LN 0463          IF((IOP.LT.1).OR.(IOP.GT.ICODE)) GOTO 909                      ZEX00444
LN 0464          GOTO 900                                                       ZEX00445
LN 0465    C                                                                    ZEX00446
LN 0466    C*****EXECUTION OF RESTORE COMMAND                                   ZEX00447
LN 0467    22000 IF(IPROG(INREG-1).EQ.-22) GOTO 22100                           ZEX00448
LN 0468          NEWPOS=ACC                                                     ZEX00449
LN 0469          INREG=INREG-1                                                  ZEX00450
LN 0470          GOTO 22300                                                     ZEX00451
LN 0471    22100 NEWPOS=IPROG(INREG-2)                                          ZEX00452
LN 0472          INREG=INREG-3                                                  ZEX00453
LN 0473    22300 NSW=2                                                          ZEX00454
LN 0474          CALL ZHOPPR(VALUE,NSTOP,NEWPOS,NSW)                            ZEX00455
LN 0475          IF(NSTOP.NE.0) GOTO 22400                                      ZEX00456
LN 0476          GOTO 900                                                       ZEX00457
LN 0477    C**** EXECUTION --COMPUTED GOTO                                      ZEX00458
LN 0478    27000 MIC=IPROG(INREG-1)                                             ZEX00459
LN 0479          MIV=ACC                                                        ZEX00460
LN 0480          IF((MIV.LT.1).OR.(MIV.GT.MIC)) GOTO 44001                      ZEX00461
LN 0481          INREG=INREG-MIV                                                ZEX00462
LN 0482          GOTO 4000                                                      ZEX00463
LN 0483    C                                                                    ZEX00464
LN 0484    C*****LOAD ALPHANUMERIC CODE INTO PRINT POSITION BY NPR              ZEX00465
LN 0485    30000 IX=ACC                                                         ZEX00466
LN 0486          NPR=1                                                          ZEX00467
LN 0487          MOP=1                                                          ZEX00468
LN 0488          INREG=INREG+1                                                  ZEX00469
```

```
LN 0489          GOTO 16000                                                  ZEX00470
LN 0490     C                                                                ZEX00471
LN 0491     C**** EXECUTION OF COMPUTED GOSUB                                ZEX00472
LN 0492     44000 MIC=IPROG(INREG-2)                                         ZEX00473
LN 0493          MIV=ACC                                                     ZEX00474
LN 0494          IF((MIV.LT.1).OR.(MIV.GT.MIC)) GOTO 44001                   ZEX00475
LN 0495          NIRET=NIRET+1                                               ZEX00476
LN 0496          IF(NIRET.GT.20) GOTO 2000                                   ZEX00477
LN 0497          IRET(NIRET)=INREG-3-MIC                                     ZEX00478
LN 0498          INREG=INREG-MIV-1                                           ZEX00479
LN 0499          GOTO 4000                                                   ZEX00480
LN 0500     44001 NERROR=41                                                  ZEX00481
LN 0501          I1=MIV                                                      ZEX00482
LN 0502          GOTO 29008                                                  ZEX00483
LN 0503     C**** EXECUTION OF INPUT                                         ZEX00484
LN 0504     45000 NSW=3                                                      ZEX00485
LN 0505          GOTO 7001                                                   ZEX00486
LN 0506     C**** GET IMAGE STATEMENT NUMBER                                 ZEX00487
LN 0507     62000 MOP=ACC                                                    ZEX00488
LN 0508          IF(INEXT.LE.NPRUS) GOTO 62030                               ZEX00489
LN 0509          NPRUS=INEXT                                                 ZEX00490
LN 0510     62030 IF(MOP.EQ.MAXIMA) GOTO 62035                               ZEX00491
LN 0511          REWIND NIMAGE                                               ZEX00492
LN 0512          DO 62010 I=1,NSTLST                                         ZEX00493
LN 0513          IF(MOP.EQ.LISTST(I)) GOTO 62020                             ZEX00494
LN 0514     62010 CONTINUE                                                   ZEX00495
LN 0515     62050 I1=MOP                                                     ZEX00496
LN 0516          NERROR=13                                                   ZEX00497
LN 0517          GOTO 29008                                                  ZEX00498
LN 0518     62020 MIV=ISTLST(I)-NCELLP                                       ZEX00499
LN 0519          DO 62025 I=1,MIV                                            ZEX00500
LN 0520          READ(NIMAGE,62040) MAXIMA,CARD                              ZEX00501
LN 0521     C-*-*                                                            ZEX00502
LN 0522     C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(NIMAGE)-  ZEX00503
LN 0523     C*-*-     ----CHECK IT----        ----CHECK IT----               ZEX00504
LN 0524     C-*-*                                                            ZEX00505
LN 0525          IF(IFEOF(NIMAGE).EQ.-1) GOTO 62050                          ZEX00506
LN 0526     62040 FORMAT(I4,80A1)                                            ZEX00507
LN 0527     62025 CONTINUE                                                   ZEX00508
LN 0528          IF(MAXIMA.NE.MOP) GOTO 62050                                ZEX00509
LN 0529     62035 NCARD=1                                                    ZEX00510
LN 0530          INREG=INREG-1                                               ZEX00511
LN 0531          GOTO 900                                                    ZEX00512
LN 0532     C**** ALPANUMERIC CONSTANT WITHIN PRINT USING-STATEMENT          ZEX00513
LN 0533     63000 INREG=INREG-1                                              ZEX00514
LN 0534          NCODE=2                                                     ZEX00515
LN 0535          CALL ZIMAGE(NCODE,NSTOP)                                    ZEX00516
LN 0536          IF(NSTOP.EQ.-1) GOTO 21000                                  ZEX00517
LN 0537          GOTO 900                                                    ZEX00518
LN 0538     C**** NUMERIC VARIABLE OR EXPRESSION WITHIN PRINT USING          ZEX00519
LN 0539     64000 INREG=INREG-1                                              ZEX00520
LN 0540          NCODE=1                                                     ZEX00521
LN 0541          CALL ZIMAGE(NCODE,NSTOP)                                    ZEX00522
LN 0542          IF(NSTOP.EQ.-1) GOTO 21000                                  ZEX00523
LN 0543          GOTO 900                                                    ZEX00524
LN 0544     C**** END OF LINE WITHIN PRINT USING COMMAND                     ZEX00525
LN 0545     65000 INREG=INREG-1                                              ZEX00526
LN 0546          NPRI=1                                                      ZEX00527
LN 0547          IF(IPROG(INREG+2).NE.-52) GOTO 65001                        ZEX00528
LN 0548          NCODE=9                                                     ZEX00529
LN 0549          CALL ZIMAGE(NCODE,NSTOP)                                    ZEX00530
```

```
LN 0550          GOTO 900                                                    ZEX00531
LN 0551    65001 NX=2                                                        ZEX00532
LN 0552          CALL PRILIN(NX)                                             ZEX00533
LN 0553          NCARD=1                                                     ZEX00534
LN 0554          GOTO 900                                                    ZEX00535
LN 0555    C**** ALPHANUMERIC VARIABLE WITHIN PRINT USING COMMAND            ZEX00536
LN 0556    66000 INREG=INREG-1                                               ZEX00537
LN 0557          NCODE=3                                                     ZEX00538
LN 0558          CALL ZIMAGE(NCODE,NSTOP)                                    ZEX00539
LN 0559          IF(NSTOP.EQ.-1) GOTO 21000                                  ZEX00540
LN 0560          GOTO 900                                                    ZEX00541
LN 0561    70000 INREG=INREG-1                                               ZEX00542
LN 0562          NCODE=9                                                     ZEX00543
LN 0563          CALL ZIMAGE(NCODE,NSTOP)                                    ZEX00544
LN 0564          IF(NSTOP.EQ.-1) GOTO 21000                                  ZEX00545
LN 0565          GOTO 900                                                    ZEX00546
LN 0566    C**** ALTER NPRI                                                  ZEX00547
LN 0567    C**** COMMA  NPRI=1                                               ZEX00548
LN 0568    C**** PUCO  NPRI=-1                                               ZEX00549
LN 0569    67000 IF(NPRI.EQ.1) GOTO 67005                                    ZEX00550
LN 0570          NPRI=1                                                      ZEX00551
LN 0571          I1=INEXT                                                    ZEX00552
LN 0572          I2=INEXT                                                    ZEX00553
LN 0573          IF(NPRUS.GT.INEXT) I2=NPRUS                                 ZEX00554
LN 0574          CALL CHECK(I1,I2)                                           ZEX00555
LN 0575          GOTO 67005                                                  ZEX00556
LN 0576    69000 IF(NPRI.EQ.-1) GOTO 67005                                   ZEX00557
LN 0577          DO 67002 I=1,125                                            ZEX00558
LN 0578          J=125-I+1                                                   ZEX00559
LN 0579          IF(CARP(J).NE.BLANK) GOTO 67003                             ZEX00560
LN 0580    67002 CONTINUE                                                    ZEX00561
LN 0581          J=0                                                         ZEX00562
LN 0582    67003 INEXT=J+1                                                   ZEX00563
LN 0583          NPRI=-1                                                     ZEX00564
LN 0584    67005 INREG=INREG-1                                               ZEX00565
LN 0585          GOTO 900                                                    ZEX00566
LN 0586    C**** COMMAND IS TAB                                              ZEX00567
LN 0587    68000 IF(ACC.GE.0.) GOTO 68001                                    ZEX00568
LN 0588          NERROR=39                                                   ZEX00569
LN 0589          X1=ACC                                                      ZEX00570
LN 0590          GOTO 29008                                                  ZEX00571
LN 0591    68001 ICC=INT(ACC+0.5)                                            ZEX00572
LN 0592          IF(ICC.EQ.0) ICC=1                                          ZEX00573
LN 0593          IF(ICC.GT.IWRIT) GOTO 68002                                 ZEX00574
LN 0594    68006 IF(ICC.LT.INEXT) GOTO 68003                                 ZEX00575
LN 0595          GOTO 68004                                                  ZEX00576
LN 0596    68003 IF(CARP(INEXT-1).NE.BLANK) GOTO 68005                       ZEX00577
LN 0597          INEXT=INEXT-1                                               ZEX00578
LN 0598          GOTO 68006                                                  ZEX00579
LN 0599    68002 MIV=ICC/IWRIT                                               ZEX00580
LN 0600          ICC=ICC-IWRIT*MIV                                           ZEX00581
LN 0601    68005 NX=1                                                        ZEX00582
LN 0602          CALL PRILIN(NX)                                             ZEX00583
LN 0603    68004 INEXT=ICC                                                   ZEX00584
LN 0604          INREG=INREG-1                                               ZEX00585
LN 0605          NPRI=-1                                                     ZEX00586
LN 0606          GOTO 900                                                    ZEX00587
LN 0607    C***** RETURN TO MAIN PROGRAMM                                    ZEX00588
LN 0608    C---- DAS FOLGENDE STATEMENT IST DURCH DIE FOLGENDE ANWEISUNG      ZEX00589
LN 0609    C----    21000 RETURN                                             ZEX00590
LN 0610    C---- ZU ERSETZEN,WENN NICHT IN *OVERLAY-TECHNIK* GEARBEITET WIRD, ZEX00591
```

```
LN 0611     C---- WOBEI DIE ZAHL 21000 IN DER ERSTEN SPALTE BEGINNEN MUSS.      ZEX00592
LN 0612     21000 CONTINUE                                                     ZEX00593
LN 0613           END                                                          ZEX00594

      USASI FORTRAN DIAGNOSTIC RESULTS FOR EXECUT

   LINE   S   ERRNUM                    MESSAGE

   0006   N     0001      PROGRAM HAS PROGRAM STATEMENT.

   0028   N     0026      NON-USAST OVERLAY OR SEGMENT STATEMENT USED.
```

```
LN 0001          SUBROUTINE ZEVAL (NSTOP)                                    ZEV00001
LN 0002     C                                                                ZEV00002
LN 0003     C*****SUBROUTINE TO EVALUATE EXPRESSIONS IN REVERSE POLISH NOTATION ZEV00003
LN 0004     C                                                                ZEV00004
LN 0005          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,ZEV00005
LN 0006         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,ZEV00006
LN 0007         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL, ZEV00007
LN 0008         3IRC,IWC,NSTEND,IEXPO,IREGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,ZEV00008
LN 0009         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI   ZEV00009
LN 0010          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,ZEV00010
LN 0011         1NIFMAX,INTZEI                                               ZEV00011
LN 0012          COMMON// CARDT(80),MERKER(26,2),CARP(140),                  ZEV00012
LN 0013         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),ZEV00013
LN 0014         1IRET(20),XXX(4),NFILE(25,3)                                 ZEV00014
LN 0015          COMMON// ISTLST(340),LISTST(340)                            ZEV00015
LN 0016          COMMON// DATAN(330)                                         ***
LN 0017          COMMON// DATA(3700)                                         ZEV00017
LN 0018          DIMENSION IPROG(3700)                                       ZEV00018
LN 0019          EQUIVALENCE (DATA(1),IPROG(1))                              ZEV00019
LN 0020          DIMENSION FEXP(100),IEXP(100)                               ZEV00020
LN 0021          EQUIVALENCE(IEXP(1),FEXP(1))                                ZEV00021
LN 0022     C                                                                ZEV00022
LN 0023     C*****INSERT STRING IN IEXP                                      ZEV00023
LN 0024          INREG=INREG-1                                               ZEV00024
LN 0025          LNGTH=IPROG(INREG)                                          ZEV00025
LN 0026          DO 5 I=1,LNGTH                                              ZEV00026
LN 0027          ILOC=INREG-I                                                ZEV00027
LN 0028      5   IEXP(I)=IPROG(ILOC)                                         ZEV00028
LN 0029     C*****RESET INREG                                                ZEV00029
LN 0030          INREG=INREG-1-LNGTH                                         ZEV00030
LN 0031     C                                                                ZEV00031
LN 0032     C*****EXPRESSION IS IN IEXP(1) THROUGH IEXP(LNGTH)               ZEV00032
LN 0033     C*****OPERANDS ARE INDICATED BY THEIR POSITIONS IN VECTOR DATA   ZEV00033
LN 0034     C                                                                ZEV00034
LN 0035     C*****OPERATIONS ARE --                                          ZEV00035
LN 0036     C*****-1 ADDITION                                                ZEV00036
LN 0037     C*****-2 SUBTRACTION                                             ZEV00037
LN 0038     C*****-3 MULTIPLICATION                                          ZEV00038
LN 0039     C*****-4 DIVISION                                                ZEV00039
LN 0040     C*****-5 EXPONENTIATION                                          ZEV00040
LN 0041     C*****-6 UNARY MINUS                                             ZEV00041
LN 0042     C*****-7 INDIRECT FOR SUBSCRIPTED VARIABLE WITHOUT A SUBSCRIPT   ZEV00042
LN 0043     C*****-8 INDIRECT FOR SUBSCRIPTED VARIABLE WITH A SUBSCRIPT      ZEV00043
LN 0044     C*****-9 SIN                                                     ZEV00044
LN 0045     C*****-10 COS                                                    ZEV00045
LN 0046     C*****-11 TAN                                                    ZEV00046
LN 0047     C*****-12 ATN                                                    ZEV00047
LN 0048     C*****-13 EXP                                                    ZEV00048
LN 0049     C*****-14 ABS                                                    ZEV00049
LN 0050     C*****-15 LOG                                                    ZEV00050
LN 0051     C*****-16 SQR                                                    ZEV00051
LN 0052     C*****-17 INT                                                    ZEV00052
LN 0053     C*****-18 RND                                                    ZEV00053
LN 0054     C*****-19 STORE ACCUMULATOR IN SUBSCRIPTED POSITION              ZEV00054
LN 0055     C*****-20 STORE ALPHANUMERIC CODE INTO ACCUMULATOR               ZEV00055
LN 0056     C                                                                ZEV00056
LN 0057     C*****SET CURRENT LAST POSITION                                  ZEV00057
LN 0058          ILAST=0                                                     ZEV00058
LN 0059          DO 1000 I=1,LNGTH                                           ZEV00059
LN 0060          IENTRY=IEXP(I)                                              ZEV00060
LN 0061          IF(IENTRY.GT.0) GOTO 900                                    ZEV00061
```

```
LN 0062     C*****ENTRY IS AN OPERAND -- PROCESS IT                                  ZEV00062
LN 0063           IENTRY=-IENTRY                                                    ZEV00063
LN 0064           IF(IENTRY.EQ.20) GOTO 200                                         ZEV00064
LN 0065           GOTO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,     ZEV00065
LN 0066          1 170,180,190), IENTRY                                             ZEV00066
LN 0067     C                                                                       ZEV00067
LN 0068      10   ILAST=ILAST-1                                                     ZEV00068
LN 0069           FEXP(ILAST)=FEXP(ILAST)+FEXP(ILAST+1)                             ZEV00069
LN 0070           GOTO 1000                                                         ZEV00070
LN 0071     C                                                                       ZEV00071
LN 0072      20   ILAST=ILAST-1                                                     ZEV00072
LN 0073           FEXP(ILAST)=FEXP(ILAST)-FEXP(ILAST+1)                             ZEV00073
LN 0074           GOTO 1000                                                         ZEV00074
LN 0075     C                                                                       ZEV00075
LN 0076      30   ILAST=ILAST-1                                                     ZEV00076
LN 0077           IF((FEXP(ILAST).EQ.0.).OR.(FEXP(ILAST+1).EQ.0.)) GOTO 36          ZEV00077
LN 0078           IF(ALOG10(ABS(FEXP(ILAST)))+ALOG10(ABS(FEXP(ILAST+1))).GT.FLOAT(  ZEV00078
LN 0079          1IEXPO)) GOTO 35                                                   ZEV00079
LN 0080      36   FEXP(ILAST)=FEXP(ILAST)*FEXP(ILAST+1)                             ZEV00080
LN 0081           GOTO 1000                                                         ZEV00081
LN 0082      35   NERROR=40                                                         ZEV00082
LN 0083           X1=FEXP(ILAST)                                                    ZEV00083
LN 0084           X2=FEXP(ILAST+1)                                                  ZEV00084
LN 0085           GOTO 44                                                           ZEV00085
LN 0086     C                                                                       ZEV00086
LN 0087      40   ILAST=ILAST-1                                                     ZEV00087
LN 0088           IF(FEXP(ILAST+1).NE.0.) GOTO 42                                   ZEV00088
LN 0089           NERROR=14                                                         ZEV00089
LN 0090           GOTO 44                                                           ZEV00090
LN 0091      42   FEXP(ILAST)=FEXP(ILAST) /FEXP(ILAST+1)                            ZEV00091
LN 0092           GOTO 1000                                                         ZEV00092
LN 0093     C                                                                       ZEV00093
LN 0094      50   ILAST=ILAST-1                                                     ZEV00094
LN 0095           IF(FEXP(ILAST).EQ.0.) GOTO 58                                     ZEV00095
LN 0096           IF(FEXP(ILAST+1).EQ.0.) GOTO 51                                   ZEV00096
LN 0097           IF(ABS(ALOG10(ABS(FEXP(ILAST)))*FEXP(ILAST+1)).GT.FLOAT(IEXPO))   ZEV00097
LN 0098          1 GOTO 57                                                          ZEV00098
LN 0099           IF(FEXP(ILAST+1).GT.0.) GOTO 56                                   ZEV00099
LN 0100           FEXP(ILAST+1)=-FEXP(ILAST+1)                                      ZEV00100
LN 0101           FEXP(ILAST)=1./FEXP(ILAST)                                        ZEV00101
LN 0102      56   ILOC=FEXP(ILAST+1)                                                ZEV00102
LN 0103           XLOC=ILOC                                                         ZEV00103
LN 0104           IF(XLOC.EQ.FEXP(ILAST+1)) GOTO 54                                 ZEV00104
LN 0105           IF(FEXP(ILAST).GE.0.) GOTO 53                                     ZEV00105
LN 0106           XLOC=ABS(FEXP(ILAST))**FEXP(ILAST+1)                              ZEV00106
LN 0107           ACC=-1.*XLOC**(1./FEXP(ILAST+1))                                  ZEV00107
LN 0108           IF(ACC.NE.FEXP(ILAST)) GOTO 55                                    ZEV00108
LN 0109           FEXP(ILAST)=-XLOC                                                 ZEV00109
LN 0110           GOTO 1000                                                         ZEV00110
LN 0111      55   NERROR=15                                                         ZEV00111
LN 0112           X1=FEXP(ILAST)                                                    ZEV00112
LN 0113           X2=FEXP(ILAST+1)                                                  ZEV00113
LN 0114           GOTO 44                                                           ZEV00114
LN 0115      57   NERROR=16                                                         ZEV00115
LN 0116           GOTO 44                                                           ZEV00116
LN 0117      53   FEXP(ILAST)=FEXP(ILAST)**FEXP(ILAST+1)                            ZEV00117
LN 0118           GOTO 1000                                                         ZEV00118
LN 0119      54   FEXP(ILAST)=FEXP(ILAST)**ILOC                                     ZEV00119
LN 0120           GOTO 1000                                                         ZEV00120
LN 0121      58   IF(FEXP(ILAST+1).EQ.0.) GOTO 51                                   ZEV00121
LN 0122           FEXP(ILAST)=0.                                                    ZEV00122
```

```
LN 0123          GOTO 1000                                                  ZEV00123
LN 0124     51   FEXP(ILAST)=1.                                             ZEV00124
LN 0125          GOTO 1000                                                  ZEV00125
LN 0126     C                                                               ZEV00126
LN 0127     60   FEXP(ILAST)=-FEXP(ILAST)                                   ZEV00127
LN 0128          GOTO 1000                                                  ZEV00128
LN 0129     C                                                               ZEV00129
LN 0130     70   ILOC=FEXP(ILAST)+1.5                                       ZEV00130
LN 0131          FEXP(ILAST)=DATA(ILOC)                                     ZEV00131
LN 0132          GOTO 1000                                                  ZEV00132
LN 0133     C                                                               ZEV00133
LN 0134     80   ILAST=ILAST-1                                              ZEV00134
LN 0135          IBEG=FEXP(ILAST)+.5                                        ZEV00135
LN 0136          ITEM=FEXP(ILAST+1)+.5                                      ZEV00136
LN 0137          LOC=IBEG+ITEM+1                                            ZEV00137
LN 0138          IUP=DATA(IBEG)+.5                                          ZEV00138
LN 0139          IF((LOC.GT.IBEG).AND.(LOC.LT.IUP)) GOTO 85                 ZEV00139
LN 0140     801  DO 81 K=1,26                                              ZEV00140
LN 0141          LOCN=DATA(K)+.5                                            ZEV00141
LN 0142          IF(LOCN.EQ.IBEG) GOTO 82                                   ZEV00142
LN 0143     81   CONTINUE                                                   ZEV00143
LN 0144          K=37                                                       ZEV00144
LN 0145     82   NERROR=17                                                  ZEV00145
LN 0146          I1=ITEM                                                    ZEV00146
LN 0147          X1=ALPH(K)                                                 ZEV00147
LN 0148     44   NSTOP=1                                                    ZEV00148
LN 0149          CALL EXERR(NERROR,I1,I2,X1,X2)                             ZEV00149
LN 0150          RETURN                                                     ZEV00150
LN 0151     C                                                               ZEV00151
LN 0152     85   FEXP(ILAST)=DATA(LOC)                                      ZEV00152
LN 0153          GOTO 1000                                                  ZEV00153
LN 0154     C                                                               ZEV00154
LN 0155     90   FEXP(ILAST)=SIN(FEXP(ILAST))                               ZEV00155
LN 0156          GOTO 1000                                                  ZEV00156
LN 0157     C                                                               ZEV00157
LN 0158     100  FEXP(ILAST)=COS(FEXP(ILAST))                               ZEV00158
LN 0159          GOTO 1000                                                  ZEV00159
LN 0160     C                                                               ZEV00160
LN 0161     110  IF(COS(FEXP(ILAST)).EQ.0.) GOTO 115                        ZEV00161
LN 0162          FEXP(ILAST)=SIN(FEXP(ILAST))/COS(FEXP(ILAST))              ZEV00162
LN 0163          GOTO 1000                                                  ZEV00163
LN 0164     115  NERROR=38                                                  ZEV00164
LN 0165          GOTO 44                                                    ZEV00165
LN 0166     C                                                               ZEV00166
LN 0167     120  FEXP(ILAST)=ATAN(FEXP(ILAST))                              ZEV00167
LN 0168          GOTO 1000                                                  ZEV00168
LN 0169     C                                                               ZEV00169
LN 0170     130  FEXP(ILAST)=EXP(FEXP(ILAST))                              ZEV00170
LN 0171          GOTO 1000                                                  ZEV00171
LN 0172     C                                                               ZEV00172
LN 0173     140  FEXP(ILAST)=ABS(FEXP(ILAST))                              ZEV00173
LN 0174          GOTO 1000                                                  ZEV00174
LN 0175     C                                                               ZEV00175
LN 0176     150  IF(FEXP(ILAST).GT.0.) GOTO 151                           ZEV00176
LN 0177          NERROR=18                                                  ZEV00177
LN 0178          IF(FEXP(ILAST).LT.0.) NERROR=19                            ZEV00178
LN 0179          GOTO 44                                                    ZEV00179
LN 0180     151  FEXP(ILAST)=ALOG(FEXP(ILAST))                              ZEV00180
LN 0181          GOTO 1000                                                  ZEV00181
LN 0182     C                                                               ZEV00182
LN 0183     160  IF(FEXP(ILAST).GE.0.) GOTO 161                            ZEV00183
```

```
LN 0184           NERROR=20                                                  ZEV00184
LN 0185           GOTO 44                                                    ZEV00185
LN 0186      161  FEXP(ILAST)=SQRT(FEXP(ILAST))                              ZEV00186
LN 0187           GOTO 1000                                                  ZEV00187
LN 0188      C                                                               ZEV00188
LN 0189      170  IF(FEXP(ILAST).LT.0.) FEXP(ILAST)=FEXP(ILAST)-1.          ZEV00189
LN 0190           FEXP(ILAST)=AINT(FEXP(ILAST))                              ZEV00190
LN 0191           GOTO 1000                                                  ZEV00191
LN 0192      C                                                               ZEV00192
LN 0193      C*****RANDOM NUMBER CALL                                        ZEV00193
LN 0194      180  FEXP(ILAST)=RNG(1)                                        ZEV00194
LN 0195           GOTO 1000                                                  ZEV00195
LN 0196      C                                                               ZEV00196
LN 0197      C******INSERT ACCUMULATOR IN SUBSCRIPTED VARIABLE               ZEV00197
LN 0198      190  ILAST=ILAST-1                                             ZEV00198
LN 0199           IBEG=FEXP(ILAST)+.5                                        ZEV00199
LN 0200           ITEM=FEXP(ILAST+1)+.5                                      ZEV00200
LN 0201           LOC=IBEG+ITEM+1                                            ZEV00201
LN 0202           IUP=DATA(IBEG)+.5                                          ZEV00202
LN 0203           IF((LOC.GT.IBEG).AND.(LOC.LT.IUP)) GOTO 195                ZEV00203
LN 0204           GOTO 901                                                   ZEV00204
LN 0205      195  DATA(LOC)=ACC                                              ZEV00205
LN 0206           GOTO 1000                                                  ZEV00206
LN 0207      C                                                               ZEV00207
LN 0208      C*****IEXP(I) CONTAINS THE ADDRESS OF AN OPERAND                ZEV00208
LN 0209      C*****INSERT VALUE IN NEXT AVAILABLE POSITION                   ZEV00209
LN 0210      900  ILAST=ILAST+1                                             ZEV00210
LN 0211           FEXP(ILAST)=DATA(IENTRY)                                   ZEV00211
LN 0212      C                                                               ZEV00212
LN 0213      C*****STORE ALPHANUMERIC CODE OF CONSTANT INTO ACCUMULATOR      ZEV00213
LN 0214           GOTO 1000                                                  ZEV00214
LN 0215      200  FEXP(1)=IEXP(I+1)                                         ZEV00215
LN 0216           GOTO 1001                                                  ZEV00216
LN 0217      1000 CONTINUE                                                   ZEV00217
LN 0218      C                                                               ZEV00218
LN 0219      C*****INSERT RESULT IN ACCUMULATOR                              ZEV00219
LN 0220      1001 ACC=FEXP(1)                                                ZEV00220
LN 0221      C                                                               ZEV00221
LN 0222           NSTOP=0                                                    ZEV00222
LN 0223           RETURN                                                     ZEV00223
LN 0224           END                                                        ZEV00224

     USASI FORTRAN DIAGNOSTIC RESULTS FOR ZEVAL


          NO ERRORS
```

ZINSNO

```
LN 0001          SUBROUTINE ZINSNO                                              ZIS00001
LN 0002    C*****SUBROUTINE TO INSERT NUMBER FROM ACCUMULATOR INTO NEXT PRINT POSITZIS00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    ZIS00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST, INEXT,ZIS00004
LN 0005          2NUMBUF,PARLET,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,ZIS00005
LN 0006          3IRC,IWC,NSTEND,IEXPO,I3EGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,ZIS00006
LN 0007          4NPRUS,NCARD,MAXIMA,PUCO,DOPU,FXSIGN,MAXSAT,NUMFIL,NZTM,NSTZEI   ZIS00007
LN 0008           COMMON// INTMAX,INTNUM,XNULL,DOOPU,IMIRC,SMALL,ISTMAX,NIPMAX,   ZIS00008
LN 0009          1NIFMAX,INTZEI                                                 ZIS00009
LN 0010           COMMON// CARDI(80),MERKER(26,2),CARP(140),                     ZIS00010
LN 0011          3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),   ZIS00011
LN 0012          1IRET(20),XXX(4),NFILE(25,3)                                   ZIS00012
LN 0013           COMMON// ISTLST(340),LISTST(340)                               ZIS00013
LN 0014           COMMON// DATAN(330)                                            ***
LN 0015           COMMON// DATA(3700)                                            ZIS00015
LN 0016           DIMENSION IPROG(3700)                                          ZIS00016
LN 0017           EQUIVALENCE (DATA(1),IPROG(1))                                 ZIS00017
LN 0018           DIMENSION CHAR(6)                                              ZIS00018
LN 0019           IST=INEXT-IZONE                                                ZIS00019
LN 0020           ISGN=0                                                         ZIS00020
LN 0021           IF(ACC.GT.0.) GOTO 110                                         ZIS00021
LN 0022           ACC=-ACC                                                       ZIS00022
LN 0023           ISGN=1                                                         ZIS00023
LN 0024    C*****TEST FOR ZERO                                                   ZIS00024
LN 0025     110   IF(ACC.GT.SMALL) GOTO 200                                      ZIS00025
LN 0026    C*****NUMBER IS TO BE PRINTED AS 0.0                                  ZIS00026
LN 0027     150   CARP(IST+8)=DIGIT(1)                                           ZIS00027
LN 0028           CARP(IST+9)=DECMAL                                             ZIS00028
LN 0029           CARP(IST+10)=DIGIT(1)                                          ZIS00029
LN 0030           GOTO 1000                                                      ZIS00030
LN 0031    C                                                                     ZIS00031
LN 0032    C*****FIND FRACTIONAL PART AND EXPONENT                               ZIS00032
LN 0033     200   ZLOG=ALOG10(ACC)                                               ZIS00033
LN 0034           IEXPN=ZLOG                                                     ZIS00034
LN 0035           IF(ZLOG.LT.0.) IEXPN=IEXPN-1                                   ZIS00035
LN 0036           FEXPN=IEXPN                                                    ZIS00036
LN 0037           FRAC=10.**(ZLOG-FEXPN)                                         ZIS00037
LN 0038     250   IFRAC=(ACC*(10.**(5-IEXPN)))+.5                                ZIS00038
LN 0039           IF(IFRAC.LE.999999) GOTO 280                                   ZIS00039
LN 0040           IEXPN=IEXPN+1                                                  ZIS00040
LN 0041           GOTO 250                                                       ZIS00041
LN 0042    C*****LOAD CHARACTERS                                                 ZIS00042
LN 0043     280   DO 300 I=1,6                                                   ZIS00043
LN 0044           IDIV=6-I                                                       ZIS00044
LN 0045           IX=IFRAC/(10**IDIV)                                            ZIS00045
LN 0046           CHAR(I)=DIGIT(IX+1)                                            ZIS00046
LN 0047     300   IFRAC=IFRAC-(IX*(10**IDIV))                                    ZIS00047
LN 0048    C                                                                     ZIS00048
LN 0049    C*****FIND LENGTH                                                     ZIS00049
LN 0050           LNGTH=6                                                        ZIS00050
LN 0051           DO 350 I=1,6                                                   ZIS00051
LN 0052           IPOS=7-I                                                       ZIS00052
LN 0053           IF(CHAR(IPOS).NE.DIGIT(1)) GOTO 380                            ZIS00053
LN 0054     350   LNGTH=LNGTH-1                                                  ZIS00054
LN 0055     380   IF(LNGTH.EQ.0) GOTO 150                                        ZIS00055
LN 0056    C                                                                     ZIS00056
LN 0057    C*****CHECK FOR A NUMBER GREATER THAN OR EQUAL TO 1                   ZIS00057
LN 0058           IF(IEXPN.GE.0) GOTO 600                                        ZIS00058
LN 0059    C*****NUMBER IS FRACTIONAL                                            ZIS00059
LN 0060           LNGTOT=LNGTH-IEXPN                                             ZIS00060
LN 0061           IF(LNGTOT.GT.12) GOTO 900                                      ZIS00061
```

```
LN 0062           LOCD=9                                                  ZIS00062
LN 0063           IF(LNGTOT.GT.6) LOCD=15-LNGTOT                          ZIS00063
LN 0064         C*****INSERT SIGN IF NECESSARY                            ZIS00064
LN 0065           LOCSGN=IST+LOCD-1                                       ZIS00065
LN 0066           CARP(LOCSGN)=DIGIT(1)                                   ZIS00066
LN 0067           IF(ISGN.EQ.0) GOTO 450                                  ZIS00067
LN 0068           CARP(LOCSGN)=CMINUS                                     ZIS00068
LN 0069       450 LOCDEC=IST+LOCD                                         ZIS00069
LN 0070           CARP(LOCDEC)=DECMAL                                     ZIS00070
LN 0071           IFSTL=IST+LOCD-IEXPN                                    ZIS00071
LN 0072           IFSTLM=IFSTL-1                                          ZIS00072
LN 0073           IF(IFSTL.EQ.(IST+LOCD+1)) GOTO 500                      ZIS00073
LN 0074           IBEG=IST+LOCD+1                                         ZIS00074
LN 0075           DO 475 I=IBEG,IFSTLM                                    ZIS00075
LN 0076       475 CARP(I)=DIGIT(1)                                        ZIS00076
LN 0077       500 DO 510 I=1,LNGTH                                        ZIS00077
LN 0078           CARP(IFSTL)=CHAR(I)                                     ZIS00078
LN 0079       510 IFSTL=IFSTL+1                                           ZIS00079
LN 0080         C*****COMPLETE                                            ZIS00080
LN 0081           GOTO 1000                                               ZIS00081
LN 0082         C                                                         ZIS00082
LN 0083         C*****NUMBER IS NOT FRACTIONAL                            ZIS00083
LN 0084         C*****NUMBER IS GREATER THAN OR EQUAL TO 1                ZIS00084
LN 0085         C*****SEE IF NUMBER IS TOO BIG TO PRINT IN NORMAL FORM    ZIS00085
LN 0086       600 IF(IEXPN.GT.12) GOTO 900                                ZIS00086
LN 0087           LOCD=9                                                  ZIS00087
LN 0088           IF(IEXPN.GT.6) LOCD=IEXPN+3                             ZIS00088
LN 0089           IFSTL=LOCD-1+IST-IEXPN                                  ZIS00089
LN 0090         C*****INSERT SIGN IF NECESSARY                            ZIS00090
LN 0091           IF(ISGN.EQ.0) GOTO 650                                  ZIS00091
LN 0092           CARP(IFSTL-1)=CMINUS                                    ZIS00092
LN 0093         C*****INSERT CHARACTERS                                   ZIS00093
LN 0094       650 DO 675 I=1,LNGTH                                        ZIS00094
LN 0095           CARP(IFSTL)=CHAR(I)                                     ZIS00095
LN 0096       670 IFSTL=IFSTL+1                                           ZIS00096
LN 0097           IF(IFSTL.EQ.(IST+LOCD)) GOTO 670                        ZIS00097
LN 0098       675 CONTINUE                                                ZIS00098
LN 0099         C                                                         ZIS00099
LN 0100         C*****CHECK FOR INTEGER                                   ZIS00100
LN 0101           IF(IFSTL.EQ.(IST+LOCD+1)) GOTO 1000                     ZIS00101
LN 0102           IF(IFSTL.LE.(IST+LOCD-1)) GOTO 700                      ZIS00102
LN 0103         C*****INSERT DECIMAL                                      ZIS00103
LN 0104           LOCDEC=IST+LOCD                                         ZIS00104
LN 0105           CARP(LOCDEC)=DECMAL                                     ZIS00105
LN 0106           GOTO 1000                                               ZIS00106
LN 0107         C*****INSERT ZEROES                                       ZIS00107
LN 0108       700 IEND=IST+LOCD-1                                         ZIS00108
LN 0109           IBEG=IFSTL                                              ZIS00109
LN 0110           DO 725 I=IBEG,IEND                                      ZIS00110
LN 0111       725 CARP(I)=DIGIT(1)                                        ZIS00111
LN 0112           GOTO 1000                                               ZIS00112
LN 0113         C                                                         ZIS00113
LN 0114         C*****INSERT NUMBER IN EXPONENTIAL FORM                   ZIS00114
LN 0115       900 IF(ISGN.EQ.1) CARP(IST+1)=CMINUS                        ZIS00115
LN 0116           CARP(IST+2)=CHAR(1)                                     ZIS00116
LN 0117           CARP(IST+3)=DECMAL                                      ZIS00117
LN 0118           DO 925 I=2,6                                            ZIS00118
LN 0119           LOC=IST+2+I                                             ZIS00119
LN 0120       925 CARP(LOC)=CHAR(I)                                       ZIS00120
LN 0121           CARP(IST+9)=ALPH(5)                                     ZIS00121
LN 0122           CARP(IST+10)=PLUS                                       ZIS00122
```

```
LN 0123           IF(IEXPN.GT.0) GOTO 950                                ZIS00123
LN 0124           CARP(IST+1)=CMINUS                                     ZIS00124
LN 0125           IEXPN=-IEXPN                                           ZIS00125
LN 0126     C                                                            ZIS00126
LN 0127     C*****INSERT EXPONENT                                        ZIS00127
LN 0128      950  IF(IEXPN.GT.9) GOTO 970                               ZIS00128
LN 0129           CARP(IST+11)=DIGIT(IEXPN+1)                            ZIS00129
LN 0130           GOTO 1000                                              ZIS00130
LN 0131      970  IF(IEXPN.GT.99) GOTO 980                              ZIS00131
LN 0132           IX=IEXPN/10                                            ZIS00132
LN 0133           CARP(IST+11)=DIGIT(IX+1)                               ZIS00133
LN 0134           IZ=IEXPN-(10*IX)                                       ZIS00134
LN 0135           CARP(IST+12)=DIGIT(IZ+1)                               ZIS00135
LN 0136           GOTO 1000                                              ZIS00136
LN 0137      980  IF(IEXPN.GT.IEXPO) GOTO 990                           ZIS00137
LN 0138           IX=IEXPN/100                                           ZIS00138
LN 0139           CARP(IST+11)=DIGIT(IX+1)                               ZIS00139
LN 0140           IZ=IEXPN-(100*IX)                                      ZIS00140
LN 0141           CARP(IST+12)=DIGIT(IZ+1)                               ZIS00141
LN 0142           LOC=IEXPN-(100*IX)-(10*IZ)                             ZIS00142
LN 0143           CARP(IST+13)=DIGIT(LOC+1)                              ZIS00143
LN 0144           GOTO 1000                                              ZIS00144
LN 0145      990  CARP(IST+11)=ASTRSK                                    ZIS00145
LN 0146           CARP(IST+12)=ASTRSK                                    ZIS00146
LN 0147           CARP(IST+13)=ASTRSK                                    ZIS00147
LN 0148     1000  IF(NPRI.EQ.1) RETURN                                   ZIS00148
LN 0149           IX=INEXT                                               ZIS00149
LN 0150           DO 1010 I=IST,IX                                       ZIS00150
LN 0151           IF(CARP(I).NE.BLANK) GOTO 1020                         ZIS00151
LN 0152     1010  CONTINUE                                              ZIS00152
LN 0153     1020  IBEG=I                                                ZIS00153
LN 0154           DO 1030 LOC=IBEG,IX                                    ZIS00154
LN 0155           IF(CARP(LOC).EQ.BLANK) GOTO 1040                       ZIS00155
LN 0156     1030  CONTINUE                                              ZIS00156
LN 0157     1040  IF((IST+LOC-IBEG+1).LT.IWRIT) GOTO 1050               ZIS00157
LN 0158           WRITE(IWO,1060) (CARP(I),I=1,IST)                      ZIS00158
LN 0159     1060  FORMAT(5X,125A1)                                       ZIS00159
LN 0160           IZ=LOC-IBEG                                            ZIS00160
LN 0161           DO 1070 I=1,IZ                                         ZIS00161
LN 0162           IX=IBEG+I-1                                            ZIS00162
LN 0163           CARP(I)=CARP(IX)                                       ZIS00163
LN 0164     1070  CONTINUE                                              ZIS00164
LN 0165           CALL CLEAR(IZ+1,140)                                   ZIS00165
LN 0166           INEXT=IZ+2                                             ZIS00166
LN 0167           RETURN                                                 ZIS00167
LN 0168     1050  LOC=LOC-1                                             ZIS00168
LN 0169           IF(IST.EQ.1) GOTO 1051                                 ZIS00169
LN 0170           IF(IBEG.LT.IST+2) RETURN                               ZIS00170
LN 0171           IF(CARP(IST-1).NE.BLANK) IST=IST+1                     ZIS00171
LN 0172     1051  DO 1090 I=IBEG,LOC                                    ZIS00172
LN 0173           CARP(IST)=CARP(I)                                      ZIS00173
LN 0174           CARP(I)=BLANK                                          ZIS00174
LN 0175           IST=IST+1                                              ZIS00175
LN 0176     1090  CONTINUE                                              ZIS00176
LN 0177           INEXT=IST+1                                            ZIS00177
LN 0178           RETURN                                                 ZIS00178
LN 0179           END                                                    ZIS00179

      USASI FORTRAN DIAGNOSTIC RESULTS FOR ZINSNO


          NO ERRORS
```

```
LN 0001          SUBROUTINE ZEXFIL(IOP)                                          EXF00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    EXF00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT, EXF00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  EXF00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, EXF00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     EXF00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    EXF00007
LN 0008         1NIFMAX,INTZEI                                                   EXF00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                      EXF00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    EXF00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                     EXF00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                EXF00012
LN 0013          COMMON// DATAN(330)                                             ***
LN 0014          COMMON// DATA(3700)                                             EXF00014
LN 0015          DIMENSION IPROG(3700)                                           EXF00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                  EXF00016
LN 0017          IF(IOP.GT.14) GOTO 62000                                        EXF00017
LN 0018          IF(IOP.GT.0) GOTO 1000                                          EXF00018
LN 0019          IF(NFILE(1,1).EQ.-1) RETURN                                     EXF00019
LN 0020    C**** REWIND COMMON-FILE                                              EXF00020
LN 0021          DO 10 I=1,NUMFIL                                                EXF00021
LN 0022          MIC=NFILE(I,3)                                                  EXF00022
LN 0023          DO 20 LOCU=1,MIC                                                EXF00023
LN 0024          LOCS=NFILE(I,2)+LOCU-1                                          EXF00024
LN 0025          REWIND LOCS                                                     EXF00025
LN 0026     20   CONTINUE                                                        EXF00026
LN 0027     10   CONTINUE                                                        EXF00027
LN 0028          RETURN                                                          EXF00028
LN 0029     1000 GOTO(48000,49000,50000,51000,52000,53000,54000,55000,56000,57000,5EXF00029
LN 0030         18000,59000,60000,61000),IOP                                     EXF00030
LN 0031    C**** COMMAND IS OPEN                                                 EXF00031
LN 0032    48000 IX=IPROG(INREG-1)                                               EXF00032
LN 0033          NWORT=IPROG(INREG-3)                                            EXF00033
LN 0034          IF(NUMFIL.EQ.0) GOTO 48007                                      EXF00034
LN 0035          ITOP=1                                                          EXF00035
LN 0036          IB=1                                                            EXF00036
LN 0037    C**** TRY TO FIND THE FILE NAME                                       EXF00037
LN 0038    48005 DO 48001 IA=ITOP,NUMFIL                                         EXF00038
LN 0039          IF(NFILE(IA,1).EQ.IX) GOTO 48010                                EXF00039
LN 0040          IF(IB.EQ.1) GOTO 48001                                          EXF00040
LN 0041          IF(NFILE(IA,1).EQ.-1) GOTO 48003                                EXF00041
LN 0042    48001 CONTINUE                                                        EXF00042
LN 0043          IF(IB.EQ.2) GOTO 48007                                          EXF00043
LN 0044          IB=IB+1                                                         EXF00044
LN 0045          ITOP=1                                                          EXF00045
LN 0046          GOTO 48005                                                      EXF00046
LN 0047    C**** ALLOCATE NEW FILE                                               EXF00047
LN 0048    48007 NUMFIL=NUMFIL+1                                                 EXF00048
LN 0049          IF(NUMFIL.GT.MAXFIL) GOTO 48012                                 EXF00049
LN 0050          IA=NUMFIL                                                       EXF00050
LN 0051          NFILE(IA,3)=IPROG(INREG-2)                                      EXF00051
LN 0052          NFILE(IA,1)=IX                                                  EXF00052
LN 0053          IF((IOP.EQ.2).OR.(IOP.EQ.13)) NFILE(IA,3)=1                     EXF00053
LN 0054          IF(IA.GT.1) GOTO 48008                                          EXF00054
LN 0055          NFILE(IA,2)=1000                                                EXF00055
LN 0056          GOTO 48002                                                      EXF00056
LN 0057    48008 NFILE(IA,2)=NFILE(IA-1,2)+NFILE(IA-1,3)                         EXF00057
LN 0058          IF(NFILE(IA,2).LE.MAXSAT) GOTO 48002                            EXF00058
LN 0059          NERROR=21                                                       EXF00059
LN 0060          GOTO 48013                                                      EXF00060
LN 0061    48002 MIC=NFILE(IA,3)                                                 EXF00061
```

```
LN 0062           DO 48004 LOCU=1,MTC                                    EXF00062
LN 0063           LOCS=NFILE(IA,2)+LOCU-1                                EXF00063
LN 0064           REWIND LOCS                                            EXF00064
LN 0065           IF(NWORT.LE.0) GOTO 48004                              EXF00065
LN 0066           DO 48009 K=1,NWORT                                     EXF00066
LN 0067           WRITE(LOCS) BLANK                                      EXF00067
LN 0068     48009 CONTINUE                                               EXF00068
LN 0069           REWIND LOCS                                            EXF00069
LN 0070     48004 CONTINUE                                               EXF00070
LN 0071           IF(IOP.EQ.2) GOTO 51011                                EXF00071
LN 0072           IF(IOP.EQ.13) GOTO 60001                               EXF00072
LN 0073           IB=4                                                   EXF00073
LN 0074           GOTO 900                                               EXF00074
LN 0075     C**** FREE FILE NAME FOUND --TRY TO INSERT                   EXF00075
LN 0076     48003 IF(IPROG(INREG-2).GT.NFILE(IA,3)) GOTO 48006           EXF00076
LN 0077           IF((NFILE(IA,3).NE.1).AND.(IPROG(INREG-2).EQ.1)) GOTO 48006  EXF00077
LN 0078           NFILE(IA,1)=IX                                         EXF00078
LN 0079           GOTO 48002                                             EXF00079
LN 0080     48006 ITOP=IA+1                                              EXF00080
LN 0081           GOTO 48005                                             EXF00081
LN 0082     48010 NERROR=22                                              EXF00082
LN 0083     48013 CALL EXERR(NERROR,I1,I2,X1,X2)                         EXF00083
LN 0084           IOP=-1                                                 EXF00084
LN 0085           RETURN                                                 EXF00085
LN 0086     48012 NERROR=23                                              EXF00086
LN 0087           I1=MSATZ                                               EXF00087
LN 0088           I2=NFILE(IA,3)                                         EXF00088
LN 0089           GOTO 48013                                             EXF00089
LN 0090     C**** COMMAND IS PUT -- SEQUENTIAL                           EXF00090
LN 0091     49000 IX=IPROG(INREG-1)                                      EXF00091
LN 0092           MSATZ=1                                                EXF00092
LN 0093           CALL FINDFI(IX,IA)                                     EXF00093
LN 0094           IF(IA.GT.NUMFIL) GOTO 48007                            EXF00094
LN 0095     51011 IF(MSATZ.GT.NFILE(IA,3)) GOTO 51012                    EXF00095
LN 0096           IF(MSATZ.LE.0) GOTO 54011                              EXF00096
LN 0097           LOCS=NFILE(IA,2)+MSATZ-1                               EXF00097
LN 0098           IF(IPROG(INREG-2).EQ.-16) GOTO 49002                   EXF00098
LN 0099     C**** FILE NAME FOUND                                        EXF00099
LN 0100           IB=2                                                   EXF00100
LN 0101           WRITE(LOCS) ACC                                        EXF00101
LN 0102      900  INREG=INREG-IB                                         EXF00102
LN 0103           RETURN                                                 EXF00103
LN 0104     49002 IC=IPROG(INREG-3)                                      EXF00104
LN 0105           DO 49003 K=1,IC                                        EXF00105
LN 0106           LOCU=INREG-3-K                                         EXF00106
LN 0107           ACC=IPROG(LOCU)                                        EXF00107
LN 0108           WRITE(LOCS) ACC                                        EXF00108
LN 0109     49003 CONTINUE                                               EXF00109
LN 0110           IB=4+IC                                                EXF00110
LN 0111           GOTO 900                                               EXF00111
LN 0112     C**** COMMAND IS PUT -- INDEX-SEQUENTIAL                     EXF00112
LN 0113     50000 IX=IPROG(INREG-1)                                      EXF00113
LN 0114           CALL FINDFI(IX,IA)                                     EXF00114
LN 0115           IF(IA.GT.NUMFIL) GOTO 51010                            EXF00115
LN 0116           GOTO 51011                                             EXF00116
LN 0117     51010 NERROR=24                                              EXF00117
LN 0118           GOTO 48013                                             EXF00118
LN 0119     51012 NERROR=25                                              EXF00119
LN 0120           I1=MSATZ                                               EXF00120
LN 0121           I2=NFILE(IA,3)                                         EXF00121
LN 0122           GOTO 48013                                             EXF00122
```

```
LN 0123     C**** COMMAND IS GET -- SEQUENTIAL                                  EXF00123
LN 0124     51000 MSATZ=1                                                       EXF00124
LN 0125     C**** COMMAND IS GET -- INDEX-SEQUENTIAL                            EXF00125
LN 0126     52000 IX=IPROG(INREG-1)                                             EXF00126
LN 0127           CALL FINDFI(IX,IA)                                            EXF00127
LN 0128           IF(IA.GT.NUMFIL) GOTO 51100                                   EXF00128
LN 0129           IF(MSATZ.GT.NFILE(IA,3)) GOTO 51012                           EXF00129
LN 0130           IF(MSATZ.LE.0) GOTO 54011                                     EXF00130
LN 0131           LOCS=NFILE(IA,2)+MSATZ-1                                      EXF00131
LN 0132           READ(LOCS) ACC                                                EXF00132
LN 0133     C*-*-                                                               EXF00133
LN 0134     C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(LOCS)-       EXF00134
LN 0135     C*-*-      ----CHECK IT----        ----CHECK IT----                 EXF00135
LN 0136     C*-*-                                                               EXF00136
LN 0137           IF(IFEOF(LOCS).EQ.-1) GOTO 51111                              EXF00137
LN 0138           IB=2                                                          EXF00138
LN 0139           GOTO 900                                                      EXF00139
LN 0140     51100 NERROR=26                                                     EXF00140
LN 0141           GOTO 48013                                                    EXF00141
LN 0142     C**** COMMAND IS RESET -- SEQUENTIAL                                EXF00142
LN 0143     53000 IX=IPROG(INREG-1)                                             EXF00143
LN 0144           CALL FINDFI(IX,IA)                                            EXF00144
LN 0145           IF(IA.GT.NUMFIL) GOTO 53001                                   EXF00145
LN 0146     53002 MSATZ=NFILE(IA,3)                                             EXF00146
LN 0147           DO 53003 K=1,MSATZ                                            EXF00147
LN 0148           LOCS=NFILE(IA,2)+K-1                                          EXF00148
LN 0149           REWIND LOCS                                                   EXF00149
LN 0150     53003 CONTINUE                                                      EXF00150
LN 0151           IB=4                                                          EXF00151
LN 0152           GOTO 900                                                      EXF00152
LN 0153     C**** COMMAND IS RESET -- INDEX-SEQUENTIAL                          EXF00153
LN 0154     54000 IX=IPROG(INREG-1)                                             EXF00154
LN 0155           CALL FINDFI(IX,IA)                                            EXF00155
LN 0156           IF(IA.GT.NUMFIL) GOTO 53001                                   EXF00156
LN 0157           IF(IPROG(INREG-2).LE.1) GOTO 53002                            EXF00157
LN 0158           IF(MSATZ.GT.NFILE(IA,3)) GOTO 51012                           EXF00158
LN 0159           IF(MSATZ.LE.0) GOTO 54011                                     EXF00159
LN 0160           LOCS=NFILE(IA,2)+MSATZ-1                                      EXF00160
LN 0161           REWIND LOCS                                                   EXF00161
LN 0162           IF(IPROG(INREG-2).LE.2) GOTO 54001                            EXF00162
LN 0163     C**** RESET POINTER INTO POSITION MPOINT                            EXF00163
LN 0164           IF((MPOINT.EQ.1).OR.(MPOINT.EQ.0)) GOTO 54001                 EXF00164
LN 0165           MPOINT=MPOINT-1                                               EXF00165
LN 0166           IF(MPOINT.LT.0) GOTO 54010                                    EXF00166
LN 0167           DO 54002 K=1,MPOINT                                           EXF00167
LN 0168           READ(LOCS) ACC                                                EXF00168
LN 0169     C*-*-                                                               EXF00169
LN 0170     C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(LOCS)-       EXF00170
LN 0171     C*-*-      ----CHECK IT----        ----CHECK IT----                 EXF00171
LN 0172     C*-*-                                                               EXF00172
LN 0173           IF(IFEOF(LOCS).EQ.-1) GOTO 51111                              EXF00173
LN 0174     54002 CONTINUE                                                      EXF00174
LN 0175     54001 IB=4                                                          EXF00175
LN 0176           GOTO 900                                                      EXF00176
LN 0177     53001 NERROR=27                                                     EXF00177
LN 0178           GOTO 48013                                                    EXF00178
LN 0179     51111 NERROR=28                                                     EXF00179
LN 0180           GOTO 48013                                                    EXF00180
LN 0181     54010 NERROR=46                                                     EXF00181
LN 0182           I2=MSATZ                                                      EXF00182
LN 0183           I1=MPOINT                                                     EXF00183
```

```
LN 0184          GOTO 48013                                                    EXF00184
LN 0185    54011 NERROR=47                                                     EXF00185
LN 0186          I1=MSATZ                                                      EXF00186
LN 0187          GOTO 48013                                                    EXF00187
LN 0188    C**** COMMAND IS CLOSE                                              EXF00188
LN 0189    55000 IX=IPROG(INREG-1)                                             EXF00189
LN 0190          CALL FINDFI(IX,IA)                                            EXF00190
LN 0191          IF(IA.GT.NUMFIL) GOTO 55001                                   EXF00191
LN 0192          NFILE(IA,1)=-1                                                EXF00192
LN 0193          IB=4                                                          EXF00193
LN 0194          GOTO 900                                                      EXF00194
LN 0195    55001 NERROR=29                                                     EXF00195
LN 0196          GOTO 48013                                                    EXF00196
LN 0197    C**** POINTER POSITION IN MPOINT                                    EXF00197
LN 0198    56000 MPOINT=ACC                                                    EXF00198
LN 0199          IB=1                                                          EXF00199
LN 0200          GOTO 900                                                      EXF00200
LN 0201    C**** GET SENTENCE NUMBER INTO MSATZ                                EXF00201
LN 0202    57000 MSATZ=ACC                                                     EXF00202
LN 0203          IB=1                                                          EXF00203
LN 0204          GOTO 900                                                      EXF00204
LN 0205    C**** COMMAND IS MAT GET                                            EXF00205
LN 0206    58000 MSATZ=1                                                       EXF00206
LN 0207    59000 IX=IPROG(INREG-1)                                             EXF00207
LN 0208          CALL FINDFI(IX,IA)                                            EXF00208
LN 0209          IF(IA.GT.NUMFIL) GOTO 51100                                   EXF00209
LN 0210    60001 IB=IPROG(INREG-2)                                             EXF00210
LN 0211          IZ=DATA(IB+27)+0.1                                            EXF00211
LN 0212          MPOINT=NFILE(IA,2)+MSATZ-1                                    EXF00212
LN 0213          IF(MSATZ.GT.NFILE(IA,3)) GOTO 51012                           EXF00213
LN 0214          IF(MSATZ.LE.0) GOTO 54011                                     EXF00214
LN 0215          LOCS=DATA(IB)+0.1                                             EXF00215
LN 0216          LOCU=DATA(LOCS)+0.1                                           EXF00216
LN 0217          IF(MERKER(IB,2).EQ.0) GOTO 58005                              EXF00217
LN 0218          IS=(LOCU-LOCS-1)/(IZ+1)-1                                     EXF00218
LN 0219          IBPLUS=1                                                      EXF00219
LN 0220          ISS=IZ                                                        EXF00220
LN 0221          GOTO 58004                                                    EXF00221
LN 0222    58005 IZ=LOCU-LOCS-2                                                EXF00222
LN 0223          ISS=1                                                         EXF00223
LN 0224          IS=1                                                          EXF00224
LN 0225          IBPLUS=-1                                                     EXF00225
LN 0226    58004 DO 58001 IN=1,IS                                              EXF00226
LN 0227          DO 58002 IM=1,IZ                                              EXF00227
LN 0228          JA=INT(DATA(IB))+IBPLUS*(ISS+1)*IN+IM                         EXF00228
LN 0229          IF(IOP.GE.13) WRITE(MPOINT) DATA(JA)                          EXF00229
LN 0230          IF(IOP.LT.13) READ(MPOINT) DATA(JA)                           EXF00230
LN 0231    C*-*-                                                               EXF00231
LN 0232    C*-*- CALL FOR A POSIBLE NON-COMPATIBLE ROUTINE -IFEOF(MPOINT)-     EXF00232
LN 0233    C*-*-     ----CHECK IT----        ----CHECK IT----                  EXF00233
LN 0234    C*-*-                                                               EXF00234
LN 0235          IF(IFFOF(MPOINT).EQ.-1) GOTO 51111                            EXF00235
LN 0236    58002 CONTINUE                                                      EXF00236
LN 0237    58001 CONTINUE                                                      EXF00237
LN 0238          IB=3                                                          EXF00238
LN 0239          GOTO 900                                                      EXF00239
LN 0240    C**** COMMAND IS MAT PUT -- SEQUENTIAL                              EXF00240
LN 0241    60000 GOTO 58000                                                    EXF00241
LN 0242    C**** COMMAND IS MAT PUT -- INDEX-SEQUENTIAL                        EXF00242
LN 0243    61000 GOTO 59000                                                    EXF00243
LN 0244    C**** WRONG CODE NUMBER                                             EXF00244
```

```
LN 0245    62000 NERROR=12                                                    EXF00245
LN 0246          CALL EXERR(NERROR,INREG,IPROG(INREG),X1,X2)                  EXF00246
LN 0247          IOP=-1                                                       EXF00247
LN 0248          RETURN                                                       EXF00248
LN 0249          END                                                          EXF00249

      USASI FORTRAN DIAGNOSTIC RESULTS FOR ZEXFIL


          NO ERRORS
```

ZEXMAT

```
LN 0001          SUBROUTINE ZEXMAT(IOP)                                          EXM00001
LN 0002    C**** SUBROUTINE TO EXECUTE MAT COMMANDS                              EXM00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    EXM00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST, INEXT,EXM00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  EXM00005
LN 0006         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, EXM00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     EXM00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    EXM00008
LN 0009         1NIFMAX,INTZEI                                                   EXM00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                      EXM00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),    EXM00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                     EXM00012
LN 0013          COMMON// ISTLST(340),LISTST(340)                                EXM00013
LN 0014          COMMON// DATAN(330)                                             ***
LN 0015          COMMON// DATA(3700)                                             EXM00015
LN 0016          COMMON/A/ DINV(12,12)                                           ***
LN 0017          DIMENSION IPROG(3700)                                           EXM00017
LN 0018          EQUIVALENCE (DATA(1),IPROG(1))                                  EXM00018
LN 0019          MOP=0                                                           EXM00019
LN 0020          MIC=0                                                           EXM00020
LN 0021          IF(IOP.EQ.49) GOTO 49000                                        EXM00021
LN 0022          IF(IOP.GT.25) GOTO 10                                           EXM00022
LN 0023          GOTO(23000,24000,25000,26000,10,28000,29000,10,31000,32000,33000, EXM00023
LN 0024         134000,35000,36000,37000,38000,39000,40000,41000,42000,43000,10,10 EXM00024
LN 0025         2,46000,47000),IOP                                               EXM00025
LN 0026     10   NERROR=11                                                       EXM00026
LN 0027          GOTO 29008                                                      EXM00027
LN 0028    C***  COMMAND IS MAT A=B+C                                            EXM00028
LN 0029    23000 MOP=23                                                          EXM00029
LN 0030          VALUE=1                                                         EXM00030
LN 0031          GOTO 24001                                                      EXM00031
LN 0032    C**** COMMAND IS MAT A=B-C                                            EXM00032
LN 0033    25000  MOP=25                                                        EXM00033
LN 0034          VALUE=-1                                                        EXM00034
LN 0035          GOTO 24001                                                      EXM00035
LN 0036    C**** COMMAND IS MATRIX MULTIPLICATION                                EXM00036
LN 0037    24000 MOP=24                                                          EXM00037
LN 0038    24001 IA=IPROG(INREG-1)                                               EXM00038
LN 0039          IB=IPROG(INREG-2)                                               EXM00039
LN 0040          IC=IPROG(INREG-3)                                               EXM00040
LN 0041          LOCS=DATA(IB)+0.1                                               EXM00041
LN 0042          LOCU=DATA(LOCS)+0.1                                             EXM00042
LN 0043          JB=DATA(IB+27)-1.                                               EXM00043
LN 0044          IF(MERKER(IB,2).EQ.0) GOTO 24110                                EXM00044
LN 0045          IZ=(LOCU-LOCS-1)/(JB+1)-1                                       EXM00045
LN 0046          GOTO 24010                                                      EXM00046
LN 0047    24110 IZ=1                                                            EXM00047
LN 0048          JB=LOCU-LOCS-2                                                  EXM00048
LN 0049    24010 IF(IA.GT.26) GOTO 24017                                         EXM00049
LN 0050          IF(MERKER(IA,1).NE.0) GOTO 24015                                EXM00050
LN 0051    24017 JA=1                                                            EXM00051
LN 0052          GOTO 24016                                                      EXM00052
LN 0053    24015 LOCS=DATA(IA)+0.1                                               EXM00053
LN 0054          LOCU=DATA(LOCS)+0.1                                             EXM00054
LN 0055          JA=DATA(IA+27)-1.                                               EXM00055
LN 0056          IF(MERKER(IA,2).EQ.0) GOTO 24111                                EXM00056
LN 0057          MIV=(LOCU-LOCS-1)/(JA+1)-1                                      EXM00057
LN 0058          GOTO 24011                                                      EXM00058
LN 0059    24111 JA=LOCU-LOCS-2                                                  EXM00059
LN 0060    24016 MIV=1                                                           EXM00060
LN 0061    24011 IS= DATA(IC+27)-1.                                              EXM00061
```

```
LN 0062          LOCS=DATA(IC)+0.1                                          EXM00062
LN 0063          LOCU=DATA(LOCS)+0.1                                        EXM00063
LN 0064          JC=IS                                                      EXM00064
LN 0065          IF(MERKER(IC,2).EQ.0) GOTO 24211                           EXM00065
LN 0066          ISS=(LOCU-LOCS-1)/(JC+1)-1                                 EXM00066
LN 0067          GOTO 170                                                   EXM00067
LN 0068    24211 JC=LOCU-LOCS-2                                             EXM00068
LN 0069          IS=JC                                                      EXM00069
LN 0070          ISS=1                                                      EXM00070
LN 0071     170  IAPLUS=1                                                   EXM00071
LN 0072          IBPLUS=1                                                   EXM00072
LN 0073          ICPLUS=1                                                   EXM00073
LN 0074          IF(MERKER(IA,2).EQ.0) IAPLUS=-JA                           EXM00074
LN 0075          IF(MERKER(IB,2).EQ.0) IBPLUS=-JB                           EXM00075
LN 0076          IF(MERKER(IC,2).EQ.0) ICPLUS=-JC                           EXM00076
LN 0077          IF(MOP.EQ.24) GOTO 24315                                   EXM00077
LN 0078          IF((JA.NE.JB).OR.(JA.NE.JC)) GOTO 43300                    EXM00078
LN 0079          IF((MIV.NE.IZ).OR.(MIV.NE.ISS)) GOTO 43300                 EXM00079
LN 0080          GOTO 24325                                                 EXM00080
LN 0081    24315 IF(JB.NE.ISS) GOTO 24260                                   EXM00081
LN 0082          IF(JA.NE.JC) GOTO 24260                                    EXM00082
LN 0083          IF(MIV.NE.IZ) GOTO 24260                                   EXM00083
LN 0084    24325 LOCS=JA+1                                                  EXM00084
LN 0085          LOCU=JB+1                                                  EXM00085
LN 0086          MIV=JC+1                                                   EXM00086
LN 0087          IF((MOP.EQ.24).AND.(IA.EQ.IC)) GOTO 24800                  EXM00087
LN 0088          DO 24100 IN=1,IZ                                           EXM00088
LN 0089          DO 24200 IM=1,IS                                           EXM00089
LN 0090          MIC=IN                                                     EXM00090
LN 0091          IF(IA.NE.IB) GOTO 24317                                    EXM00091
LN 0092          IF(MOP.NE.24) GOTO 24317                                   ***
LN 0093          IF(MERKER(IA,2).EQ.0) GOTO 24347                           ***
LN 0094          MIC=MIC-1                                                  ***
LN 0095    24317 JA=INT(DATA(IA))+IAPLUS+LOCS*MIC+IM                        EXM00093
LN 0096          IF(MOP.EQ.24) GOTO 24305                                   EXM00094
LN 0097          JB=INT(DATA(IB))+IBPLUS+LOCU*IN+IM                         EXM00095
LN 0098          JC=INT(DATA(IC))+ICPLUS+MIV*IN+IM                          EXM00096
LN 0099          DATA(JA)=DATA(JB)+VALUE*DATA(JC)                           EXM00097
LN 0100          GOTO 24200                                                 EXM00098
LN 0101    24305 IF(IA.GT.26) GOTO 24307                                    EXM00099
LN 0102          IF(MERKER(IA,1).NE.0) GOTO 24308                           ***
LN 0103          JA=DATA(IA)+1.5                                            EXM00101
LN 0104          GOTO 24308                                                 EXM00102
LN 0105    24307 JA=IA                                                      EXM00103
LN 0106    24308 ACC=0.                                                     EXM00106
LN 0107          DO 24300 IO=1,ISS                                          EXM00107
LN 0108          JB=INT(DATA(IB))+IBPLUS+LOCU*IN+IO                         EXM00108
LN 0109          JC=INT(DATA(IC))+ICPLUS+MIV*IO+IM                          EXM00109
LN 0110          ACC=ACC+DATA(JB)*DATA(JC)                                  EXM00110
LN 0111    24300 CONTINUE                                                   EXM00111
LN 0112          DATA(JA)=ACC                                               EXM00112
LN 0113    24200 CONTINUE                                                   EXM00113
LN 0114    24100 CONTINUE                                                   EXM00114
LN 0115          INREG=INREG-4                                              EXM00115
LN 0116          IF(MOP.NE.24) GOTO 900                                     EXM00116
LN 0117          IF(IA.NE.IB) GOTO 900                                      EXM00117
LN 0118          IF(IA.GT.26) GOTO 900                                      EXM00118
LN 0119          IF(MERKER(IA,1).EQ.0) GOTO 900                             EXM00119
LN 0120          DO 24410 IN=1,IZ                                           EXM00120
LN 0121          DO 24420 IM=1,IS                                           EXM00121
LN 0122          MIC=IZ-IN+1                                                EXM00122
```

```
LN 0123          JA=INT(DATA(IA))+IAPLUS+LOCS*MIC+IM                                EXM00123
LN 0124          JB=INT(DATA(IA))+IAPLUS+LOCS*(MIC-1)+IM                            EXM00124
LN 0125          DATA(JA)=DATA(JB)                                                  EXM00126
LN 0126    24420 CONTINUE                                                           EXM00127
LN 0127    24410 CONTINUE                                                           EXM00128
LN 0128          GOTO 900                                                           EXM00129
LN 0129    24800 IF(MERKER(IA,2).EQ.0) GOTO 24260                                   EXM00130
LN 0130          DO 24801 IM=1,IS                                                   EXM00131
LN 0131          DO 24802 IN=1,IZ                                                   EXM00132
LN 0132          JA=INT(DATA(IA))+IAPLUS+LOCS*IN+IM-1                               EXM00133
LN 0133          ACC=0.                                                             ***
LN 0134          DO 24803 IO=1,ISS                                                  EXM00134
LN 0135          JB=INT(DATA(IB))+IBPLUS+LOCU*IN+IO                                 EXM00135
LN 0136          JC=INT(DATA(IC))+ICPLUS+MIV*IO+IM                                  EXM00136
LN 0137          ACC=ACC+DATA(JB)*DATA(JC)                                          EXM00137
LN 0138    24803 CONTINUE                                                           EXM00138
LN 0139          DATA(JA)=ACC                                                       EXM00139
LN 0140    24802 CONTINUE                                                           EXM00140
LN 0141    24801 CONTINUE                                                           EXM00141
LN 0142          DO 24804 IM=1,IS                                                   EXM00142
LN 0143          DO 24805 IN=1,IZ                                                   EXM00143
LN 0144          MIV=IS-IM+1                                                        EXM00144
LN 0145          JA=INT(DATA(IA))+IAPLUS+LOCS*IN+MIV                                ***
LN 0146          JB=JA-1                                                            EXM00146
LN 0147          DATA(JA)=DATA(JB)                                                  EXM00147
LN 0148    24805 CONTINUE                                                           EXM00148
LN 0149    24804 CONTINUE                                                           EXM00149
LN 0150          INREG=INREG-4                                                      EXM00150
LN 0151          GOTO 900                                                           EXM00151
LN 0152    24260 NX=3                                                               EXM00152
LN 0153          CALL PRILIN(NX)                                                    EXM00153
LN 0154          WRITE(IWC,24290) ALPH(IA),MIV,JA,ALPH(IB),IZ,JB,ALPH(IC),ISS,JC    EXM00154
LN 0155    24290 FORMAT(1H0,79HPROGRAM STOPPED -- SIZE EPROR IN SUBSCRIPTS OF THE VEXM00155
LN 0156         1ARIABLES *  ROWS  * COLUMNS/57X,A1,2I10/57X,A1,2I10/57X,A1,2I10)   EXM00156
LN 0157          IF(MOP.NE.24) GOTO 29008                                           EXM00157
LN 0158          NERROR=30                                                          EXM00158
LN 0159          GOTO 29008                                                         EXM00159
LN 0160    24347 NERROR=51                                                          ***
LN 0161          I1=IA                                                              ***
LN 0162          I2=IC                                                              ***
LN 0163          GOTO 29008                                                         ***
LN 0164    C**** TRANSPONATION AF A MATRIX                                          EXM00160
LN 0165    26000 IA=IPROG(INREG-1)                                                  EXM00161
LN 0166          IB=IPROG(INREG-2)                                                  EXM00162
LN 0167          IZ=DATA(IA+27)-1.                                                  EXM00163
LN 0168          IAPLUS=1                                                           EXM00164
LN 0169          LOCS=DATA(IA)+0.1                                                  EXM00165
LN 0170          LOCU=DATA(LOCS)+0.1                                                EXM00166
LN 0171          IF(MERKER(IA,2).EQ.0) GOTO 26111                                   EXM00167
LN 0172          IS=(LOCU-LOCS-1)/(IZ+1)-1                                          EXM00168
LN 0173          MIC=IZ                                                             EXM00169
LN 0174          GOTO 26011                                                         EXM00170
LN 0175    26111 IS=1                                                               EXM00171
LN 0176          MIC=1                                                              EXM00172
LN 0177          IAPLUS=-1                                                          EXM00173
LN 0178          IZ=LOCU-LOCS-2                                                     EXM00174
LN 0179    26011 ISS=DATA(IB+27)-1.                                                 EXM00175
LN 0180          IBPLUS=1                                                           EXM00176
LN 0181          LOCS=DATA(IB)+0.1                                                  EXM00177
LN 0182          LOCU=DATA(LOCS)+0.1                                                EXM00178
LN 0183          IF(MERKER(IB,2).EQ.0) GOTO 26112                                   EXM00179
```

```
LN 0184           IC=(LOCU-LOCS-1)/(ISS+1)-1                                  EXM00180
LN 0185           MIV=ISS                                                     EXM00181
LN 0186           GOTO 26012                                                  EXM00182
LN 0187     26112 IBPLUS=-1                                                   EXM00183
LN 0188           IC=1                                                        EXM00184
LN 0189           MIV=1                                                       EXM00185
LN 0190           ISS=LOCU-LOCS-2                                             EXM00186
LN 0191     26012 IF(IZ.NE.IC) GOTO 26003                                     EXM00187
LN 0192           IF(IS.NE.ISS) GOTO 26003                                    EXM00188
LN 0193           DO 26002 J=1,IS                                             EXM00189
LN 0194           DO 26001 I=1,IZ                                             EXM00190
LN 0195           JA=INT(DATA(IA))+I+J*(MIC+1)+IAPLUS                         EXM00191
LN 0196           JB=INT(DATA(IB))+J+I*(MIV+1)+IBPLUS                         EXM00192
LN 0197           IF(IA.NE.IB) GOTO 26005                                     EXM00193
LN 0198           IF(I.LE.J) GOTO 26001                                       EXM00194
LN 0199           ACC=DATA(JA)                                                EXM00195
LN 0200           DATA(JA)=DATA(JB)                                           EXM00196
LN 0201           DATA(JB)=ACC                                                EXM00197
LN 0202           GOTO 26001                                                  EXM00198
LN 0203     26005 DATA(JA)=DATA(JB)                                           EXM00199
LN 0204     26001 CONTINUE                                                    EXM00200
LN 0205     26002 CONTINUE                                                    EXM00201
LN 0206           INREG=INREG-3                                               EXM00202
LN 0207           GOTO 900                                                    EXM00203
LN 0208     C                                                                 EXM00204
LN 0209     26003 WRITE(IWC,24290)ALPH(IA),IS,IZ,ALPH(IB),IC,ISS              EXM00205
LN 0210           NERROR=31                                                   EXM00206
LN 0211           GOTO 29008                                                  EXM00207
LN 0212     C                                                                 EXM00208
LN 0213     C**** MAT READ NO REDIMENSION                                     EXM00209
LN 0214     28000 NSW=1                                                       EXM00210
LN 0215     28100 IA=IPROG(INREG-1)                                           EXM00211
LN 0216     29003 IZ=DATA(IA+27)-1.                                           EXM00212
LN 0217           LOCS=DATA(IA)+0.1                                           EXM00213
LN 0218           LOCU=DATA(LOCS)+0.1                                         EXM00214
LN 0219           ISS=IZ                                                      EXM00215
LN 0220           IAPLUS=1                                                    EXM00216
LN 0221           IF(MERKER(IA,2).EQ.0) GOTO 28104                            EXM00217
LN 0222           IS=(LOCU-LOCS-1)/(IZ+1)-1                                   EXM00218
LN 0223           GOTO 28004                                                  EXM00219
LN 0224     28104 IZ=LOCU-LOCS-2                                              EXM00220
LN 0225     28006 ISS=1                                                       EXM00221
LN 0226           IS=1                                                        EXM00222
LN 0227           IAPLUS=-1                                                   EXM00223
LN 0228     28004 DO 28002 IN=1,IS                                            EXM00224
LN 0229           DO 28003 IM=1,IZ                                            EXM00225
LN 0230           IF((MOP.LE.-35).AND.(MOP.GE.-41)) GOTO 28005                EXM00226
LN 0231           IFR=1                                                       EXM00227
LN 0232           CALL ZHOPPR(VALUE,NSTOP,IFR,NSW)                            EXM00228
LN 0233           IF(NSTOP.NE.0) GOTO 22400                                   EXM00229
LN 0234     28005 JA=INT(DATA(IA))+IAPLUS+(ISS+1)*IN+IM                       EXM00230
LN 0235           DATA(JA)=VALUE                                              EXM00231
LN 0236     28003 CONTINUE                                                    EXM00232
LN 0237     28002 CONTINUE                                                    EXM00233
LN 0238           INREG=INREG-2                                               EXM00234
LN 0239           IF(MOP.EQ.-35) GOTO 35010                                   EXM00235
LN 0240           GOTO 900                                                    EXM00236
LN 0241     C**** REDIMENSION                                                 ***
LN 0242     29100 IA=IPROG(INREG-1)                                           EXM00238
LN 0243           LOCS=DATA(IA)+0.1                                           EXM00239
LN 0244           IF(XXX(3).LT.1.) GOTO 29005                                 ***
```

```
LN 0245          IF(MERKER(IA,2).EQ.0) GOTO 29001                        EXM00240
LN 0246          IF(XXX(4).LT.1.) GOTO 29004                             EXM00241
LN 0247          DATA(IA+27)=XXX(4)+1.01                                 EXM00243
LN 0248          DATA(LOCS)=(INT(XXX(3))+1)*(INT(XXX(4))+1)+LOCS+1       EXM00244
LN 0249          GOTO 29002                                              EXM00245
LN 0250    29001 DATA(LOCS)=INT(XXX(3))+LOCS+2                           ***
LN 0251    29002 NSIZE=(MERKER(IA,1)+1)*(MERKER(IA,2)+1)+1               EXM00248
LN 0252          LOCU=INT(DATA(LOCS))-LOCS                               EXM00249
LN 0253          IF(LOCU.GT.NSIZE) GOTO 29010                            EXM00250
LN 0254          XXX(3)=0.                                               EXM00251
LN 0255          XXX(4)=0.                                               EXM00252
LN 0256          GOTO 29003                                              EXM00253
LN 0257    29004 NERROR=32                                               EXM00254
LN 0258          X2=XXX(4)                                               EXM00255
LN 0259          GOTO 29009                                              EXM00256
LN 0260    29005 NERROR=33                                               EXM00257
LN 0261          IF(MERKER(IA,2).EQ.0) NERROR=32                         EXM00258
LN 0262          X2=XXX(3)                                               EXM00259
LN 0263          GOTO 29009                                              EXM00260
LN 0264    29010 X2=(MERKER(IA,1)+1)*(MERKER(IA,2)+1)                    EXM00261
LN 0265          IF(MERKER(IA,2).NE.0) GOTO 29110                        EXM00262
LN 0266          I1=1                                                    EXM00263
LN 0267          I2=XXX(3)                                               EXM00264
LN 0268          X2=MERKER(IA,1)+1                                       EXM00265
LN 0269          GOTO 29210                                              EXM00266
LN 0270    29110 I1=XXX(3)                                               EXM00267
LN 0271          I2=XXX(4)                                               EXM00268
LN 0272    29210 NERROR=34                                               EXM00269
LN 0273    29009 X1=ALPH(IA)                                             EXM00270
LN 0274          GOTO 29008                                              EXM00271
LN 0275    C*** MAT READ WITH ACTUAL SIZE                                EXM00272
LN 0276    29000 NSW=1                                                   EXM00273
LN 0277    29150 IA=IPROG(INREG-1)                                       EXM00274
LN 0278          LOCS=DATA(IA)+0.1                                       EXM00275
LN 0279          LOCU=DATA(LOCS)+0.1                                     EXM00276
LN 0280          IF(XXX(3).LT.1.) GOTO 29005                             ***
LN 0281          IF(MERKER(IA,2).EQ.0) GOTO 29200                        EXM00277
LN 0282          IF(XXX(4).LT.1.) GOTO 29004                             EXM00278
LN 0283          IZ=DATA(IA+27)-1.                                       EXM00280
LN 0284          IS=(LOCU-LOCS-1)/(IZ+1)-1                               EXM00281
LN 0285          IF(XXX(4).GT.FLOAT(IZ)) GOTO 29100                      EXM00282
LN 0286          IF(XXX(3).GT.FLOAT(IS)) GOTO 29100                      EXM00283
LN 0287          IS=XXX(3)                                               EXM00284
LN 0288          IAPLUS=1                                                EXM00285
LN 0289          ISS=IZ                                                  EXM00286
LN 0290          IZ=XXX(4)                                               EXM00287
LN 0291          GOTO 28004                                              EXM00288
LN 0292    29200 IZ=XXX(3)                                               ***
LN 0293          IF(IZ.GT.(LOCU-LOCS-2)) GOTO 29100                      EXM00291
LN 0294          GOTO 28006                                              EXM00292
LN 0295    C                                                             EXM00293
LN 0296    C**** PRINT WITHOUT REDIMENSION  -- NUMERIC                   EXM00294
LN 0297    31000 IA=IPROG(INREG-1)                                       EXM00295
LN 0298          NX=2                                                    EXM00296
LN 0299          CALL PRILIN(NX)                                         EXM00297
LN 0300          IS=DATA(IA+27)-1.                                       EXM00298
LN 0301          LOCS=DATA(IA)+0.1                                       EXM00299
LN 0302          LOCU=DATA(LOCS)+0.1                                     EXM00300
LN 0303          IF(MERKER(IA,2).EQ.0) GOTO 31101                        EXM00301
LN 0304          IZ=(LOCU-LOCS-1)/(IS+1)-1                               EXM00302
LN 0305          ISS=IS                                                  EXM00303
```

```
LN 0306          IAPLUS=1                                                    EXM00304
LN 0307          IF(MIC.NE.32) GOTO 31004                                    EXM00305
LN 0308          IF(XXX(3).LT.1.) GOTO 29005                                 ***
LN 0309          IF(XXX(4).LT.1.) GOTO 29004                                 ***
LN 0310          IF(XXX(4).GT.FLOAT(IS)) GOTO 29100                          ***
LN 0311          IF(XXX(3).GT.FLOAT(IZ)) GOTO 29100                          ***
LN 0312          IZ=XXX(3)                                                   EXM00306
LN 0313          IS=XXX(4)                                                   EXM00307
LN 0314          GOTO 31004                                                  EXM00308
LN 0315    31101 IS=LOCU-LOCS-2                                              EXM00309
LN 0316          ISS=1                                                       EXM00310
LN 0317          IZ=1                                                        EXM00311
LN 0318          IAPLUS=-1                                                   EXM00312
LN 0319          IF(MIC.NE.32) GOTO 31004                                    EXM00313
LN 0320          IF(XXX(3).LT.1.) GOTO 29005                                 ***
LN 0321          IF(XXX(3).GT.FLOAT(IS)) GOTO 29100                          ***
LN 0322          IS=XXX(3)                                                   EXM00314
LN 0323    31004 DO 31002 IN=1,IZ                                            EXM00315
LN 0324          IF(NSTZEI.EQ.3) GOTO 31027                                  EXM00316
LN 0325          IF(MERKER(IA,2).EQ.0) WRITE(IWC,31020) ALPH(IA)             EXM00317
LN 0326          IF(MERKER(IA,2).NE.0) WRITE(IWC,31007)ALPH(IA),IN           EXM00318
LN 0327    31020 FORMAT(5X,8HVEKTOR *,A1,1H*)                                EXM00319
LN 0328    31007 FORMAT(5X,8HMATRIX *,A1,9H* ZEILE *,I4,1H*)                 EXM00320
LN 0329    31027 DO 31003 IM=1,IS                                            EXM00321
LN 0330          JA=INT(DATA(IA))+IAPLUS+(ISS+1)*IN+IM                       EXM00322
LN 0331          IF(MOP.EQ.-33) GOTO 31008                                   EXM00323
LN 0332          ACC=DATA(JA)                                                EXM00324
LN 0333          INEXT=INEXT+IZONE                                           EXM00325
LN 0334          CALL ZINSNO                                                 EXM00326
LN 0335          IF(IM.EQ.IS) GOTO 31006                                     EXM00327
LN 0336          IF(INEXT.LT.IWRIT) GOTO 31003                               EXM00328
LN 0337          GOTO 31006                                                  EXM00329
LN 0338    31008 IX=DATA(JA)                                                 EXM00330
LN 0339          IPOS=INEXT                                                  EXM00331
LN 0340          DO 31015 K=1,5                                              EXM00332
LN 0341          JB=IX-((IX/INTZEI)*INTZEI)                                  EXM00333
LN 0342          CARP(IPOS)=ALPH(JB+1)                                       ***
LN 0343          IPOS=IPOS+1                                                 ***
LN 0344    31015 IX=IX/INTZEI                                                EXM00336
LN 0345          IF(NPRI.EQ.1) INEXT=INEXT+IZONE                             EXM00337
LN 0346          IF(NPRI.EQ.-1) INEXT=INEXT+5                                EXM00338
LN 0347          IF(INEXT.GE.IWRIT) GOTO 31006                               EXM00339
LN 0348          GOTO 31003                                                  EXM00340
LN 0349    31006 NX=1                                                        EXM00341
LN 0350          CALL PRILIN(NX)                                             EXM00342
LN 0351          INEXT=1                                                     EXM00343
LN 0352          NPRUS=1                                                     EXM00344
LN 0353          IPOS=0                                                      EXM00345
LN 0354    31003 CONTINUE                                                    EXM00346
LN 0355    31002 CONTINUE                                                    EXM00347
LN 0356          INREG=INREG-2                                               EXM00348
LN 0357          NPRI=1                                                      ***
LN 0358          IF((INEXT.EQ.1).AND.(NPRUS.EQ.1)) GOTO 900                  ***
LN 0359          NX=1                                                        EXM00349
LN 0360          CALL PRILIN(NX)                                             EXM00350
LN 0361          GOTO 900                                                    EXM00353
LN 0362    C**** PRINT WITH REDIMENSION -- NUMERIC                           EXM00354
LN 0363    32000 MOP=-32                                                     EXM00355
LN 0364          MIC=32                                                      EXM00356
LN 0365          GOTO 31000                                                  EXM00357
LN 0366    C**** PRINT WITHOUT REDIMENSION --ALPHANUMERIC                    EXM00358
```

```
LN 0367    33000 MOP=-33                                                    EXM00359
LN 0368          GOTO 31000                                                 EXM00360
LN 0369    C**** PRINT WITH REDIMENSION -- ALPHANUMERIC                     EXM00361
LN 0370    34000 MIC=32                                                     EXM00362
LN 0371          MOP=-33                                                    EXM00363
LN 0372          GOTO 31000                                                 EXM00364
LN 0373    C                                                                EXM00365
LN 0374    C**** IDENTITY MATRIX                                            EXM00366
LN 0375    C**** NO REDIMENSION                                             EXM00367
LN 0376    35000 IA=IPROG(INREG-1)                                          EXM00368
LN 0377          MOP=-35                                                    EXM00369
LN 0378          VALUE=0.                                                   EXM00370
LN 0379          GOTO 29003                                                 EXM00371
LN 0380    35010 IF(IS.EQ.IZ) GOTO 35012                                    EXM00372
LN 0381          NERROR=35                                                  EXM00373
LN 0382          X1=ALPH(IA)                                                EXM00374
LN 0383          I1=IS                                                      EXM00375
LN 0384          I2=IZ                                                      EXM00376
LN 0385          GOTO 29009                                                 EXM00377
LN 0386    35012 DO 35011 IN=1,IS                                           EXM00378
LN 0387          JA=INT(DATA(IA))+1+IS*IN+2*IN                              EXM00379
LN 0388          DATA(JA)=1.                                                EXM00380
LN 0389    35011 CONTINUE                                                   EXM00381
LN 0390          GOTO 900                                                   EXM00382
LN 0391    C**** IDENTITY MATRIX WITH REDIMENSION                           EXM00383
LN 0392    36000 MOP=-35                                                    EXM00384
LN 0393          VALUE=0.                                                   EXM00385
LN 0394          GOTO 29100                                                 EXM00386
LN 0395    C**** MATRIX=1 NO REDIMENSION                                    EXM00387
LN 0396    37000 IA=IPROG(INREG-1)                                          EXM00388
LN 0397          VALUE=1.                                                   EXM00389
LN 0398          MOP=-37                                                    EXM00390
LN 0399          GOTO 29003                                                 EXM00391
LN 0400    C**** MATRIX=1 WITH REDIMENSION                                  EXM00392
LN 0401    38000 MOP=-38                                                    EXM00393
LN 0402          VALUE=1.                                                   EXM00394
LN 0403          GOTO 29100                                                 EXM00395
LN 0404    C**** MATRIX=0 NO REDIMENSION                                    EXM00396
LN 0405    39000 IA=IPROG(INREG-1)                                          EXM00397
LN 0406          VALUE=0.                                                   EXM00398
LN 0407          MOP=-39                                                    EXM00399
LN 0408          GOTO 29003                                                 EXM00400
LN 0409    C**** MATRIX=0 WITH REDIMENSION                                  EXM00401
LN 0410    40000 VALUE=0.                                                   EXM00402
LN 0411          MOP=-40                                                    EXM00403
LN 0412          GOTO 29100                                                 EXM00404
LN 0413    C**** A=EXPRESSION -- MAT LET COMMAND                            EXM00405
LN 0414    41000 VALUE=ACC                                                  EXM00406
LN 0415          IA=IPROG(INREG-1)                                          EXM00407
LN 0416          MOP=-41                                                    EXM00408
LN 0417          GOTO 29003                                                 EXM00409
LN 0418    C**** A(I,J)=B(I,J) -- MAT LET COMMAND                           EXM00410
LN 0419    42000 IB=IPROG(INREG-1)                                          EXM00411
LN 0420          IA=IPROG(INREG-2)                                          EXM00412
LN 0421          MOP=4                                                      EXM00413
LN 0422          VALUE=1.                                                   EXM00414
LN 0423          INREG=INREG-3                                              EXM00415
LN 0424          GOTO 42500                                                 EXM00416
LN 0425    C**** A(I,J)=B(I,J) -BINARY OPERATOR)- (EXPRESSION) *MAT LET COMMAND EXM00417
LN 0426    43000 MOP=IPROG(INREG-1)                                         EXM00418
LN 0427          IOP=1                                                      EXM00419
```

```
LN 0428          IF(IPROG(INREG-2).NE.-43) GOTO 43001                    EXM00420
LN 0429          IOP=-1                                                  EXM00421
LN 0430          INREG=INREG-1                                           EXM00422
LN 0431     43001 IA=IPROG(INREG-2)                                      EXM00423
LN 0432          IB=IPROG(INREG-3)                                       EXM00424
LN 0433          INREG=INREG-4                                           EXM00425
LN 0434          VALUE=ACC                                               EXM00426
LN 0435     42500 IZ=DATA(IA+27)-1.                                      EXM00427
LN 0436          LOCS=DATA(IA)                                           EXM00428
LN 0437          LOCU=DATA(LOCS)                                         EXM00429
LN 0438          IAPLUS=1                                                EXM00430
LN 0439          ISS=IZ                                                  EXM00431
LN 0440          IF(MERKER(IA,2).EQ.0) GOTO 42501                        EXM00432
LN 0441          IS=(LOCU-LOCS-1)/(IZ+1)-1                               EXM00433
LN 0442          GOTO 42502                                              EXM00434
LN 0443     42501 IZ=LOCU-LOCS-2                                         EXM00435
LN 0444          IAPLUS=-1                                               EXM00436
LN 0445          ISS=1                                                   EXM00437
LN 0446          IS=ISS                                                  EXM00438
LN 0447     42502 IC=DATA(IB+27)-1.                                      EXM00439
LN 0448          IBPLUS=1                                                EXM00440
LN 0449          LOCS=DATA(IB)                                           EXM00441
LN 0450          LOCU=DATA(LOCS)                                         EXM00442
LN 0451          IF(MERKER(IB,2).EQ.0) GOTO 42503                        EXM00443
LN 0452          JC=(LOCU-LOCS-1)/(IC+1)-1                               EXM00444
LN 0453          IF(IZ.GT.IC) IZ=IC                                      EXM00445
LN 0454          IF(IS.GT.JC) IS=JC                                      EXM00446
LN 0455          GOTO 43200                                              EXM00447
LN 0456     42503 JC=LOCU-LOCS-2                                         EXM00448
LN 0457          IBPLUS=-1                                               EXM00449
LN 0458          IC=1                                                    EXM00450
LN 0459          IF(IZ.GT.JC) IZ=JC                                      EXM00451
LN 0460     43200 DO 43110 IN=1,IS                                       EXM00452
LN 0461          DO 43120 IM=1,IZ                                        EXM00453
LN 0462          JA=INT(DATA(IA))+IAPLUS+(ISS+1)*IN+IM                   EXM00454
LN 0463          JB=INT(DATA(IB))+IBPLUS+(IC+1)*IN+IM                    EXM00455
LN 0464          IF(MOP.NE.1) GOTO 43130                                 EXM00456
LN 0465          DATA(JA)=DATA(JB)+VALUE                                 EXM00457
LN 0466          GOTO 43120                                              EXM00458
LN 0467     43130 IF(MOP.NE.2) GOTO 43140                                EXM00459
LN 0468          IF(IOP.EQ.1) DATA(JA)=DATA(JB)-VALUE                    EXM00460
LN 0469          IF(IOP.EQ.-1) DATA(JA)=VALUE-DATA(JB)                   EXM00461
LN 0470          GOTO 43120                                              EXM00462
LN 0471     43140 IF(MOP.NE.3) GOTO 43150                                EXM00463
LN 0472          IF(IOP.EQ.-1) GOTO 43143                                EXM00464
LN 0473          IF(VALUE.NE.0.) GOTO 43141                              EXM00465
LN 0474     43144 NERROR=42                                              EXM00466
LN 0475          GOTO 29008                                              EXM00467
LN 0476     43141 DATA(JA)=DATA(JB)/VALUE                                EXM00468
LN 0477          GOTO 43120                                              EXM00469
LN 0478     43143 IF(DATA(JB).EQ.0.) GOTO 43144                          EXM00470
LN 0479          DATA(JA)=VALUE/DATA(JB)                                 EXM00471
LN 0480          GOTO 43120                                              EXM00472
LN 0481     43150 IF(MOP.NE.4) GOTO 43160                                EXM00473
LN 0482          IF((DATA(JA).EQ.0.).OR.(DATA(JB).EQ.0.)) GOTO 43151     EXM00474
LN 0483          IF(ALOG10(ABS(DATA(JB)))+ALOG10(ABS(VALUE)).GT.FLOAT(IEXPO))
LN 0484         1 GOTO 35                                                EXM00476
LN 0485     43151 DATA(JA)=DATA(JB)*VALUE                                EXM00477
LN 0486          GOTO 43120                                              EXM00478
LN 0487     43160 IF(MOP.NE.5) GOTO 43170                                EXM00479
LN 0488          IF(IOP.EQ.-1) GOTO 43180                                EXM00480
```

```
LN 0489          ACC=DATA(JR)                                                    EXM00481
LN 0490          VAL=VALUE                                                       EXM00482
LN 0491          GOTO 43190                                                      EXM00483
LN 0492    43180 ACC=VALUE                                                       EXM00484
LN 0493          VAL=DATA(JR)                                                    EXM00485
LN 0494    43190 IF(ACC.EQ.0.) GOTO 58                                           EXM00486
LN 0495          IF(VAL.EQ.0.) GOTO 51                                           EXM00487
LN 0496          IF(ABS(ALOG10(ABS(ACC))*VAL).GT.FLOAT(IEXPO)) GOTO 57           EXM00488
LN 0497          IF(VAL.GT.0.) GOTO 56                                           EXM00489
LN 0498          VAL=-VAL                                                        EXM00490
LN 0499          ACC=1./ACC                                                      EXM00491
LN 0500     56   MIV=VAL                                                         EXM00492
LN 0501          XS=MIV                                                          EXM00493
LN 0502          IF(XS.EQ.VAL) GOTO 54                                           EXM00494
LN 0503          IF(ACC.GE.0.) GOTO 53                                           EXM00495
LN 0504          XS=ABS(ACC)**VAL                                                EXM00496
LN 0505          XV=-1.*XS**(1./VAL)                                             EXM00497
LN 0506          IF(XV.NE.ACC) GOTO 55                                           EXM00498
LN 0507          DATA(JA)=-XS                                                    EXM00499
LN 0508          GOTO 43120                                                      EXM00500
LN 0509     55   NERROR=43                                                       EXM00501
LN 0510          X1=ACC                                                          EXM00502
LN 0511          X2=VAL                                                          EXM00503
LN 0512          GOTO 29008                                                      EXM00504
LN 0513     57   NERROR=44                                                       EXM00505
LN 0514          GOTO 29008                                                      EXM00506
LN 0515     53   DATA(JA)=ACC**VAL                                               EXM00507
LN 0516          GOTO 43120                                                      EXM00508
LN 0517     54   DATA(JA)=ACC**MIV                                               EXM00509
LN 0518          GOTO 43120                                                      EXM00510
LN 0519     58   IF(VAL.EQ.0.) GOTO 51                                           EXM00511
LN 0520          DATA(JA)=0.                                                     EXM00512
LN 0521          GOTO 43120                                                      EXM00513
LN 0522     51   DATA(JA)=1.                                                     EXM00514
LN 0523    43120 CONTINUE                                                        EXM00515
LN 0524    43110 CONTINUE                                                        EXM00516
LN 0525          IOP=1                                                           EXM00517
LN 0526          GOTO 900                                                        EXM00518
LN 0527    43300 WRITE(IWC,24290) ALPH(IA),MIV,JA,ALPH(IB),IZ,JB,ALPH(IC),ISS,JC EXM00519
LN 0528          NERROR=36                                                       EXM00520
LN 0529          GOTO 29008                                                      EXM00521
LN 0530    C**** EXECUTION OF MAT INPUT                                          EXM00522
LN 0531    46000 NSW=3                                                           EXM00523
LN 0532          GOTO 28100                                                      EXM00524
LN 0533    C**** MAT INPUT WITH ACTUAL SIZE IN INDEX                             EXM00525
LN 0534    47000 NSW=3                                                           EXM00526
LN 0535          GOTO 29150                                                      EXM00527
LN 0536    29008 CALL EXERR(NERROR,I1,I2,X1,X2)                                  EXM00528
LN 0537          IOP=-1                                                          EXM00529
LN 0538     900  RETURN                                                         EXM00530
LN 0539    22400 NERROR=NSTOP+3                                                  EXM00531
LN 0540          I1=VALUE                                                        EXM00532
LN 0541          CALL EXERR(NERROR,I1,I2,X1,X2)                                  EXM00533
LN 0542          IOP=-1                                                          EXM00534
LN 0543          RETURN                                                          EXM00535
LN 0544    43170 NERROR=11                                                       EXM00536
LN 0545          GOTO 29008                                                      EXM00537
LN 0546     35   NERROR=45                                                       EXM00538
LN 0547          GOTO 29008                                                      EXM00539
LN 0548    C**** INVERSION                                                       EXM00540
LN 0549    49000 MOP=1                                                           EXM00541
```

```
LN 0550     C**** JB=SIZE LIMIT TO INVERT A MATRIX                          EXM00542
LN 0551     C**** DINV MUST BE DIMENSIONED TO EQUAL JB                      EXM00543
LN 0552           JB=12                                                     ***
LN 0553           IA=IPROG(INREG-2)                                         EXM00545
LN 0554     49009 IZ=DATA(IA+27)-1.                                         EXM00546
LN 0555           LOCS=DATA(IA)                                             EXM00547
LN 0556           LOCU=DATA(LOCS)                                           EXM00548
LN 0557           IS=(LOCU-LOCS-1)/(IZ+1)-1                                 EXM00549
LN 0558           IF(MOP.EQ.2) GOTO 49003                                   EXM00550
LN 0559           IR=IZ                                                     EXM00551
LN 0560           IC=IS                                                     EXM00552
LN 0561     49003 IF((IR.NE.IZ).OR.(IC.NE.IS)) GOTO 49010                   EXM00553
LN 0562           IF(IS.NE.IZ) GOTO 49007                                   EXM00554
LN 0563           IF(IS.GT.JB) GOTO 49017                                   EXM00555
LN 0564           DO 49001 IN=1,IS                                          EXM00556
LN 0565           DO 49002 IM=1,IZ                                          EXM00557
LN 0566           JA=INT(DATA(IA))+1+(IZ+1)*IN+IM                           EXM00558
LN 0567           IF(MOP.EQ.1) GOTO 49005                                   EXM00559
LN 0568           DATA(JA)=DINV(IN,IM)                                      EXM00560
LN 0569           GOTO 49002                                                EXM00561
LN 0570     49005 DINV(IN,IM)=DATA(JA)                                      EXM00562
LN 0571     49002 CONTINUE                                                  EXM00563
LN 0572     49001 CONTINUE                                                  EXM00564
LN 0573           IF(MOP.EQ.2) GOTO 49008                                   EXM00565
LN 0574           CALL SUBINV(IS,MOP)                                       EXM00566
LN 0575           IF(MOP.EQ.-1) GOTO 49011                                  EXM00567
LN 0576           MOP=2                                                     EXM00568
LN 0577           IA=IPROG(INREG-1)                                         EXM00569
LN 0578           GOTO 49009                                                EXM00570
LN 0579     49007 NERROR=35                                                 EXM00571
LN 0580           X1=ALPH(IA)                                               EXM00572
LN 0581           I1=IS                                                     EXM00573
LN 0582           I2=IZ                                                     EXM00574
LN 0583           GOTO 29008                                                EXM00575
LN 0584     49008 INREG=INREG-3                                             EXM00576
LN 0585           GOTO 900                                                  EXM00577
LN 0586     49010 NERROR=48                                                 EXM00578
LN 0587           GOTO 29008                                                EXM00579
LN 0588     49011 NERROR=49                                                 EXM00580
LN 0589           I1=IA                                                     EXM00581
LN 0590           GOTO 29008                                                EXM00582
LN 0591     49017 NERROR=50                                                 EXM00583
LN 0592           I1=10                                                     EXM00584
LN 0593           GOTO 29008                                                EXM00585
LN 0594           END                                                       EXM00586


      USASI FORTRAN DIAGNOSTIC RESULTS FOR ZEXMAT


           NO ERRORS


THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

  A
```

```
LN 0001          SUBROUTINE EXERR(NERROR,I1,I2,X1,X2)                                EXR00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,        EXR00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT,   EXR00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,QQUOTE,MAXFIL,   EXR00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,  EXR00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI      EXR00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,       EXR00007
LN 0008         1NIFMAX,INTZEI                                                      EXR00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                         EXR00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),       EXR00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                        EXR00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                   EXR00012
LN 0013          COMMON// DATAN(330)                                                ***
LN 0014          COMMON// DATA(3700)                                                EXR00014
LN 0015          DIMENSION IPROG(3700)                                              EXR00015
LN 0016          DIMENSION IERR(51)                                                 ***
LN 0017          EQUIVALENCE (DATA(1),IPROG(1))                                     EXR00017
LN 0018          DATA IERR(1)/8HA 332.46/                                           EXR00018
LN 0019          DATA IERR(2)/8HA 332.41/                                           EXR00019
LN 0020          DATA IERR(3)/8HA 332.46/                                           EXR00020
LN 0021          DATA IERR(4)/8HA 322.12/                                           EXR00021
LN 0022          DATA IERR(5)/8HA 332.11/                                           EXR00022
LN 0023          DATA IERR(6)/8HA  322.2/                                           EXR00023
LN 0024          DATA IERR(7)/8HA 332.15/                                           EXR00024
LN 0025          DATA IERR(8)/8HA 332.13/                                           EXR00025
LN 0026          DATA IERR(9)/8HC  324.3/                                           ***
LN 0027          DATA IERR(10)/8HA 332.43/                                          EXR00027
LN 0028          DATA IERR(11)/8HC  324.3/                                          ***
LN 0029          DATA IERR(12)/8HC  324.3/                                          ***
LN 0030          DATA IERR(13)/8HA 332.23/                                          EXR00030
LN 0031          DATA IERR(14)/8HA  324.1/                                          EXR00031
LN 0032          DATA IERR(15)/8HA  324.1/                                          EXR00032
LN 0033          DATA IERR(16)/8HA  324.1/                                          EXR00033
LN 0034          DATA IERR(17)/8HA 331.21/                                          EXR00034
LN 0035          DATA IERR(18)/8HA  324.1/                                          EXR00035
LN 0036          DATA IERR(19)/8HA  324.1/                                          EXR00036
LN 0037          DATA IERR(20)/8HA  324.1/                                          EXR00037
LN 0038          DATA IERR(21)/8HB 332.11/                                          EXR00038
LN 0039          DATA IERR(22)/8HB 332.11/                                          EXR00039
LN 0040          DATA IERR(23)/8HB 332.11/                                          EXR00040
LN 0041          DATA IERR(24)/8HB 332.21/                                          EXR00041
LN 0042          DATA IERR(25)/8HB 332.21/                                          EXR00042
LN 0043          DATA IERR(26)/8HB 332.32/                                          EXR00043
LN 0044          DATA IERR(27)/8HB 332.23/                                          EXR00044
LN 0045          DATA IERR(28)/8HB 332.32/                                          EXR00045
LN 0046          DATA IERR(29)/8HB 332.12/                                          EXR00046
LN 0047          DATA IERR(30)/8HB331.211/                                          EXR00047
LN 0048          DATA IERR(31)/8HB331.211/                                          EXR00048
LN 0049          DATA IERR(32)/8HB331.312/                                          EXR00049
LN 0050          DATA IERR(33)/8HB331.312/                                          EXR00050
LN 0051          DATA IERR(34)/8HB331.312/                                          EXR00051
LN 0052          DATA IERR(35)/8HB331.211/                                          EXR00052
LN 0053          DATA IERR(36)/8HB331.221/                                          EXR00053
LN 0054          DATA IERR(37)/8HA 332.23/                                          EXR00054
LN 0055          DATA IERR(38)/8HA  324.1/                                          EXR00055
LN 0056          DATA IERR(39)/8HA 332.22/                                          EXR00056
LN 0057          DATA IERR(40)/8HA  324.1/                                          EXR00057
LN 0058          DATA IERR(41)/8HA332.412/                                          EXR00058
LN 0059          DATA IERR(42)/8HB331.211/                                          EXR00059
LN 0060          DATA IERR(43)/8HB331.211/                                          EXR00060
LN 0061          DATA IERR(44)/8HB331.211/                                          EXR00061
```

```
LN 0062          DATA IERR(45)/8HB331.211/                                      EXR00062
LN 0063          DATA IERR(46)/8HB 332.23/                                      EXR00063
LN 0064          DATA IERR(47)/8HB 332.21/                                      EXR00064
LN 0065          DATA IERR(48)/8HB331.221/                                      EXR00065
LN 0066          DATA IERR(49)/8HB331.221/                                      EXR00066
LN 0067          DATA IERR(50)/8HB331.221/                                      EXR00067
LN 0068          DATA IERR(51)/8HB331.211/                                      ***
LN 0069          NN=1                                                           EXR00068
LN 0070          NX=3                                                           EXR00069
LN 0071          CALL PRILIN(NX)                                                EXR00070
LN 0072          WRITE(IWC,1000)                                                EXR00071
LN 0073     1000 FORMAT(1H0,19HPROGRAM STOPPED -- )                             EXR00072
LN 0074          IF(NN.EQ.0) GOTO 2000                                          EXR00073
LN 0075          GOTO(2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44EXR00074
LN 0076         1,46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,86EXR00075
LN 0077         2,90,92,94,62,98,100,103),NERROR                                ***
LN 0078     102  FORMAT(1H+,19X,17HI CAN ONLY HANDLE,I4,23H GOSUBS WITHOUT RETURNS)EXR00077
LN 0079     2    WRITE(IWC,102) NIRMAX                                          EXR00078
LN 0080          GOTO 2000                                                      EXR00079
LN 0081     4    WRITE(IWC,104) I1                                              EXR00080
LN 0082     104  FORMAT(1H+,19X,14HI CANNOT GO TO,I5,24H BECAUSE IT IS NOT THERE)  EXR00081
LN 0083          GOTO 2000                                                      EXR00082
LN 0084     6    WRITE(IWC,106)                                                 EXR00083
LN 0085     106  FORMAT(1H+,19X,49HI CANNOT RETURN SINCE I DID NOT COME FROM A GOSUEXR00084
LN 0086         1B)                                                             EXR00085
LN 0087          GOTO 2000                                                      EXR00086
LN 0088     8    WRITE(IWC,108) IEXPO,IEXPO                                     EXR00087
LN 0089     108  FORMAT(1H+,19X,38HI CAN HANDLE ONLY NUMBERS BETWEEN 10**,I3,11H ANEXR00088
LN 0090         1D 10**(-,I3,1H))                                               EXR00089
LN 0091          GOTO 1999                                                      EXR00090
LN 0092     10   WRITE(IWC,110)                                                 EXR00091
LN 0093     110  FORMAT(1H+,19X,23HWRONG CHARACTER IN DATA)                     EXR00092
LN 0094          GOTO 1999                                                      EXR00093
LN 0095     12   WRITE(IWC,112)                                                 EXR00094
LN 0096     112  FORMAT(1H+,19X,57HQUOTATION MARKS ARE NOT PAIRED IN THE FOLLOWING EXR00095
LN 0097         1DATA-CARD)                                                     EXR00096
LN 0098          GOTO 1999                                                      EXR00097
LN 0099     14   WRITE(IWC,114) I1,NCFLLD                                       EXR00098
LN 0100     114  FORMAT(1H+,19X,39HYOU SET DATA POINTER ON POSITION NUMBER,I6,27H BEXR00099
LN 0101         1UT I ONLY CAN DO IT UNTIL,I6,8H OR ZERO)                       EXR00100
LN 0102          GOTO 2000                                                      EXR00101
LN 0103     16   WRITE(IWC,116)                                                 EXR00102
LN 0104     116  FORMAT(1H+,19X,11HEND OF DATA)                                 EXR00103
LN 0105          GOTO 2000                                                      EXR00104
LN 0106     18   WRITE(IWC,118) INREG,IPROG(INREG)                              EXR00105
LN 0107     118  FORMAT(1H+,19X,8HERROR AT,I6,3X,5HCODE=,I12)                   EXR00106
LN 0108          GOTO 2000                                                      EXR00107
LN 0109     20   WRITE(IWC,120)                                                 EXR00108
LN 0110     120  FORMAT(1H+,19X,33HZERO STEP SIZE IN A FOR STATEMENT)           EXR00109
LN 0111          GOTO 2000                                                      EXR00110
LN 0112     22   WRITE(IWC,122) INREG,IPROG(INREG)                              EXR00111
LN 0113     122  FORMAT(1H ,19X,17HMAT CODE ERROR AT,I6,6H CODE=,I12)           EXR00112
LN 0114          GOTO 2000                                                      EXR00113
LN 0115     24   WRITE(IWC,124) INREG,IPROG(INREG)                              EXR00114
LN 0116     124  FORMAT(1H ,19X,18HFILE CODE ERROR AT,I6,6H CODE=,I12)          EXR00115
LN 0117          GOTO 2000                                                      EXR00116
LN 0118     26   WRITE(IWC,126) I1                                              EXR00117
LN 0119     126  FORMAT(1H+,19X,71HPERHAPS I AM CRAZY BUT I CANNOT FIND AN IMAGE STEXR00118
LN 0120         1ATEMENT WITH THE NUMBER,I6)                                    EXR00119
LN 0121          GOTO 2000                                                      EXR00120
LN 0122     28   WRITE(IWC,128)                                                 EXR00121
```

```
LN 0123     128  FORMAT(1H+,19X,37HTHERE IS AN ATTEMPT TO DIVIDE BY ZERO)          EXR00122
LN 0124          GOTO 2000                                                        EXR00123
LN 0125      30  WRITE(IWC,130) X1,X2                                             EXR00124
LN 0126     130  FORMAT(1H+,19X,50HILLEGAL EXPONENTIATION -- THE ARGUMENT IS NEGATIEXR00125
LN 0127         1VE,E16.8,16H THE EXPONENT IS,E16.8)                             EXR00126
LN 0128          GOTO 2000                                                       EXR00127
LN 0129      32  WRITE(IWC,132) IEXPO                                             ***
LN 0130     132  FORMAT(1H+,19X,35HEXPONENT OVERFLOW IN EXPONENTIATION,?X,9HMAXIMUM***
LN 0131         1 =,I3)                                                          ***
LN 0132          GOTO 2000                                                       EXR00130
LN 0133      34  WRITE(IWC,134) I1,X1                                             EXR00131
LN 0134     134  FORMAT(1H+,19X,41HTHERE IS NOT ENOUGH SPACE FOR ITEM NUMBER,I10,18EXR00132
LN 0135         1H IN LIST OR TABLE ,A1)                                         EXR00133
LN 0136          GOTO 2000                                                       EXR00134
LN 0137      36  WRITE(IWC,136)                                                   EXR00135
LN 0138     136  FORMAT(1H+,19X,27HTHE ARGUMENT OF LOG IS ZERO)                   EXR00136
LN 0139          GOTO 2000                                                       EXR00137
LN 0140      38  WRITE(IWC,138)                                                   EXR00138
LN 0141     138  FORMAT(1H+,19X,31HTHE ARGUMENT OF LOG IS NEGATIVE)              EXR00139
LN 0142          GOTO 2000                                                       EXR00140
LN 0143      40  WRITE(IWC,140)                                                   EXR00141
LN 0144     140  FORMAT(1H+,19X,31HTHE ARGUMENT OF SQR IS NEGATIVE)              EXR00142
LN 0145          GOTO 2000                                                       EXR00143
LN 0146      42  I1=MAXSAT-1000                                                  EXR00144
LN 0147          WRITE(IWC,142) I1                                               EXR00145
LN 0148     142  FORMAT(1H+,19X,30HYOU WANT TO ALLOCATE MORE THAN,I5,10H SENTENCES)EXR00146
LN 0149          GOTO 2000                                                       ***
LN 0150      44  WRITE(IWC,144)                                                   EXR00148
LN 0151     144  FORMAT(1H+,19X,50HTHERE IS AN ATTEMPT TO OPEN AN ALREADY OPENED FIEXR00149
LN 0152         1LE)                                                             EXR00150
LN 0153          GOTO 2000                                                       ***
LN 0154      46  WRITE(IWC,146) MAXFIL                                            EXR00152
LN 0155     146  FORMAT(1H+,19X,15HI CAN OPEN ONLY,I5,6H FILES)                   EXR00153
LN 0156          GOTO 2000                                                       ***
LN 0157      48  WRITE(IWC,148)                                                   EXR00155
LN 0158     148  FORMAT(1H+,19X,67HINDEX-SEQUENTIAL FILE MUST BE OPENED BY A PRECEEEXR00156
LN 0159         1DING OPEN STATEMENT)                                            EXR00157
LN 0160          GOTO 2000                                                       ***
LN 0161      50  WRITE(IWC,150) I1,I2                                             EXR00159
LN 0162     150  FORMAT(1H+,19X,15HSENTENCE NUMBER,I6,32H EXCEEDS MAXIMAL SENTENCE EXR00160
LN 0163         1NUMBER,I6)                                                      EXR00161
LN 0164          GOTO 2000                                                       ***
LN 0165      52  WRITE(IWC,152)                                                   EXR00163
LN 0166     152  FORMAT(1H+,19X,41HYOU CANNOT GET DATA FROM AN UNOPENED FILE)     EXR00164
LN 0167          GOTO 2000                                                       ***
LN 0168      54  WRITE(IWC,154)                                                   EXR00166
LN 0169     154  FORMAT(1H+,19X,33HYOU CANNOT RESET AN UNOPENED FILE)            EXR00167
LN 0170          GOTO 2000                                                       ***
LN 0171      56  WRITE(IWC,156)                                                   EXR00169
LN 0172     156  FORMAT(1H+,19X,19HEND OF FILE REACHED)                          EXR00170
LN 0173          GOTO 2000                                                       ***
LN 0174      58  WRITE(IWC,158)                                                   EXR00172
LN 0175     158  FORMAT(1H+,19X,33HYOU CANNOT CLOSE AN UNOPENED FILE)            EXR00173
LN 0176          GOTO 2000                                                       ***
LN 0177      60  WRITE(IWC,160)                                                   EXR00175
LN 0178     160  FORMAT(1H+,19X,30HERROR IN MATRIX MULTIPLICATION)               EXR00176
LN 0179          GOTO 2000                                                       ***
LN 0180      62  WRITE(IWC,162)                                                   EXR00178
LN 0181     162  FORMAT(1H+,19X,50HNO CORRESPONDING SIZE OF THE SUBSCRIPTS FOR MATREXR00179
LN 0182         1IX)                                                             EXR00180
LN 0183          IF(NERROR.EQ.31) WRITE(IWC,161)                                 EXR00181
```

```
LN 0184          IF(NERROR.EQ.48) WRITE(IWC,163)                                EXR00182
LN 0185      161 FORMAT(1H+,70X,13HTRANSPONATTION)                              EXR00183
LN 0186      163 FORMAT(1H+,70X,9HINVERSION)                                    EXR00184
LN 0187          GOTO 2000                                                      ***
LN 0188       64 WRITE(IWC,164) X1,X2                                           EXR00186
LN 0189      164 FORMAT(1H+,19X,28HCOLUMN INDEX OF THE MATRIX *,A1,21H* IS NEGATIVEEXR00187
LN 0190         1 OR ZERO,F12.2)                                                EXR00188
LN 0191          GOTO 2000                                                      ***
LN 0192       66 WRITE(IWC,166) X1,X2                                           EXR00190
LN 0193      166 FORMAT(1H+,19X,25HROW INDEX OF THE MATRIX *,A1,21H* IS NEGATIVE OREXR00191
LN 0194         1 ZERO,F12.2)                                                   EXR00192
LN 0195          GOTO 2000                                                      ***
LN 0196       68 WRITE(IWC,168) X1,I1,I2,X2                                     EXR00194
LN 0197      168 FORMAT(1H+,19X,35HILLEGAL REDIMENSION OF THE MATRIX *,A1,7H* ROWS=EXR00195
LN 0198         1,I5,9H COLUMNS=,I5,26H  MAXIMAL STORAGE ELEMENTS,F7.0)         EXR00196
LN 0199          GOTO 2000                                                      ***
LN 0200       70 WRITE(IWC,170) X1,I1,I2                                        EXR00198
LN 0201      170 FORMAT(1H+,19X,8HMATRIX *,A1,30H* IS NOT DIMENSIONED N*N ROWS=,I5,EXR00199
LN 0202         19H COLUMNS=,I5)                                                EXR00200
LN 0203          GOTO 2000                                                      ***
LN 0204       72 WRITE(IWC,172)                                                 EXR00202
LN 0205      172 FORMAT(1H+,19X,52HERROR IN MATRIX OPERATION -- ADDITION OR SUBTRACEXR00203
LN 0206         1TION)                                                          EXR00204
LN 0207          GOTO 2000                                                      ***
LN 0208       74 WRITE(IWC,174) IIMAGE,MAXIMA                                   EXR00206
LN 0209      174 FORMAT(1H+,19X,15HOUTPUT EXCEEDS ,I4,30H CHARACTERS AT IMAGE STATEEXR00207
LN 0210         1MENT,I5)                                                       EXR00208
LN 0211          GOTO 2000                                                      EXR00209
LN 0212       76 WRITE(IWC,176) TEXPO                                           ***
LN 0213      176 FORMAT(1H+,19X,28HEXPONENT OVERFLOW IN TANGENT,2X,9HMAXIMUM =,I3) ***
LN 0214          GOTO 2000                                                      EXR00212
LN 0215       78 WRITE(IWC,178) X1                                              EXR00213
LN 0216      178 FORMAT(1H+,19X,47HNO NEGATIVE COLUMN-NUMBER IN PRINT TAB ALLOWED=,EXR00214
LN 0217         1F10.0)                                                         EXR00215
LN 0218          GOTO 2000                                                      EXR00216
LN 0219       80 WRITE(IWC,180) X1,X2,IEXPO                                     ***
LN 0220      180 FORMAT(1H+,19X,48HEXPONENT OVERFLOW OR UNDERFLOW IN MULTIPLICATIONEXR00218
LN 0221         1,E16.9,3H * ,E16.9,2X,9HMAXIMUM =,I3)                          ***
LN 0222          GOTO 2000                                                      EXR00220
LN 0223       82 WRITE(IWC,182) I1                                              EXR00221
LN 0224      182 FORMAT(1H ,109HTHE VALUE OF THE EXPRESSION IN A COMPUTED GOTO OR GEXR00222
LN 0225         1OSUB CANNOT BE USED TO SELECT A STATEMENT NUMBER - VALUE =,I10)  EXR00223
LN 0226          GOTO 2000                                                      EXR00224
LN 0227       84 WRITE(IWC,184)                                                 EXR00225
LN 0228      184 FORMAT(1H+,19X,35HDIVISION ERROR IN MAT LET STATEMENT/)        EXR00226
LN 0229          GOTO 28                                                        EXR00227
LN 0230       86 WRITE(IWC,186)                                                 EXR00228
LN 0231      186 FORMAT(1H+,19X,41HEXPONENTIATION ERROR IN MAT LET STATEMENT/)  EXR00229
LN 0232          IF(NERROR.EQ.44) GOTO 32                                       EXR00230
LN 0233          GOTO 30                                                        EXR00231
LN 0234       90 WRITE(IWC,190)                                                 EXR00232
LN 0235      190 FORMAT(1H+,19X,41HMULTIPLICATION ERROR IN MAT LET STATEMENT/)  EXR00233
LN 0236          GOTO 80                                                        EXR00234
LN 0237       92 WRITE(IWC,192) I2,I1                                           EXR00235
LN 0238      192 FORMAT(1H+,19X,8HSENTENCE,I4,43H CANNOT BE RESET TO NEGATIVE INDEX***
LN 0239         1-POSITION,I4)                                                  ***
LN 0240          GOTO 2000                                                      ***
LN 0241       94 WRITE(IWC,194) I1                                              EXR00239
LN 0242      194 FORMAT(1H+,19X,15HSENTENCE NUMBER,I4,25H IS LESS OR EQUAL TO ZERO)EXR00240
LN 0243          GOTO 2000                                                      ***
LN 0244       98 WRITE(IWC,198) ALPH(I1)                                        EXR00242
```

```
LN 0245     198  FORMAT(1H+,19X,27HDETERMINANT OF THE MATRIX *,A1,9H* IS ZERO)     EXR00243
LN 0246          GOTO 2000                                                        ***
LN 0247     100  WRITE(IWC,101) I1,I1                                            EXR00245
LN 0248     101  FORMAT(1H+,19X,20HI CAN ONLY INVERT A ,I3,3H * ,I3,7H MATRIX)    EXR00246
LN 0249          GOTO 2000                                                        ***
LN 0250     103  WRITE(IWC,105) ALPH(I1),ALPH(I1),ALPH(I2),ALPH(I1)              ***
LN 0251     105  FORMAT(1H+,19X,34HMATRIX MULTIPLICATION OF THE FORM ,A1,1H=,A1,1H****
LN 0252         1,A1,15H NOT ALLOWED - ,A1,40H MUST BE DIMENSIONED AS A ONE-ROW MAT***
LN 0253         2RIX)                                                            ***
LN 0254          GOTO 2000                                                        ***
LN 0255    1999 WRITE(IWC,1998) CARDP                                            EXR00248
LN 0256    1998 FORMAT(10X,80A1)                                                 EXR00249
LN 0257    2000 WRITE(IWC,2001) NERROR                                           EXR00250
LN 0258    2001 FORMAT(20X,11HERROR CODE=,I3)                                    EXR00251
LN 0259          WRITE(IWC,2002) IERR(NERROR)                                    EXR00252
LN 0260    2002 FORMAT(1H ,13X,5(1H*),21H SEE BASIC TEXTBOOK =,2X,A8,2X,5(1H*))  EXR00253
LN 0261          NERROR=-1                                                       EXR00254
LN 0262          IF(NSTLST.GT.0) GOTO 841                                        EXR00255
LN 0263          NSTAT=0                                                         EXR00256
LN 0264          GOTO 840                                                        EXR00257
LN 0265     841  DO 835 K=1,NSTLST                                              EXR00258
LN 0266          IF(ISTLST(K).LT.INREG) GOTO 837                                 EXR00259
LN 0267     835  CONTINUE                                                       EXR00260
LN 0268          K=NSTLST                                                        EXR00261
LN 0269     837  IF(K.LT.2) K=2                                                 EXR00262
LN 0270          NSTAT=LISTST(K-1)                                               EXR00263
LN 0271     840  WRITE(IWC,838) NSTAT                                           EXR00264
LN 0272     838  FORMAT(13X,27H ***** I WAS AT LINE NUMBER,I5)                   EXR00265
LN 0273          NX=3                                                            EXR00266
LN 0274          CALL PRILIN(NX)                                                 EXR00267
LN 0275          IRC=IMIRC                                                       EXR00268
LN 0276          RETURN                                                          EXR00269
LN 0277          END                                                             EXR00270


    USASI FORTRAN DIAGNOSTIC RESULTS FOR EXERR


         NO ERRORS
```

CLEAR

```
LN 0001          SUBROUTINE CLEAR(IANF,IEND)                                    CLR00001
LN 0002     C**** SUBROUTINE TO FILL UP CARP WITH BLANKS                         CLR00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   CLR00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT,CLR00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,CLR00005
LN 0006         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,CLR00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI   CLR00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,   CLR00008
LN 0009         1NIFMAX,INTZEI                                                  CLR00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                     CLR00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),   CLR00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                    CLR00012
LN 0013          COMMON// ISTLST(340),LISTST(340)                               CLR00013
LN 0014          COMMON// DATAN(330)                                            ***
LN 0015          COMMON// DATA(3700)                                            CLR00015
LN 0016          DIMENSION IPROG(3700)                                          CLR00016
LN 0017          EQUIVALENCE (DATA(1),IPROG(1))                                 CLR00017
LN 0018          DO 10 I=IANF,IEND                                              CLR00018
LN 0019          CARP(I)=BLANK                                                  CLR00019
LN 0020       10 CONTINUE                                                       CLR00020
LN 0021          RETURN                                                         CLR00021
LN 0022          END                                                            CLR00022

     USASI FORTRAN DIAGNOSTIC RESULTS FOR CLEAR


         NO ERRORS
```

ZNUMB

```
LN 0001          SUBROUTINE ZNUMB(IANF,IEND,NUM,ISGN,IFIR)                      ZNU00001
LN 0002     C**** SUBROUTINE TO TRANSFORM NUM(INTEGER) INTO DIGITS               ZNU00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   ZNU00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT, ZNU00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  ZNU00005
LN 0006         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZNU00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     ZNU00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,    ZNU00008
LN 0009         1NIFMAX,INTZEI                                                  ZNU00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                     ZNU00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),   ZNU00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                    ZNU00012
LN 0013          COMMON// ISTLST(340),LISTST(340)                               ZNU00013
LN 0014          COMMON// DATAN(330)                                            ***
LN 0015          COMMON// DATA(3700)                                            ZNU00015
LN 0016          DIMENSION IPROG(3700)                                          ZNU00016
LN 0017          EQUIVALENCE (DATA(1),IPROG(1))                                 ZNU00017
LN 0018          DO 10 I=1,80                                                   ZNU00018
LN 0019      10  CARDT(I)=BLANK                                                  ZNU00019
LN 0020          NUMZ=IEND-IANF+1                                               ZNU00020
LN 0021          MEM=NUMZ                                                       ZNU00021
LN 0022          IF(ISGN.EQ.2) GOTO 200                                         ZNU00022
LN 0023          I=1                                                            ZNU00023
LN 0024          IF(NUM.EQ.0) GOTO 500                                          ZNU00024
LN 0025          IF(ISGN.EQ.0) GOTO 35                                          ZNU00025
LN 0026     336  NUMZ=NUMZ-1                                                    ZNU00026
LN 0027          I=I+1                                                          ZNU00027
LN 0028      35  IF(NUMZ.LE.INTNUM) GOTO 355                                   ZNU00028
LN 0029          GOTO 336                                                       ZNU00029
LN 0030     355  IF(NUM.GE.10**NUMZ) GOTO 200                                  ZNU00030
LN 0031      40  NUMZ=NUMZ-1                                                    ZNU00031
LN 0032          IX=NUM/10**NUMZ                                                ZNU00032
LN 0033          IF(IX.NE.0) GOTO 20                                            ZNU00033
LN 0034          IF(IFIR.GE.1) GOTO 20                                          ZNU00034
LN 0035          CARDT(I)=BLANK                                                 ZNU00035
LN 0036          GOTO 30                                                        ZNU00036
LN 0037      20  IFIR=IFIR+1                                                    ZNU00037
LN 0038          IF(IFIR.GT.1) GOTO 25                                          ZNU00038
LN 0039          IF(ISGN.EQ.1) CARDT(I-1)=CMINUS                                ZNU00039
LN 0040      25  CARDT(I)=DIGIT(IX+1)                                          ZNU00040
LN 0041          NUM=NUM-IX*10**NUMZ                                            ZNU00041
LN 0042          IF(NUMZ.EQ.0) GOTO 400                                         ZNU00042
LN 0043      30  I=I+1                                                         ZNU00043
LN 0044          GOTO 40                                                        ZNU00044
LN 0045     C**** SPACE IS TOO SMALL TO INSERT DIGITS                           ZNU00045
LN 0046      200 ZEICH=ASTRSK                                                   ZNU00046
LN 0047      205 IF(ISGN.EQ.1) NUMZ=NUMZ+1                                      ZNU00047
LN 0048          DO 210 I=1,NUMZ                                                ZNU00048
LN 0049      210 CARDT(I)=ZEICH                                                 ZNU00049
LN 0050          GOTO 400                                                       ZNU00050
LN 0051     C**** ONLY NEGATIVE SIGN                                            ZNU00051
LN 0052      300 IF(IFIR.EQ.0) CARDT(NUMZ)=CMINUS                               ZNU00052
LN 0053          GOTO 400                                                       ZNU00053
LN 0054     C**** NUM IS ZERO                                                   ZNU00054
LN 0055      500 CARDT(NUMZ)=DIGIT(1)                                           ZNU00055
LN 0056          IF(ISGN.EQ.1) GOTO 300                                         ZNU00056
LN 0057          IF(IFIR.EQ.0) GOTO 400                                         ZNU00057
LN 0058          ZEICH=DIGIT(1)                                                 ZNU00058
LN 0059          GOTO 205                                                       ZNU00059
LN 0060      400 DO 410 I=1,MEM                                                 ZNU00060
LN 0061          CARP(NPRUS)=CARDT(I)                                           ZNU00061
```

```
LN 0062           NPRUS=NPRUS+1                                          ZNU00062
LN 0063           IF(NPRUS.GT.IIMAGE) GOTO 420                           ZNU00063
LN 0064      410  CONTINUE                                               ZNU00064
LN 0065           RETURN                                                 ZNU00065
LN 0066     C**** OUTPUT BUFFER OVERFLOW                                 ZNU00066
LN 0067      420  NERROR=37                                              ZNU00067
LN 0068           CALL EXERR(NERROR,I1,I2,X1,X2)                         ZNU00068
LN 0069           IANF=-1                                                ZNU00069
LN 0070           RETURN                                                 ZNU00070
LN 0071           END                                                    ZNU00071

      USASI FORTRAN DIAGNOSTIC RESULTS FOR ZNUMB


          NO ERRORS
```

ZIMAGE

```
LN 0001          SUBROUTINE ZIMAGE(ICODE,NSTOP)                                  ZIM00001
LN 0002     C**** SUBROUTINE TO TRANSLATE IMAGE INTO CARP                         ZIM00002
LN 0003          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,    ZIM00003
LN 0004         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST,INEXT, ZIM00004
LN 0005         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL, ZIM00005
LN 0006         3IRC,IWC,NSTEND,TEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, ZIM00006
LN 0007         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI  ZIM00007
LN 0008          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,   ZIM00008
LN 0009         1NIFMAX,INTZEI                                                  ZIM00009
LN 0010          COMMON// CARDT(80),MERKER(26,2),CARP(140),                     ZIM00010
LN 0011         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),   ZIM00011
LN 0012         1IRET(20),XXX(4),NFILE(25,3)                                    ZIM00012
LN 0013          COMMON// ISTLST(340),LISTST(340)                               ZIM00013
LN 0014          COMMON// DATAN(330)                                            ***
LN 0015          COMMON// DATA(3700)                                            ZIM00015
LN 0016          DIMENSION IPROG(3700)                                          ZIM00016
LN 0017          EQUIVALENCE (DATA(1),IPROG(1))                                 ZIM00017
LN 0018     C**** ICODE=1 NUMERIC VARIABLE OR EXPRESSION                        ZIM00018
LN 0019     C**** ICODE=2 ALPHANUMERIC CONSTANT                                 ZIM00019
LN 0020     C**** ICODE=3 ALPHANUMERIC VARIABLE                                 ZIM00020
LN 0021          NSTEU=0                                                        ZIM00021
LN 0022          INEXT=1                                                        ZIM00022
LN 0023          NSTOP=0                                                        ZIM00023
LN 0024          IDEC=0                                                         ZIM00024
LN 0025          IBEG1=0                                                        ZIM00025
LN 0026          IBEG2=0                                                        ZIM00026
LN 0027          IEND1=0                                                        ZIM00027
LN 0028          IEND2=0                                                        ZIM00028
LN 0029          IF((NCARD.NE.1).OR.(NPRUS.NE.1)) GOTO 10                       ZIM00029
LN 0030     C**** CHECK TO SKIP LINES                                           ZIM00030
LN 0031          IF((CARD(2).NE.DOPU).OR.(CARD(3).NE.DOPU)) GOTO 10             ZIM00031
LN 0032          CALL ZDIGIT(CARD(1),IA)                                        ZIM00032
LN 0033          NSKIP=IA-1                                                     ZIM00033
LN 0034          NCARD=4                                                        ZIM00034
LN 0035          IF(IA.LE.10) GOTO 15                                           ZIM00035
LN 0036       20 NSKIP=1                                                        ZIM00036
LN 0037       15 IF(NSKIP.LE.0) GOTO 20                                         ZIM00037
LN 0038          DO 16 I=1,NSKIP                                                ZIM00038
LN 0039          WRITE(IWC,17)                                                  ZIM00039
LN 0040       17 FORMAT(1H )                                                    ZIM00040
LN 0041       16 CONTINUE                                                       ZIM00041
LN 0042       10 IF(ICODE.EQ.9) GOTO 810                                        ZIM00042
LN 0043          DO 30 I=NCARD,80                                               ZIM00043
LN 0044          IF(CARD(I).EQ.QUOTE) GOTO 100                                  ZIM00044
LN 0045          IF(CARD(I).EQ.DOPU) GOTO 40                                    ZIM00045
LN 0046          CARP(NPRUS)=CARD(I)                                            ZIM00046
LN 0047          NPRUS=NPRUS+1                                                  ZIM00047
LN 0048          IF(NPRUS.GT.IIMAGE) GOTO 2556                                  ZIM00048
LN 0049       30 CONTINUE                                                       ZIM00049
LN 0050       40 IF(NPRI.EQ.-1) GOTO 45                                         ZIM00050
LN 0051          NX=2                                                           ZIM00051
LN 0052          CALL PRILIN(NX)                                                ZIM00052
LN 0053       45 NCARD=1                                                        ZIM00053
LN 0054          IF((CARD(2).EQ.DOPU).AND.(CARD(3).EQ.DOPU)) NCARD=4            ZIM00054
LN 0055          GOTO 10                                                        ZIM00055
LN 0056      100 IBEG1=I                                                        ZIM00056
LN 0057          IF(I.EQ.1) GOTO 101                                            ZIM00057
LN 0058          IF(CARD(I-1).EQ.DECMAL) IDEC=I-1                               ZIM00058
LN 0059      101 DO 110 I=IBEG1,80                                              ZIM00059
LN 0060          IF(CARD(I).EQ.DECMAL) GOTO 115                                 ZIM00060
LN 0061          IF(CARD(I).EQ.EXSIGN) GOTO 120                                 ZIM00061
```

```
LN 0062          IF(CARD(I).NE.QUOTE) GOTO 130                                  ZIM00062
LN 0063          GOTO 110                                                       ZIM00063
LN 0064      115 IDEC=I                                                         ZIM00064
LN 0065      110 CONTINUE                                                       ZIM00065
LN 0066          I=81                                                           ZIM00066
LN 0067      130 IEND1=I-1                                                      ZIM00067
LN 0068          IF(IDEC.EQ.0) GOTO 200                                         ZIM00068
LN 0069          GOTO 300                                                       ZIM00069
LN 0070      120 IEND1=I-1                                                      ZIM00070
LN 0071          IBEG2=I                                                        ZIM00071
LN 0072          DO 135 I=IBEG2,80                                              ZIM00072
LN 0073          IF(CARD(I).NE.FXSIGN) GOTO 140                                 ZIM00073
LN 0074      135 CONTINUE                                                       ZIM00074
LN 0075          I=81                                                           ZIM00075
LN 0076      140 IEND2=I-1                                                      ZIM00076
LN 0077          GOTO 400                                                       ZIM00077
LN 0078     C**** I-FORMAT FOUND (INTEGER)                                      ZIM00078
LN 0079      200 IF(ICODE.GE.2) GOTO 2000                                       ZIM00079
LN 0080          IFIR=0                                                         ZIM00080
LN 0081      201 IF(ACC.LT.FLOAT(INTMAX)) GOTO 2011                             ZIM00081
LN 0082          IF(NSTEU.EQ.1) GOTO 218                                        ZIM00082
LN 0083          IF(IEND1-4.LT.IBEG1) GOTO 350                                  ZIM00083
LN 0084          IEND2=IEND1                                                    ZIM00084
LN 0085          IEND1=IEND1-5                                                  ZIM00085
LN 0086          IBEG2=IEND1+1                                                  ZIM00086
LN 0087      218 NSTEU=2                                                        ZIM00087
LN 0088          NEXPO=0                                                        ZIM00088
LN 0089      217 IF(ACC.LT.FLOAT(INTMAX)) GOTO 2011                             ZIM00089
LN 0090          NEXPO=NEXPO+1                                                  ZIM00090
LN 0091          ACC=ACC/10.                                                    ZIM00091
LN 0092          GOTO 217                                                       ZIM00092
LN 0093     2011 INPRI=ACC                                                      ZIM00093
LN 0094          ISGN=0                                                         ZIM00094
LN 0095          IF(INPRI.GE.0) GOTO 208                                        ZIM00095
LN 0096          ISGN=1                                                         ZIM00096
LN 0097          INPRI=-INPRI                                                   ZIM00097
LN 0098      208 IF(NSTEU.GE.1) GOTO 205                                        ZIM00098
LN 0099          CALL ZNUMB(IBEG1,IEND1,INPRI,ISGN,IFIR)                        ZIM00099
LN 0100          GOTO 500                                                       ZIM00100
LN 0101     C**** F-FORMAT FOUND (REAL)                                         ZIM00101
LN 0102      300 IF(ICODE.GE.2) GOTO 2000                                       ZIM00102
LN 0103          IF(IEND1.NE.IDEC) GOTO 301                                     ZIM00103
LN 0104          IEND1=IDEC-1                                                   ZIM00104
LN 0105          GOTO 200                                                       ZIM00105
LN 0106      301 ISGN=0                                                         ZIM00106
LN 0107          IF(ACC.GE.0.) GOTO 310                                         ZIM00107
LN 0108          ISGN=1                                                         ZIM00108
LN 0109          ACC=-ACC                                                       ZIM00109
LN 0110      310 INPRI=ACC                                                      ZIM00110
LN 0111          IF(INPRI.EQ.0) GOTO 330                                        ZIM00111
LN 0112          IF(IDEC.LT.IBEG1) GOTO 350                                     ZIM00112
LN 0113      340 IFIR=0                                                         ZIM00113
LN 0114          CALL ZNUMB(IBEG1,IDEC-1,INPRI,ISGN,IFIR)                       ZIM00114
LN 0115          IF(IBEG1.EQ.-1) GOTO 2555                                      ZIM00115
LN 0116          GOTO 320                                                       ZIM00116
LN 0117      330 IF((ISGN.EQ.1).AND.(IDEC.LE.IBEG1)) GOTO 360                   ZIM00117
LN 0118          IF((IDEC.LE.IBEG1).AND.(ISGN.EQ.0)) GOTO 345                   ZIM00118
LN 0119          GOTO 340                                                       ZIM00119
LN 0120      320 CARP(NPRUS)=DECMAL                                             ZIM00120
LN 0121          NPRUS=NPRUS+1                                                  ZIM00121
LN 0122      345 ISGN=0                                                         ZIM00122
```

```
LN 0123          INPRI=(ACC-INT(ACC))*10**(IEND1-IDEC)                  ZIM00123
LN 0124          IFIR=1                                                 ZIM00124
LN 0125          CALL ZNUMB(IDEC+1,IEND1,INPRI,ISGN,IFIR)               ZIM00125
LN 0126          GOTO 500                                               ZIM00126
LN 0127      350 ISGN=2                                                 ZIM00127
LN 0128          CALL ZNUMB(IBEG1,IEND1,INPRI,ISGN,IFIR)                ZIM00128
LN 0129          GOTO 500                                               ZIM00129
LN 0130      360 CARP(NPRUS-1)=CMINUS                                   ZIM00130
LN 0131          CARP(NPRUS)=DECMAL                                     ZIM00131
LN 0132          NPRUS=NPRUS+1                                          ZIM00132
LN 0133          IDEC=IDEC+1                                            ZIM00133
LN 0134          IF(NSTEU.EQ.1) GOTO 1005                               ZIM00134
LN 0135          GOTO 345                                               ZIM00135
LN 0136     C**** TRANSLATE E-FORMAT                                    ZIM00136
LN 0137     400  IF(ICODE.GE.2) GOTO 2000                              ZIM00137
LN 0138          IFIR=1                                                 ZIM00138
LN 0139          NSTEU=1                                                ZIM00139
LN 0140          NEXPO=0                                                ZIM00140
LN 0141          ISGN=0                                                 ZIM00141
LN 0142          IF(IDEC.NE.0) GOTO 1000                                ZIM00142
LN 0143     C**** INTEGER WITH EXPONENT                                 ZIM00143
LN 0144          GOTO 201                                               ZIM00144
LN 0145      205 MEM=IEND1-IBEG1+1                                      ZIM00145
LN 0146          IF(ISGN.EQ.1) MEM=MEM-1                                ZIM00146
LN 0147          IF(MEM.LE.INTNUM) GOTO 405                             ZIM00147
LN 0148          MEM=INTNUM                                             ZIM00148
LN 0149      405 IF(INPRI.GE.10**MEM) GOTO 410                          ZIM00149
LN 0150          IF(NSTEU.EQ.2) GOTO 4051                               ZIM00150
LN 0151          IF(INPRI.LT.10**(MEM-1)) GOTO 420                      ZIM00151
LN 0152     4051 CALL ZNUMB(IBEG1,IEND1,INPRI,ISGN,IFIR)                ZIM00152
LN 0153          GOTO 550                                               ZIM00153
LN 0154      410 NEXPO=NEXPO+1                                          ZIM00154
LN 0155          INPRI=INPRI/10                                         ZIM00155
LN 0156          GOTO 405                                               ZIM00156
LN 0157      420 NEXPO=NEXPO-1                                          ZIM00157
LN 0158          INPRI=INPRI*10                                         ZIM00158
LN 0159          GOTO 405                                               ZIM00159
LN 0160     C**** F-FORMAT WITH EXPONENT                                ZIM00160
LN 0161     1000 IF(IDEC.GT.IBEG1) GOTO 1100                           ZIM00161
LN 0162          IF(ACC.GT.0.) GOTO 1005                                ZIM00162
LN 0163          GOTO 360                                               ZIM00163
LN 0164     1005 MEM=0                                                 ZIM00164
LN 0165          GOTO 1205                                              ZIM00165
LN 0166     1100 IF(ACC.GE.0.) GOTO 1200                               ZIM00166
LN 0167          ISGN=1                                                 ZIM00167
LN 0168          IFIR=0                                                 ZIM00168
LN 0169          ACC=-ACC                                               ZIM00169
LN 0170     1200 MEM=IDEC-IBEG1                                        ZIM00170
LN 0171          IF(ISGN.EQ.1) MEM=MEM-1                                ZIM00171
LN 0172          IF(MEM.GT.0) GOTO 1205                                 ZIM00172
LN 0173          CARP(NPRUS)=CMINUS                                     ZIM00173
LN 0174          CARP(NPRUS+1)=DECMAL                                   ZIM00174
LN 0175          NPRUS=NPRUS+2                                          ZIM00175
LN 0176     1205 IF(ACC.GE.FLOAT(10**MEM)) GOTO 1210                   ZIM00176
LN 0177          IF(ACC.LT.FLOAT(10**(MEM-1))) GOTO 1220                ZIM00177
LN 0178          IF(MEM.EQ.0) GOTO 451                                  ZIM00178
LN 0179          INPRI=ACC                                              ZIM00179
LN 0180          CALL ZNUMB(IBEG1,IDEC-1,INPRI,ISGN,IFIR)               ZIM00180
LN 0181          IF(IBEG1.EQ.-1) GOTO 2555                              ZIM00181
LN 0182          GOTO 450                                               ZIM00182
LN 0183     1210 NEXPO=NEXPO+1                                         ZIM00183
```

```
LN 0184          ACC=ACC/10.                                          ZIM00184
LN 0185          GOTO 1205                                            ZIM00185
LN 0186    1220  NEXPO=NEXPO-1                                        ZIM00186
LN 0187          ACC=ACC*10.                                          ZIM00187
LN 0188          GOTO 1205                                            ZIM00188
LN 0189     450  CARP(NPRUS)=DECMAL                                   ZIM00189
LN 0190          NPRUS=NPRUS+1                                        ZIM00190
LN 0191     451  INPRI=(ACC-INT(ACC))*10**(IEND1-IDEC)                ZIM00191
LN 0192          ISGN=0                                               ZIM00192
LN 0193          IBEG1=IDEC+1                                         ZIM00193
LN 0194          CALL ZNUMB(IBEG1,IEND1,INPRI,ISGN,IFIR)              ZIM00194
LN 0195    C**** TRANSLATE EXPONENT                                   ZIM00195
LN 0196     550  CARP(NPRUS)=ALPH(5)                                  ZIM00196
LN 0197          CARP(NPRUS+1)=PLUS                                   ZIM00197
LN 0198          IF(NEXPO.GE.0) GOTO 555                              ZIM00198
LN 0199          CARP(NPRUS+1)=CMINUS                                 ZIM00199
LN 0200          NEXPO=-NEXPO                                         ZIM00200
LN 0201     555  NPRUS=NPRUS+2                                        ZIM00201
LN 0202          ISGN=0                                               ZIM00202
LN 0203          IFIR=1                                               ZIM00203
LN 0204          IBEG1=IBEG2+2                                        ZIM00204
LN 0205          CALL ZNUMB(IBEG1,IEND2,NEXPO,ISGN,IFIR)              ZIM00205
LN 0206          IEND1=IEND2                                          ZIM00206
LN 0207          GOTO 500                                             ZIM00207
LN 0208    C**** PRINT ALPHANUMERIC VARIABLE                          ZIM00208
LN 0209    2000  IF((IDEC.NE.0).OR.(IBEG2.NE.0)) GOTO 2010            ZIM00209
LN 0210          IF(ICODE.EQ.2) GOTO 3000                             ZIM00210
LN 0211          IX=ACC                                               ZIM00211
LN 0212    3020  MEM=IEND1-IBEG1+1                                    ZIM00212
LN 0213          DO 16400 I=1,5                                       ZIM00213
LN 0214          IZ=IX-((IX/48)*48)                                   ZIM00214
LN 0215          IF(I.GT.MEM) GOTO 16401                              ZIM00215
LN 0216          IF(NPRUS.GT.IIMAGE) GOTO 2556                        ZIM00216
LN 0217          CARP(NPRUS)=ALPH(IZ+1)                               ZIM00217
LN 0218          NPRUS=NPRUS+1                                        ZIM00218
LN 0219          IX=IX/48                                             ZIM00219
LN 0220   16400  CONTINUE                                             ZIM00220
LN 0221   16401   IF(IDEC.GT.0) GOTO 3010                            ZIM00221
LN 0222          GOTO 500                                             ZIM00222
LN 0223    2010  IF(IEND2.NE.0) IEND1=IEND2                          ZIM00223
LN 0224          GOTO 350                                             ZIM00224
LN 0225    C**** ALPHANUMERIC CONSTANT                                ZIM00225
LN 0226    3000  IDEC=IPROG(INREG)                                    ZIM00226
LN 0227          INREG=INREG-1                                        ZIM00227
LN 0228    3010  IX=IPROG(INREG)                                     ZIM00228
LN 0229          INREG=INREG-1                                        ZIM00229
LN 0230          IDEC=IDEC-1                                          ZIM00230
LN 0231          GOTO 3020                                            ZIM00231
LN 0232     500  NCARD=IEND1+1                                       ZIM00232
LN 0233          IF(IBEG1.EQ.-1) NSTOP=-1                             ZIM00233
LN 0234     510  DO 510 I=NCARD,80                                    ZIM00234
LN 0235          IF(CARD(I).EQ.QUOTE) GOTO 520                        ZIM00235
LN 0236          IF(CARD(I).EQ.DOPU) GOTO 525                         ZIM00236
LN 0237          IF(NPRUS.GT.IIMAGE) GOTO 2556                        ZIM00237
LN 0238          CARP(NPRUS)=CARD(I)                                  ZIM00238
LN 0239          NPRUS=NPRUS+1                                        ZIM00239
LN 0240     510  CONTINUE                                             ZIM00240
LN 0241     520  NCARD=I                                              ZIM00241
LN 0242          IF(IPROG(INREG).EQ.-67) NPRI=1                       ZIM00242
LN 0243          RETURN                                               ZIM00243
LN 0244     525  NCARD=1                                              ZIM00244
```

```
LN 0245           IF((NPRI.EQ.-1).AND.(IPROG(INREG).EQ.-67)) GOTO 526          ZIM00245
LN 0246           IF((NPRI.EQ.1).AND.(IPROG(INREG).EQ.-69)) RETURN             ZIM00246
LN 0247           IF(NPRI.EQ.-1) RETURN                                        ZIM00247
LN 0248      526  NX=2                                                         ZIM00248
LN 0249           CALL PRILIN(NX)                                              ZIM00249
LN 0250           RETURN                                                       ZIM00250
LN 0251     2555  NSTOP=-1                                                    ZIM00251
LN 0252           RETURN                                                       ZIM00252
LN 0253     2556  NERROP=37                                                    ZIM00253
LN 0254           CALL EXERR(NERROR,I1,I2,X1,X2)                               ZIM00254
LN 0255           NSTOP=-1                                                     ZIM00255
LN 0256           RETURN                                                       ZIM00256
LN 0257           END                                                          ZIM00257
```

USASI FORTRAN DIAGNOSTIC RESULTS FOR ZIMAGE

LINE	S	ERRNUM	MESSAGE
0123	N	0020	NON-USASI MIXED MODE USAGE WITH ARITHMETIC OPERATOR.
0191	N	0020	NON-USASI MIXED MODE USAGE WITH ARITHMETIC OPERATOR.

```
LN 0001          SUBROUTINE PRILTN(NX)                                          PRI00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   PRI00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTDT,NIFOR,NIRET,NSTLST,INEXT,PRI00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,PRI00004
LN 0005         3IRC,IWC,NSTEND,TEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE,PRI00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI  PRI00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DNOPU,IMIRC,SMALL,ISTMAX,NIPMAX,  PRI00007
LN 0008         1NIFMAX,INTZEI                                                 PRI00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                     PRI00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),  PRI00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                   PRI00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                               PRI00012
LN 0013          COMMON// DATAN(330)                                            ***
LN 0014          COMMON// DATA(3700)                                            PRI00014
LN 0015          DIMENSION IPROG(3700)                                          PRI00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                 PRI00016
LN 0017          DO 20 I=1,IIMAGE                                               ***
LN 0018          IF(CARP(I).EQ.QUOTE) CARP(I)=BLANK                             ***
LN 0019     20   CONTINUE                                                       ***
LN 0020          IF(NX.EQ.1) GOTO 10                                            PRI00017
LN 0021          IF((NX.EQ.3).AND.(NPPUS.EQ.1)) GOTO 10                         PRI00018
LN 0022          IF(NPRUS.EQ.1) GOTO 40                                         PRI00019
LN 0023          WRITE(IWC,998) (CARP(I),I=1,IIMAGE)                            PRI00020
LN 0024          GOTO 30                                                        PRI00021
LN 0025     10   IF(INEXT.EQ.1) GOTO 40                                         PRI00022
LN 0026          WRITE(IWC,999) (CARP(I),I=1,IWRIT)                             PRI00023
LN 0027     30   CALL CLEAR(1,140)                                              PRI00024
LN 0028          INEXT=1                                                        PRI00025
LN 0029          NPRUS=1                                                        PRI00026
LN 0030     999  FORMAT(1X,125A1)                                              PRI00027
LN 0031     998  FORMAT(1X,135A1)                                              PRI00028
LN 0032     40   RETURN                                                         PRI00029
LN 0033          END                                                            PRI00030


     USASI FORTRAN DIAGNOSTIC RESULTS FOR PRILIN


          NO ERRORS
```

CHECK

```
LN 0001          SUBROUTINE CHECK(I1,I2)                                        CHE00001
LN 0002          COMMON// ACC,ASTRSK,BLANK,CMINUS,COMMA,DECMAL,DOLSGN,EQUALS,   CHE00002
LN 0003         1INREG,LNGCRP,NCELLD,NCELLP,NERRS,NEXTOT,NIFOR,NIRET,NSTLST,INEXT, CHE00003
LN 0004         2NUMBUF,PARLFT,PARRT,PLUS,QUOTE,SLASH,VLESS,VGREAT,DQUOTE,MAXFIL,  CHE00004
LN 0005         3IRC,IWC,NSTEND,IEXPO,IBEGST,IWRIT,IPEND,IZONE,IIMAGE,NPRI,NIMAGE, CHE00005
LN 0006         4NPRUS,NCARD,MAXIMA,PUCO,DOPU,EXSIGN,MAXSAT,NUMFIL,NZIM,NSTZEI     CHE00006
LN 0007          COMMON// INTMAX,INTNUM,XNULL,DDOPU,IMIRC,SMALL,ISTMAX,NIRMAX,     CHE00007
LN 0008         1NIFMAX,INTZEI                                                     CHE00008
LN 0009          COMMON// CARDT(80),MERKER(26,2),CARP(140),                        CHE00009
LN 0010         3ALPH(48),BUFFER(40),CARD(80),CARDP(80),DIGIT(10),IFOR(20,2),      CHE00010
LN 0011         1IRET(20),XXX(4),NFILE(25,3)                                       CHE00011
LN 0012          COMMON// ISTLST(340),LISTST(340)                                  CHE00012
LN 0013          COMMON// DATAN(330)                                               ***
LN 0014          COMMON// DATA(3700)                                               CHE00014
LN 0015          DIMENSION IPROG(3700)                                             CHE00015
LN 0016          EQUIVALENCE (DATA(1),IPROG(1))                                    CHE00016
LN 0017          NX=2                                                              CHE00017
LN 0018          IF(I2.GE.IWRIT) CALL PRILIN(NX)                                   CHE00018
LN 0019          IF(I2.LE.1) RETURN                                                CHE00019
LN 0020          I1=1                                                              CHE00020
LN 0021     1    I1=I1+IZONE                                                       CHE00021
LN 0022          IF(I1.LT.I2) GOTO 1                                               CHE00022
LN 0023          I2=1                                                              CHE00023
LN 0024          NPRUS=I2                                                          CHE00024
LN 0025          INEXT=I1                                                          CHE00025
LN 0026          RETURN                                                            CHE00026
LN 0027          END                                                               CHE00027

      USASI FORTRAN DIAGNOSTIC RESULTS FOR CHECK

           NO ERRORS
```

```
   LN 0001          SUBROUTINE SUBINV(I,M)                                INV00001
   LN 0002          COMMON/A/ D(12,12)                                    ***
   LN 0003          DIMENSION E(10,10)                                    INV00003
   LN 0004          DO 10 J=1,I                                           INV00004
   LN 0005          DO 20 K=1,I                                           INV00005
   LN 0006          E(J,K)=0.                                             INV00006
   LN 0007     20   CONTINUE                                              INV00007
   LN 0008     10   CONTINUE                                              INV00008
   LN 0009          DO 30 J=1,I                                           INV00009
   LN 0010          E(J,J)=1.                                             INV00010
   LN 0011     30   CONTINUE                                              INV00011
   LN 0012          DO 40 J=1,I                                           INV00012
   LN 0013          P=D(J,J)                                              INV00013
   LN 0014          IF(P.EQ.0.) GOTO 110                                  INV00014
   LN 0015          DO 50 K=1,I                                           INV00015
   LN 0016          E(J,K)=E(J,K)/P                                       INV00016
   LN 0017          D(J,K)=D(J,K)/P                                       INV00017
   LN 0018     50   CONTINUE                                              INV00018
   LN 0019          L=1                                                   INV00019
   LN 0020     80   IF(L.EQ.J) GOTO 60                                    INV00020
   LN 0021          P=D(L,J)                                              INV00021
   LN 0022          DO 70 K=1,I                                           INV00022
   LN 0023          E(L,K)=E(L,K)-E(J,K)*P                                INV00023
   LN 0024          D(L,K)=D(L,K)-D(J,K)*P                                INV00024
   LN 0025     70   CONTINUE                                              INV00025
   LN 0026     60   L=L+1                                                 INV00026
   LN 0027          IF(L.LE.I) GOTO 80                                    INV00027
   LN 0028     40   CONTINUE                                              INV00028
   LN 0029          DO 90 J=1,I                                           INV00029
   LN 0030          DO 100 K=1,I                                          INV00030
   LN 0031          D(J,K)=E(J,K)                                         INV00031
   LN 0032    100   CONTINUE                                              INV00032
   LN 0033     90   CONTINUE                                              INV00033
   LN 0034          RETURN                                                INV00034
   LN 0035    110   M=-1                                                  INV00035
   LN 0036          RETURN                                                INV00036
   LN 0037          END                                                   INV00037

        USASI FORTRAN DIAGNOSTIC RESULTS FOR SUBINV

             NO ERRORS

THE FOLLOWING ARE COMMON BLOCK NAMES OR NAMES NOT ASSIGNED STORAGE

   A
RLDR(BASI,PUM,LGO)
AUX,ALIB
```

EDV in 20 Stunden

Ein IDV-Lernprogramm in

47 Lernschritten mit
82 Fotos und Illustrationen
81 Fragen und Lösungen
17 Repetitionshinweisen

kartoniert Fr./DM 18.–

Nach erfolgreichem Studium dieses Lernprogramms kennen Sie die Bedeutung und Arbeitsweise der elektronischen Daten-Verarbeitung (EDV). Dieses Wissen hilft Ihnen, den Einsatz der EDV in einem modernen Betrieb besser zu verstehen.

Inhalt

Was ist Datenverarbeitung?
Was sind Daten?
Wie können Daten dem Computer eingegeben werden?
Wie ist ein Computer aufgebaut?
Wie entsteht ein Programm?
Wie und für was kann ein Computer eingesetzt werden?
Welche EDV-Berufsgruppen gibt es?

VERLAG PAUL HAUPT BERN UND STUTTGART

Institut für Elektronische Datenverarbeitung, Zürich (Hrsg.)

Elektronische Datenverarbeitung

Ein PU-Lehrgang mit Steuertexten für Manager, Sachbearbeiter, EDV-Fachleute, Studenten, allgemein Interessierte.
„IDV-Lernprogramm". 1971.
Band 1: Einführung in die elektronische Datenverarbeitung. 291 Seiten, Gebunden 68.–
Band 2: Grundlagen der EDV-Organisation. 325 Seiten, Gebunden 68.–

FORTRAN

Ein PU-Lehrgang mit Programmbeispielen für Ingenieure, Techniker, Ökonomen, Naturwissenschafter.
„IDV-Lernprogramm". 1971.
Band 1: Basistext und Lösungen
Band 2: Steuertext und Übungen.
Komplett 212 Seiten, Gebunden 68.–

Algebra für EDV

Ein PU-Lehrgang mit Repetitorium, Aufgaben und Lösungen in COBOL und FORTRAN.
„IDV-Lernprogramm". 1974. 159 Seiten, Spiralheftung 68.–

Moderne Programmablauf-Planung

Ein PU-Lehrgang mit Ablaufplänen, Entscheidungstabellen, normierter Programmierung und Lösungsbeispielen.
„IDV-Lernprogramm". 1975. 229 Seiten, Spiralheftung 68.–

Verlag de Gruyter Berlin + New York / Verlag Paul Haupt Bern

P. Appel

Schweizerische Versuche zur Realisierung integrierter Systeme der Planung, Steuerung und Kontrolle in der Fertigungsindustrie

„Prüfen und Entscheiden" 6. 1970. 112 Seiten, kart. 28.–

Um das Phänomen von Theorie und Praxis der integrierten Datenverarbeitung zu untersuchen, hat der Autor bei sechs ausgewählten Industriefirmen Erhebungen vorgenommen und hält die Ergebnisse gesamthaft und für jede Firma einzeln fest.

R. Baer

Die Dialog-Datenverarbeitung

Konzept und Anwendung in der Praxis

„Planung und Kontrolle in der Unternehmung" 5. 1975. 160 Seiten, geb. 48.–

Ziel des Buches ist es, ausgehend von einer grundsätzlichen Auseinandersetzung mit dem Mensch-Maschinen-Dialog, dem EDV-Anwender neue, mehr Erfolg versprechende Wege des Computereinsatzes aufzuzeigen.

E. Fehr

Produktionsplanung und -steuerung mit elektronischer Datenverarbeitung

Theoretische Konzeption und ihre praktische Realisierbarkeit

„Planung und Kontrolle in der Unternehmung" 2. 1968. 208 Seiten mit 9 Zeichnungen und 3 Falttafeln, geb. 32.80

Der Wert der Arbeit liegt in der Vermittlung einer Gesamtschau, die dem Betriebswirtschafter, der nicht gerade Fachmann auf dem Gebiete der elektronischen Datenverarbeitung ist, ein sachliches und gutes Bild gibt.

H. Zuberbühler

Elektronische Datenverarbeitung in der Industrie

Ergebnisse und empirische Untersuchungen

„Führung und Organisation der Unternehmung" 17. 1973. 210 Seiten mit 6 Darstellungen und 59 Tabellen, geb. 44.–

Der Verfasser hat in 30 schweizerischen Industrieunternehmungen verschiedener Grössen und Branchen die elektronische Datenverarbeitung einer detaillierten Analyse unterzogen.

VERLAG PAUL HAUPT BERN UND STUTTGART

Prof. Dr. Karl Weber
Dipl.-Ing. Carl Wolfram Türschmann

BASIC

Lehr- und Handbuch der Programmiersprache BASIC mit wirtschaftswissenschaftlichen Anwendungsbeispielen

Band 1: 228 Seiten mit 31 Zeichnungen, 16 Tabellen und vielen Computerausdrucken.
Uni-Taschenbücher 588. Kart. DM 20.80

Band 2: 244 Seiten mit 13 Zeichnungen, 9 Tabellen und vielen Computerausdrucken.
Uni-Taschenbücher 589. Kart. DM 20.80

Die behandelten Themen sind:

1. BASIC-Charakteristik: Darstellung der Entwicklung der Programmiersprache BASIC: Dialogbetrieb; Alternativbetrieb; Monologbetrieb.
2. BASIC-Problembearbeitung: Datenfluss- und Programmablaufspläne im allgemeinen; BASIC-orientierte Programmablaufspläne.
3. BASIC-Struktur: Beschreibung der Programmiersprache BASIC mit allen Statement-Formen; tabellarische Vergleiche verschiedener Compiler.
4. BASIC-Anwendungsprogramme: Programme aus Statistik, Volks- und Betriebswirtschaftslehre, inklusive Operations Research.
5. UGBIC-Compiler. Beschreibung des University of Giessen BASIC Interpretive Compilers.

Verlag Paul Haupt Bern und Stuttgart

Dieses Buch schliesst unmittelbar an das als UTB-Band 588 und 589 veröffentlichte Werk «BASIC 1 und 2. Lehr- und Handbuch der Programmiersprache BASIC mit wirtschaftswissenschaftlichen Anwendungsbeispielen» an. Während in den beiden ersten Bänden Struktur und Einsatzmöglichkeiten der Programmiersprache BASIC behandelt werden, befasst sich der dritte Band der BASIC-Trilogie mit BASIC-Compilern auf FORTRAN-Basis.

Das erste Kapitel vermittelt einen konzisen Überblick über die Entwicklung der Programmiersprache BASIC. Im zweiten Kapitel werden die BASIC-Compiler UWBIC, FOSBIC und VIEWIT durch entsprechende Hinweise auf Ursprung, Sprachumfang und Compilerstruktur charakterisiert. Im dritten Kapitel wird das FOSBIC-Compilersystem dargestellt, wobei speziell auf FOSBIC-Entwicklung, -Struktur und -Implementation eingetreten wird.