

```
1 $C COPIEREN DER COMPILER-AENDERUNGEN
2 $JOB
3 $DELETE, DJ(F)
4 $PACK, DJ
5 $COPY, MS, DJ(F)
6 $CC=DJ(F)
7     END
```

```
1 $C DJ(FBEF) ERSTELLUNG DES BEFEHLSSTREIFENS
2 $JØB
3 $FBEF
4 $CC=XX(FJX)
5     END
```

```

1      DIMENSION ICØR(4),MPHA(28)
2      DATA IRAT/31808/
3      DATA MPHA/'01','02','03','04','05','06','07','08','09',
4      1'10','11','12','13','14','15','16','17','18','20','21',
5      2'22','23','24','25','26','27','28','29'/
6      WRITE(7,1000)
7      1000  FØRMAT('GEBE CØRE-GRØESSE IN A4')
8      READ(7,1001)ICØR
9      1001  FØRMAT(4A1)
10     WRITE(2,1002)
11     1002  FØRMAT('$C DJ(F) CØMPILER AENDERUNGEN'/'$JØB')
12     IAD=ICØR(1)+256
13     WRITE(2,2000)IRAT
14     2000  FØRMAT('$SI=DS(8FØØ1)''/'$EDIT'/A1'FZZ50+1,1')
15     WRITE(2,2001)IAD,IRAT
16     2001  FØRMAT('      MDX  L3 PØLRX-/'A1'ØØ4'/A1'C=Y')
17     WRITE(2,2002)
18     2002  FØRMAT('$DELETE,DS(8FØØ1)''/'$PACK,DS''/'$CØPY,WS,DS(8FØØ1))
19     DØ 1 M=1,2
20     WRITE(2,1009)
21     1009  FØRMAT('$PACK,DS')
22     DØ 1 I=1,14
23     IND=I+14*(M-1)
24     IPHA=MPHA(IND)
25     WRITE(2,1003)IPHA
26     1003  FØRMAT('$SI=DS(8FO'A2)''/'$EDIT')
27     WRITE(2,1004)IRAT
28     1004  FØRMAT(A1,'MEMBY,1')
29     WRITE(2,1005)ICØR,IPAT
30     1005  FØRMAT('MEMBY EQU      /'4A1/A1'C=Y')
31     WRITE(2,1006)IPHA
32     1006  FØRMAT('$DELETE,DS(8FO'A2)')
33     WRITE(2,1007)IPHA
34     1007  FØRMAT('$CØPY,WS,DS(8FO'A2)')
35     1      CØNTINUE
36     WRITE(2,1008)
37     1008  FØRMAT('$PACK,DS''/'$CC=DJ(8FØØØ)''/'      END')
38     STØP
39     END

```

```

DBØS CC
? $EØD
? $JØB
? $CIC
? * J
? *BUILD
  *BUILD
DBØS CC
? $JØB
? $LØAD
  GEBE CØRE-GRØESSE IN A4
? 3FFE
  STØP
DBØS CC
? $CØPY,WS,TY
  $C DJ(F) CØMPILER AENDERUNGEN
  $JØB
  $SI=DS(8F001)
  $EDIT
  @FZZ50+1,1
    MDX L3 RØLFX-/4004
  @C=Y
  $DELETE,DS(8F001)
  $PACK,DS
  $CØPY,WS,DS(8F001)
  $PACK,DS
  $SI=DS(8F001)
  $EDIT
  @MEMRY,1
  MEMRY EQU      /3FFE
  @C=Y
  $DELETE,DS(8F001)
  $CØPY,WS,DS(8F001)
  $SI=DS(8F002)
  $EDIT
  @MEMRY,1
  MEMRY EQU      /3FFE
  @C=Y
  $DELETE,DS(8F002)
  $CØPY,WS,DS(8F002)
  $SI=DS(8F003)
  $EDIT
  @MEMRY,1
  MEMRY EQU      /3FFE
  @C=Y
  $DELETE,DS(8F003)
  $CØPY,WS,DS(8F003)
  $SI=DS(8F004)
  **CØNSØLE INTERRUPT
  **AT LØCATION 03AF
DBØS CC
? $DELETE,DJ(FBEF)
? $PACK,DJ
? $CØPY,WS,DJ(FBEF)
? $JØB
? $SI=XX(FBEF)
? $LIST
  GEBE PROGRAMMNAME MAX 40 ZEICHEN
? XX(FBEF)
  GEBE STARTZEILE FØRMAT 15
?

```

+++++

```

1  *   GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2  ****
3  *
4  *   PROGRAM NAME   FORTRAN PH-01
5  *
6  *   MODEL NUMBER   8F001
7  *
8  *   PURPOSE        FORTRAN INPUT PHASE
9  *
10 *   PROGRAMMER     DICK WALLMANN, MODS-MARK ELFIELD
11 *
12 * ***** REVISION LIST *****
13 *
14 *   RV DATE       SCO   BY   REASON FOR CHANGE
15 *   --  - - - - - - - - - - - - - - - - - - - -
16 *
17 *   01 11/16/70   RPH  INTIAL RELEASE
18 *
19 *****
20 *****
21 * GA 18/30 FORTRAN COMPILER 05/01/70
22 * STATUS - VERSION 1, MODIFICATION 0
23 * FUNCTION/OPERATION
24 * * READS THE CONTROL RECORDS, AND SETS
25 *   CORRESPONDING INDICATORS IN THE FORTRAN
26 *   COMMUNICATIONS AREA (FCOM).
27 * * READS THE SOURCE STATEMENTS AND STORES THEM
28 *   IN THE STRING AREA, EACH STATEMENT IS
29 *   PRECEDED WITH A PARTIALLY COMPLETED ID WORD.
30 * * CHECKS FOR A MAXIMUM OF FIVE CONTINUATION
31 *   RECORDS PER STATEMENT.
32 * * LISTS THE SOURCE PROGRAM IF REQUIRED.
33 * ENTRY POINTS-
34 * * FZZ50-ENTERED BY A CALL FROM THE SUPERVISOR
35 * INPUT-
36 * * INPUT IS VIA CONTROL AND SOURCE STATEMENTS
37 *   ENTERED THROUGH LOGICAL UNIT 1
38 * OUTPUT-
39 * * LISTING ON LOGICAL UNIT 5
40 *   CONTROL RECORDS, AND OF ALL SOURCE
41 *   STATEMENTS IF SO REQUESTED.
42 * * INFORMATION FOR THE OTHER PHASES DEALING
43 *   WITH THE CURRENT COMPILATION STORED IN FCOM.
44 * * SOURCE STATEMENTS WITH PARTIALLY COMPLETED
45 *   IDS STORED IN THE STRING AREA.
46 * EXTERNAL REFERENCES-
47 * * SUBROUTINES = LIO
48 * * COMMUNICATIONS AREAS = FIXED AREA
49 * * OTHER FORTRAN PHASES = NONE
50 * EXITS-
51 *   NORMAL-
52 * * EXIT IS MADE THROUGH THE INTERPHASE
53 *   ROLLER SUBROUTINE WITH A REQUEST TO ENTER
54 *   PHASE 2. THIS EXIT IS MADE AFTER THE END
55 *   STATEMENT IS ENCOUNTERED.
56 *   ERROR-
57 * * // CONTROL CARD ENCOUNTERED-
58 *   EXIT IS MADE THROUGH THE INTERPHASE
59 *   ROLLER SUBROUTINE WITH A REQUEST TO ENTER

```

```

60 *          THE RECOVERY PHASE 28,
61 * TABLES/WORK AREAS-
62 *   * FCOM
63 *   SOFS - START OF STRING
64 *   EOFS - END OF STRING
65 *   SOFST- START OF SYMBOL TABLE
66 *   SOFNS- START OF NON-STATEMENT NUMBERS
67 *   SOFXT- START OF SUBSCRIBED TEMPORARIES
68 *   SOFGT- START OF GENERATED TEMPORARIES
69 *   EOFST- END OF SYMBOL TABLE
70 *   COMON- NEXT AVAILABLE COMMON
71 *   CSIZE- SIZE OF COMMON
72 *   ERROR- OVERLAP ERROR SWITCH
73 *   FNAME- PROGRAM NAME (2 WORDS)
74 *   SORF - SUBROUTINE(-) OR FUNCTION(+)
75 *   CCWD - CONTROL CARD WORD
76 *   BIT 15 TRANSFER TRACE
77 *   BIT 14 ARITHMETIC TRACE
78 *   BIT 13 EXTENDED PRECISION
79 *   BIT 12 LIST SYMBOL TABLE
80 *   BIT 11 LIST SUBPROGRAM NAMES
81 *   BIT 10 LIST SOURCE PROGRAM
82 *   BIT 9 ONE WORD INTEGERS
83 *   BIT 8 PUNCH
84 *   BIT 7 NONPROCESS PROGRAM
85 *   BIT 6 NOT USED
86 *   BIT 5 NOT USED
87 *   BIT 4 NOT USED
88 *   BIT 3 NOT USED
89 *   BIT 2 NOT USED
90 *   BIT 1 NOT USED
91 *   BIT 0 NOT USED
92 *   IOCS - IOCS CONTROL CARD WORD
93 *   BIT 15 CARD
94 *   BIT 14 PAPER TAPE
95 *   BIT 13 TYPEWRITER
96 *   BIT 12 1443 PRINTER
97 *   BIT 11 MAGNETIC TAPE
98 *   BIT 10 KEYBOARD
99 *   BIT 9 NOT USED
100 *   BIT 8 DISK
101 *   BIT 7 NOT USED
102 *   BIT 6 NOT USED
103 *   BIT 5 NOT USED
104 *   BIT 4 NOT USED
105 *   BIT 3 PLOTTER
106 *   BIT 1 UNFORMATTED DISK
107 *   BIT 0 UNFORMATTED TAPE
108 *   DFCNT- DEFINE FILE COUNT
109 *   LCOMN - TWO WORDS FOR INSKEL COMMON
110 *   ICCER - IOCS CONTROL CARD ERROR WORD
111 * * PROGRAMMED SWITCHES-
112 *   THE SWITCHES USED IN PHASE 1 FOLLOW-
113 *   IF NON-ZERO, THE SWITCH IS TRANSFER = T
114 *   IF ZERO, THE SWITCH IS NORMAL = N
115 *   * SWITCH 1- FZ951
116 *   N= SOURCE CARD NOT YET ENCOUNTERED
117 *   T= SOURCE CARD ENCOUNTERED
118 *   * SWITCH 2- FZ952
119 *   N= CONTINUATION CARD NOT ALLOWED

```

```

120 *          T= CONTINUATION CARD OK
121 *      * SWITCH 3= FZ953
122 *          N= NO NORM CALCULATION REQUIRED
123 *          T= STMT REQUIRES NORM CALCULATION
124 *      * SWITCH 4= FZ954
125 *          N= NORMAL
126 *          T= EXCESS CONTINUATION RECORDS
127 *      * SWITCH 5= FZ955
128 *          N= NORMAL
129 *          T= STMT NOT POSSIBLY A FORMAT STMT
130 *      * SWITCH 6= FZ956
131 *          N= NORMAL
132 *          T= FORMAT STMT, DON'T REMOVE BLANKS
133 *      * SWITCH 7= FZ957
134 *          N= STORE CHAR IN LEFT OF WORD
135 *          T= STORE CHAR IN RIGHT OF WORD
136 *      * SWITCH 9= FZ959
137 *          N= NORMAL
138 *          T= DATA STATEMENT BEING PROCESSED
139 *      * SWITCH 14= FZ95E
140 *          N= END OF CTRL CARD NOT ANTICIPATED
141 *          T= NORMAL
142 *      * STRING AREA
143 *          THE STRING AREA DURING COMPILATION CONTAINS
144 *          BOTH THE STATEMENT STRING AND THE SYMBOL
145 *          TABLE. THE STATEMENT STRING IS BUILT BY
146 *          PHASE 1 IN AN ASCENDING CHAIN BEGINNING IN
147 *          THE LOW-ADDRESSED WORDS OF THE STRING
148 *          AREA. THE SYMBOL TABLE IS BUILT DURING
149 *          THE COMPILATION PROCESS IN THE HIGH-
150 *          ADDRESSED WORDS OF THE STRING AREA (THE
151 *          BEGINNING OF ROLRX- 5). THE ACTUAL START
152 *          AND END ADDRESSES OF THE STRING AREA MAY BE
153 *          FOUND IN ENTRIES IN FCOM.
154 *      * HEAD
155 *          START OF A 49 WORD AREA CONTAINING THE PAGE
156 *          HEADING FOR LISTINGS.
157 *      * ROLRX
158 *          START OF A 50 WORD AREA CONTAINING THE
159 *          FORTRAN INTERPHASE ROLLER SUBROUTINE.
160 *      * NOTE
161 *          THE AREAS HEAD, ROLRX, PHNTB AND FCOM ARE
162 *          NOT OVERLAYED BY ANY SUBSEQUENT PHASE
163 *          UNTIL PHASE 28, RECOVERY.
164 *ATTRIBUTES-
165 *      * ERRORS DETECTED BY THIS PHASE ARE NUMBERS
166 *          1 AND 2 AND UTAPE/UDISK BOTH SPECIFIED.
167 *NOTES-N/A
168 *          ABS
169 *          THE FOLLOWING PROGRAM (FZXXX) IS IN
170 *          CORE ONLY UNTIL THE CONTROL CARDS
171 *          ARE ANALYZED.
172 *          SYSTEM EQUATES
173 *          REF LIO
174 *          REF ENDMN
175 *          REF BULKA
176 *          REF CORE
177 *          REF FSOVL
178 *          PHASE 1 EQUATES
179 MEMRY EQU /ZFFF MAXIMUM CORE SIZE

```

180	PHSIZ	EQU	4*320	MAXIMUM PHASE SIZE
181	OVERL	EQU	MEMRY-PHSIZ	PHASES 2-29 START
182	FCOM	EQU	OVERL-22	FORTRAN COMM. TABLE
183	PHNTB	EQU	FCOM-56	PHASE TABLE
184	ROLRX	EQU	PHNTB-50	INTERPHASE CALL
185		ORG	ROLRX-2*320	
186	FZZ50	LDX	I3 CORE	XR3=ACTUAL CORE SIZE
187		MDX	L3 ROLRX-78004	SET SYMBOL TABLE END
188		NOP		PREVIOUS INST SKIPS IF 32
189		STX	L3 FX003	START OF SYMBOL TABLE
190		STX	L3 FX004	START OF NON-STATEMENT VO
191		STX	L3 FX005	START OF SUBSCRIPT TEMPS.
192		STX	L3 FX006	START OF GENERATED TEMPS.
193		STX	L3 FX007	END OF SYMBOL TABLE
194		LDX	L3 /FFFF	SET ADDRESS OF
195		STX	L3 FX008	NEXT AVAILABLE COMMON
196		LDX	L3 /80BF	
197		STX	L3 FX014	SET IOCS(CARD)
198		SLA	16	SET THE FOLLOWING TO ZERO
199		STO	L FX009	SIZE OF COMMON
200		STO	L FX010	OVERLAP ERROR SWITCH
201		STO	L FX012	SUBROUTINE OR FUNCTION SW
202		STO	L FX013	CONTROL CARD SWITCH WORD
203		STO	L FX011	PROGRAM NAME
204		STO	L FX011+1	2ND WORD OF NAME
205		STO	L FX015	FILE COUNT
206		STO	L F1951	NORMALIZE SWITCH 1
207		LDX	I1 ENDMN	
208		STX	L1 SOFS	STRING BASE
209	*			OPEN INPUT AND PRINT FILES
210		BSI	L LIO	
211		DC	/5001	
212		DC	CAREA	
213		DC	0	
214		BSI	L LIO	
215		DC	/5005	
216		DC	CAREB	
217		DC	0	
218		BSC	L F1000	
219	*			CONSTANTS AND SWITCHES
220	FZ95E	DC	0	SWITCH 14
221	FZ911	DC	.T	T TEST CONSTANT
222	FZ912	DC	.A-,T	A TEST
223	FZ913	DC	.L-,A	L TEST
224	FZ914	DC	.E-,L	E TEST
225	FZ915	DC	.I-,E	I TEST
226	FZ916	DC	.O-,I	O TEST
227	FZ918	DC	.*--,O	* TEST
228	FZ922	DC	.N-,*	N TEST
229	FZ91A	DC	.P-,N	P TEST
230	FZ919	DC	.-	- FOR CONTROL CARD ERROR
231	FZ920	DC	0	LENGTH FOR CTL CARD COMP
232	FZ921	DC	*	PAGE RESTORE SWITCH
233	FZ003	EQU	*	
234	FZ004	SLA	16	
235		STO	FZ95E	NORMALIZE SWITCH 14
236		BSI	L FZ600	GET FIRST NON-BLANK CHAR
237		S	FZ911	IS THE CHARACTER T
238		BSC	L FZ008,+-	BRANCH IF YES
239		S	FZ912	IS THE CHARACTER A

240	BSC	L	FZ012,+-	BRANCH IF YES
241	S		FZ913	IS THE CHARACTER L
242	BSC	L	FZ014,+-	BRANCH IF YES
243	S		FZ914	IS THE CHARACTER E
244	BSC	L	FZ022,+-	BRANCH IF YES
245	S		FZ915	IS THE CHARACTER I
246	BSC	L	FZ024,+-	BRANCH IF YES
247	S		FZ916	IS THE CHARACTER O
248	BSC	L	FZ052,+-	BRANCH IF YES
249	S		FZ918	IS THE CHARACTER *
250	BSC	L	F1003,+-	BRANCH IF YES
251	S		FZ922	IS CHARACTER N
252	BSC	L	FZ100,+-	BRANCH IF YES
253	S		FZ91A	IS CHARACTER P
254	BSC	L	FZ021,+-	BRANCH IF YES
255	*			BAD CONTROL CARD ENTRY
256	FZ005	LD	FZ919	LOAD - ERROR INDICATOR
257		STO	I F1003+1	PLACE MINUS INTO CC RECDR
258	*			PRINT CARD IN ERROR AND
259		BSC	L F1003	GET ANOTHER,
260	FZ008	LDX	3 12	SET CHARACTER COUNT
261		STX	3 FZ920	TO 12
262		LDX	L3 FZ971	*TRANSFERTRACE*
263	FZ010	BSC	L FZ056	GO CHECK IF VALID CONTROL
264	FZ012	LDX	3 14	SET CHARACTER
265		STX	3 FZ920	COUNT TO 14
266		LDX	L3 FZ972	*ARITHMETICTRACE*
267		MDX	FZ010	GO CHECK
268	FZ014	LDX	3 3	SET CHARACTER
269		STX	3 FZ920	COUNT TO 3
270		LDX	L3 FZ981	*ISI*
271		BSI	L FZ060	GO CHECK, RETURN IF EQUAL
272		BSI	L FZ600	GET NEXT NON-BLANK CHAR
273		S	FZ923	IS CHARACTER S
274		BSC	L FZ020,Z	BRANCH IF NO
275		BSI	L FZ600	GET NEXT NON-BLANK CHAR
276		S	FZ924	IS CHARACTER Y
277		BSC	L FZ016,Z	BRANCH IF NO
278		LDX	3 9	SET CHARACTER
279		STX	3 FZ920	COUNT TO 9
280		LDX	L3 FZ982	*MBULTABLE*
281		MDX	FZ010	GO CHECK
282	FZ016	S	FZ925	IS CHARACTER O
283		BSC	L FZ018,Z	BRANCH IF NO
284		LDX	3 11	SET CHARACTER
285		STX	3 FZ920	COUNT TO 11
286		LDX	L3 FZ983	*URCEPRUGRAM*
287		MDX	FZ010	GO CHECK
288	FZ018	S	FZ926	IS CHARACTER U
289		BSC	L FZ005,Z	BRANCH TO ERROR IF NO,
290		LDX	3 13	SET CHARACTER
291		STX	3 FZ920	COUNT TO 13
292		LDX	L3 FZ984	*BPROGRAMNAMES*
293		MDX	FZ010	GO CHECK
294	FZ020	S	FZ927	IS CHARACTER A
295		BSC	L FZ005,Z	BRANCH TO ERROR IF NO
296		LDX	3 2	SET CHARACTER
297		STX	3 FZ920	COUNT TO 2
298		LDX	L3 FZ985	*LL*
299		MDX	FZ010	GO CHECK

300	FZ021	LDX	3	4	
301		STX	3	FZ920	
302		LDX	L3	FZ97B	
303		MDX		FZ010	BR TO A BRANCH
304	FZ022	LDX	3	16	SET CHARACTER
305		STX	3	FZ920	COUNT TO 16
306		LDX	L3	FZ974	*XTENDEDPRECISION*
307		MDX		FZ010	GO CHECK
308	*			CONSTANTS	
309	FZ923	DC		.S	S TEST
310	FZ924	DC		.Y	Y TEST
311	FZ925	DC		.O-,Y	O TEST
312	FZ926	DC		.U-,O	U TEST
313	FZ927	DC		.A-,S	A TEST
314	FZ928	DC		/4D	LEFT PAREN TEST
315	FZ930	DC		0	TEMPORARY I/O DEVICE SW
316	FZ931	DC		.C	C TEST
317	FZ932	DC		.P-,C	P TEST
318	FZ933	DC		.T-,P	T TEST
319	FZ934	DC		.K-,T	K TEST
320	FZ935	DC		.1-,K	1 TEST
321	FZ936	DC		.D-,1	D TEST
322	FZ93U	DC		.U-,D	U TEST
323	FZ937	DC		.M-,U	M TEST
324	FZ938	DC		/006B	COMMA TEST
325	FZ939	DC		/5D-/6B	RIGHT PAREN TEST
326	FZ941	DC		.L	L TEST
327	FZ942	DC		.A-,L	A TEST
328	FZ024	LDX	3	3	SET CHARACTER
329		STX	L3	FZ920	COUNT TO 3
330		LDX	L3	FZ975	*OCS*
331		BSI	L	FZ060	GO CHECK, RETURN IF VALID
332	FZ026	BSI	L	FZ600	GET NEXT NON-BLANK CHAR
333		S		FZ928	IS CHARACTER LFT PAREN
334		BSC	L	FZ005,Z	BRANCH TO ERROR IF NOT YE
335		SLA		16	
336		STO	L	FZ95E	ZERO SWITCH 14
337		STO		FZ930	CLEAR TEMP I/O DEVICE SW
338	FZ028	BSI	L	FZ600	GET NEXT NON-BLANK CHAR
339		S		FZ931	IS CHARACTER C
340		BSC	L	FZ036,+	BRANCH IF YES
341		S		FZ932	IS CHARACTER P
342		BSC	L	FZ038,+	BRANCH IF YES
343		S		FZ933	IS CHARACTER T
344		BSC	L	FZ040,+	BRANCH IF YES
345		S		FZ934	IS CHARACTER K
346		BSC	L	FZ042,+	BRANCH IF YES
347		S		FZ935	IS CHARACTER 1
348		BSC	L	FZ044,+	BRANCH IF YES
349		S		FZ936	IS CHARACTER D
350		BSC	L	FZ046,+	BRANCH IF YES
351		S		FZ93U	IS CHARACTER U
352		BSC	L	FZ053,+	BRANCH IF YES
353		S		FZ937	IS CHARACTER M
354		BSC	L	FZ005,Z	BRANCH TO ERROR IF NO
355		LDX	3	11	SET CHARACTER COUNT
356		STX	L3	FZ920	COUNT TO 11
357		LDX	L3	FZ97F	AGNETIC TAPE
358	FZ032	BSI	L	FZ060	GO CHECK, RETURN IF VALID
359	FZ034	LD		FZ930	INDICATE REQUESTED

360		OR	3 0	*I/O DEVICE
361		STO	FZ930	
362		BSI	L FZ600	GET NEXT NON-BLANK CHAR
363		S	FZ938	IS CHARACTER COMMA
364		BSC	L FZ028,+-	IF YES, CHECK NEXT DEVICE
365		S	FZ939	IS CHARACTER RIGHT PAREV
366		BSC	L FZ005,Z	BRANCH TO ERROR IF NO
367		LD	L FX014	LOAD IOCS WORD
368		OR	FZ930	ADD DEVICES FROM THIS CAR
369		STO	L FX014	STORE INTO IOCS WORD
370		BSC	L F1003	PRINT CARD AND GET ANOTHE
371	FZ036	LDX	3 3	SET CHARACTER
372		STX	L3 FZ920	COUNT TO 3
373		LDX	L3 FZ976	*ARD*
374		MDX	FZ032	GO CHECK
375	*			PAPER TAPE OR PLOTTER
376	FZ038	BSI	L FZ600	GET NEXT NON-BLANK CHAR
377		S	FZ941	IS CHARACTER L
378		BSC	L FZ048,+-	BRANCH IF YES
379		S	FZ942	IS CHARACTER A
380		BSC	L FZ005,Z	BRANCH TO ERROR IF NO
381	FZ039	LDX	3 7	SET CHARACTER
382		STX	L3 FZ920	COUNT TO 7
383		LDX	L3 FZ978	*PERTAPE*
384		MDX	FZ032	GO CHECK
385	FZ040	LDX	3 9	SET CHARACTER
386		STX	L3 FZ920	COUNT TO 9
387		LDX	L3 FZ979	*YPEWRITER*
388		MDX	FZ032	GO CHECK
389	FZ042	LDX	3 7	SET CHARACTER
390		STX	L3 FZ920	COUNT TO 7
391		LDX	L3 FZ97A	*EYBOARD*
392		MDX	FZ032	GO CHECK
393	FZ044	LDX	3 10	SET CHARACTER
394		STX	L3 FZ920	COUNT TO 10
395		LDX	L3 FZ97E	*44S PRINTER*
396		MDX	FZ032	GO CHECK
397	*			A REGISTER = D
398	FZ046	LDX	3 3	SET CHARACTER
399		STX	L3 FZ920	COUNT TO 3
400		LDX	L3 FZ97C	*ISK*
401		MDX	FZ032	GO CHECK
402	FZ048	LDX	3 5	SET CHARACTER
403		STX	L3 FZ920	COUNT TO 5
404		LDX	L3 FZ97D	*OTIER*
405		MDX	FZ032	GO CHECK
406	FZ052	LDX	3 14	SET CHARACTER
407		STX	L3 FZ920	COUNT = 14
408		LDX	L3 FZ986	*NEWORDINTEGERS*
409		MDX	FZ056	GO CHECK
410	FZ053	BSI	L FZ600	GET NEXT NON BLANK CHARTE
411		S	FZ943	IS CHARACTER D
412		BSC	L FZ055,+-	BRANCH IF YES
413		S	FZ944	IS CHARACTER T
414		BSC	L FZ005,Z	BRANCH IF NO
415	FZ054	LDX	3 3	SET CHARACTER
416		STX	L3 FZ920	COUNT TO 3
417		LDX	L3 FZ977	UTAPE
418		MDX	FZ032	GO CHECK
419	FZ055	LDX	3 3	SET CHARACTER

```

420          STX  L3 FZ920      COUNT TO 3
421          LDX  L3 FZ988      UDISK TABLE
422          MDX          FZ032      GO CHECK IT
423 FZ056 BSI          FZ060      GO CHECK, RETURN IF EQUAL
424          MDX  L FZ95E,1      TAG SWITCH 14
425          BSI          FZ600      GET NEXT NON-BLANK CHAR
426 FZ058 BSC  L FZ005      BRANCH TO ERROR IF RETURN
427 *                                COMPARE CHARACTERS FOR EQUAL
428 *                                RETURN IF ALL CHARACTERS EQUAL,
429 *                                GO TO ERROR IF NOT
430 FZ060 DC          0          BSI ADDRESS
431          BSI          FZ600      GET NEXT NON-BLANK CHAR
432          S          3 0          IS CHARACTER CORRECT
433          BSC  L FZ005,Z      BRANCH IF NOT
434          MDX  3 1          INCREMENT FOR NEXT COMPAR
435          MDX  L FZ920,-1      DECREMENT CHARACTER COUNT
436          MDX          FZ060+1    CHECK NEXT CHARACTER
437          BSC  I FZ060      RETURN, COMPARE COMPLETE
438 *                                CONSTANTS
439 FZ943 DC          .D          D TEST
440 FZ944 DC          .T-,D      T TEST
441 FZ945 DC          71         TEST FOR LAST COLUMN
442 FZ946 DC          0          WORK WORD
443 FZ947 DC          /0040      EBC BLANK
444 *                                THIS SUBROUTINE PICKS UP NON-BLANK
445 *                                COLUMNS FROM THE INPUT CARD, ENTRY
446 *                                IS A BSI INSTRUCTION TO THE LABEL
447 *                                (FZ600), UPON EXITING FROM THE
448 *                                ROUTINE THE A REGISTER WILL CONTAIN
449 *                                THE CHARACTER IN UNPACKED EBC CODE,
450 FZ600 DC          0          BSI ENTRY
451          STX  2 FZ946      XR2 = POINTER IN CARD B/J
452          LD   FZ946      LOAD POINTER
453          S   FZ945      CHECK IF
454          S   L F1003+1    END OF CARD,
455          BSC  L FZ601,+Z    BRANCH IF NO
456          LD   L FZ95E      LOAD SWITCH 14
457          BSC  L FZ005,+ -  BRANCH TO ERROR IF NOT ON
458          LD   3 0          SET CURRENT CONTROL OPTIO
459          OR   L FX013      INTO CONTROL CARD
460          STO  L FX013      WORD,
461          BSC  L F1003      PRINT AND GET NEXT CARD
462 FZ601 LD   2 1          LOAD NEXT CHARACTER IN CD
463          MDX  2 1          INCREMENT POINTER
464          STO  FZ946      SAVE CHAR
465          EOR  FZ947      CHECK IF CHARACTER BLANK
466          BSC  L FZ600+1,+ -  BRANCH IF BLANK
467          LD   FZ946      LOAD CHARACTER
468          BSC  I FZ600      RETURN
469 ***** CONTROL CARD TABLES *****
470 *                                TRANSFER TRACE
471 FZ971 DC          .R          R
472          DC          .A          A
473          DC          .I          N
474          DC          .S          S
475          DC          .F          F
476          DC          .E          E
477          DC          .R          R
478          DC          .T          T
479          DC          .R          R

```

480		DC	.A	A
481		DC	.C	C
482		DC	.E	E
483		DC	/0001	INDICATOR
484	*			ARITHMETIC TRACE
485	FZ972	DC	.R	R
486		DC	.I	I
487		DC	.T	T
488		DC	.H	
489		DC	.M	
490		DC	.E	
491		DC	.T	T
492		DC	.I	I
493		DC	.C	C
494		DC	.T	T
495		DC	.R	R
496		DC	.A	A
497		DC	.C	C
498		DC	.E	E
499		DC	/0002	INDICATOR
500	*			IOCS
501	FZ973	DC	.O	O
502		DC	.C	C
503		DC	.S	S
504	*			EXTENDED PRECISION
505	FZ974	DC	.X	X
506		DC	.T	T
507		DC	.E	E
508		DC	.I	I
509		DC	.D	D
510		DC	.E	E
511		DC	.D	D
512		DC	.P	P
513		DC	.R	R
514		DC	.E	E
515		DC	.C	C
516		DC	.I	I
517		DC	.S	S
518		DC	.I	I
519		DC	.O	O
520		DC	.I	I
521		DC	/0004	INDICATOR
522	*			CARD
523	FZ976	DC	.A	A
524		DC	.R	R
525		DC	.D	D
526		DC	/0001	INDICATOR
527	*			TAPE
528	FZ977	DC	.A	A
529		DC	.P	P
530		DC	.E	E
531		DC	/0000	INDICATOR
532	*			PAPER TAPE
533	FZ978	DC	.P	P
534		DC	.E	E
535		DC	.R	R
536		DC	.T	T
537		DC	.A	A
538		DC	.P	P
539		DC	.E	E

540		DC	/0002	INDICATOR
541	*			TYPEWRITER
542	FZ979	DC	.Y	Y
543		DC	.P	P
544		DC	.E	E
545		DC	.W	W
546		DC	.R	R
547		DC	.I	I
548		DC	.T	T
549		DC	.E	E
550		DC	.R	R
551		DC	/0004	INDICATOR
552	*			KEYBOARD
553	FZ97A	DC	.E	E
554		DC	.Y	Y
555		DC	.B	B
556		DC	.O	O
557		DC	.A	A
558		DC	.R	R
559		DC	.D	D
560		DC	/0020	INDICATOR
561	*			PUNCH
562	FZ97B	DC	.U	U
563		DC	.N	N
564		DC	.C	C
565		DC	.H	H
566		DC	/0080	INDICATOR
567	*			DISK
568	FZ97C	DC	.I	I
569		DC	.S	S
570		DC	.K	K
571		DC	/0080	INDICATOR
572	*			PLOITER
573	FZ97D	DC	.O	O
574		DC	.T	T
575		DC	.T	T
576		DC	.E	E
577		DC	.R	R
578		DC	/1000	INDICATOR
579	*			1443 PRINTER
580	FZ97E	DC	.4	4
581		DC	.4	4
582		DC	.3	3
583		DC	.P	P
584		DC	.R	R
585		DC	.I	I
586		DC	.N	N
587		DC	.T	T
588		DC	.E	E
589		DC	.R	R
590		DC	/0008	INDICATOR
591	*			MAGNETIC TAPE
592	FZ97F	DC	.A	A
593		DC	.G	G
594		DC	.I	I
595		DC	.E	E
596		DC	.T	T
597		DC	.I	I
598		DC	.C	C
599		DC	.T	T

600		DC	.A	A
601		DC	.P	P
602		DC	.E	E
603		DC	/0010	INDICATOR
604	*			LIST
605	FZ981	DC	.I	I
606		DC	.S	S
607		DC	.T	T
608	*			SYMBOL TABLE
609	FZ982	DC	.I	M
610		DC	.B	B
611		DC	.O	O
612		DC	.L	L
613		DC	.T	T
614		DC	.A	A
615		DC	.B	B
616		DC	.L	L
617		DC	.E	E
618		DC	/0008	INDICATOR
619	*			SOURCE PROGRAM
620	FZ983	DC	.J	U
621		DC	.R	R
622		DC	.C	C
623		DC	.E	E
624		DC	.P	P
625		DC	.R	R
626		DC	.O	O
627		DC	.S	S
628		DC	.R	R
629		DC	.A	A
630		DC	.M	M
631		DC	/0020	INDICATOR
632	*			SUBPROGRAM NAMES
633	FZ984	DC	.B	B
634		DC	.P	P
635		DC	.R	R
636		DC	.O	O
637		DC	.G	G
638		DC	.R	R
639		DC	.A	A
640		DC	.M	M
641		DC	.N	N
642		DC	.A	A
643		DC	.I	I
644		DC	.E	E
645		DC	.S	S
646		DC	/0010	INDICATOR
647	*			ALL
648	FZ985	DC	.L	L
649		DC	.L	L
650		DC	/0038	INDICATOR
651	*			ONE WORD INTEGERS
652	FZ986	DC	.I	N
653		DC	.E	E
654		DC	.W	W
655		DC	.O	O
656		DC	.R	R
657		DC	.D	D
658		DC	.I	I
659		DC	.N	N

660	DC	.T	T
661	DC	.E	E
662	DC	.G	G
663	DC	.E	E
664	DC	.R	R
665	DC	.S	S
666	DC	/0040	INDICATOR
667	*		NONPROCESS PROGRAM
668	FZ987	DC	.O
669	DC	.N	N
670	DC	.P	P
671	DC	.R	R
672	DC	.O	O
673	DC	.C	C
674	DC	.E	E
675	DC	.S	S
676	DC	.S	S
677	DC	.P	P
678	DC	.R	R
679	DC	.O	O
680	DC	.G	G
681	DC	.R	R
682	DC	.A	A
683	DC	.M	M
684	DC	/0100	INDICATOR
685	*		UDISK
686	FZ988	DC	.I
687	DC	.S	S
688	DC	.K	K
689	DC	/4000	INDICATOR
690	*		PROCESS *NAME CARD
691	FZ100	LDX	3 16 SET CHARACTER
692		STX	L3 FZ920 COUNT TO 16
693		LDX	L3 FZ987 *NONPROCESSPROGRAM
694		BSC	L FZ056 BRANCH TO A BRANCH
695	*		END OF INITIALIZATION AND CONTROL
696	*		CARD PROCESSING SECTION OF PHASE 1,
697	*		BEGINNING OF MAIN BODY OF PHASE 1
698	*		THIS SECTION LIES IN HIGH
699	*		ER CORE THAN DOES THE
700	*		SYMBL TABL AND STRNG AREA
701	*		THE ABOVE SECTION WILL BE OVERLAYED
702	*		WITH SYMBOL TABLE ENTRIES IN
703	*		SUBSEQUENT PHASES,
704	*	ROLRX	ROUTINE - READS REQUIRED OVERLAY
705	*		CALLING SEQUENCE BSI L ROLRX
706	*		DC PHASE-NUMBER
707	*		ALL PHASES ARE READ INTO AND
708	*		START AT OVERL.
709		ORG	ROLRX EXECUTION ADDR
710		DC	** LINK WORD TO CALLING POINT
711		LD	I ROLRX GET PHASE NUMBER
712		STO	PHCNT
713		SLA	16 INTERPHASE DUMP DISENABLED
714	*		TO ENABLE THE INTERPHASE
715	*		DUMP ROUTINE, PATCH THE
716	*		1010, (SLA 16), INSTRUCT
717	*		ION TO 080F, (XIO PHCNT
718		S	PHCNT SUBTRACT PHASE COUNT FROM
719	*		BIT SW DATA READ WITH XIO

720	BSC	L	RETRN,Z	GO CALL NEXT PHASE ON +OR-
721	LDX	1	2*29	GET INTERPHASE DUMP
722	BSC	L	PHSRD	DO INTERPHASE DUMP
723	RETRN	LDX	I1 PHCNT	INTERPHASE DUMP RETRN
724		MDX	I1 PHCNT	FORM PHNTB POINTER
725	BSC	L	PHSRD	DO REQUIRED PHASE
726	BSS	E		FORCE I-COUNTER EVEN
727	PHCNT	DC	*-*	CURRENT PHASE NUMBER
728		DC	/0740	CONT WD TO READ BIT SWS
729	*			INTU THE ACCUMULATOR
730	PHSRD	MDX	L1 PHNTB-4	FORM TABLE ADDRESS
731		LD	1 0	DISK SECTOR
732	A	L	F\$OVL	BASE SECTOR ADDRESS
733	STO	L	OVERL-1	
734		LD	1 1	WORD COUNT
735	STO	L	OVERL-2	
736	BSI	I	BULK	READ NEXT PHASE
737		DC	DISKL	
738		LD	DISKL	
739	BSC	L	*-3,Z	LOOP UNTILL DONE
740	BSC	L	OVERL	TO PHASE START
741	DISKL	DC	0	DISK CONTROL LIST
742		DC	0	
743		BSS	4	
744		DC	0	
745		DC	/1000	READ OPERATION
746		DC	OVERL-2	
747	*	OVERLAY	SEGEMENT	SIZE EQUATES
748	FR02	EQU	3	
749	FR03	EQU	2	
750	FR04	EQU	3	
751	FR05	EQU	4	
752	FR06	EQU	3	
753	FR07	EQU	2	
754	FR08	EQU	3	
755	FR09	EQU	4	
756	FR10	EQU	3	
757	FR11	EQU	3	
758	FR12	EQU	3	
759	FR13	EQU	3	
760	FR14	EQU	4	
761	FR15	EQU	4	
762	FR16	EQU	4	
763	FR17	EQU	4	
764	FR18	EQU	4	
765	FR19	EQU	0	
766	FR20	EQU	4	
767	FR21	EQU	3	
768	FR22	EQU	3	
769	FR23	EQU	2	
770	FR24	EQU	3	
771	FR25	EQU	3	
772	FR26	EQU	4	
773	FR27	EQU	3	
774	FR28	EQU	2	
775	FR29	EQU	2	
776	*	RELATIVE POSITION	EQUATES	
777	FRX02	EQU	0	
778	FRX03	EQU	FRX02+FR02	
779	FRX04	EQU	FRX03+FR03	

780	FRX05	EQU	FRX04+FR04
781	FRX06	EQU	FRX05+FR05
782	FRX07	EQU	FRX06+FR06
783	FRX08	EQU	FRX07+FR07
784	FRX09	EQU	FRX08+FR08
785	FRX10	EQU	FRX09+FR09
786	FRX11	EQU	FRX10+FR10
787	FRX12	EQU	FRX11+FR11
788	FRX13	EQU	FRX12+FR12
789	FRX14	EQU	FRX13+FR13
790	FRX15	EQU	FRX14+FR14
791	FRX16	EQU	FRX15+FR15
792	FRX17	EQU	FRX16+FR16
793	FRX18	EQU	FRX17+FR17
794	FRX19	EQU	FRX18+FR18
795	FRX20	EQU	FRX19+FR19
796	FRX21	EQU	FRX20+FR20
797	FRX22	EQU	FRX21+FR21
798	FRX23	EQU	FRX22+FR22
799	FRX24	EQU	FRX23+FR23
800	FRX25	EQU	FRX24+FR24
801	FRX26	EQU	FRX25+FR25
802	FRX27	EQU	FRX26+FR26
803	FRX28	EQU	FRX27+FR27
804	FRX29	EQU	FRX28+FR28
805	*****	MPX	FORTRAN PHASE NAME TABLE*****
806		ORG	PHNTB
807		DC	FRX02 PHASE=02 DISK SECTOR
808		DC	320*FR02 WORD COUNT
809		DC	FRX03 PHASE=03 DISK SECTOR
810		DC	320*FR03 WORD COUNT
811		DC	FRX04 PHASE=04 DISK SECTOR
812		DC	320*FR04 WORD COUNT
813		DC	FRX05 PHASE=05 DISK SECTOR
814		DC	320*FR05 WORD COUNT
815		DC	FRX06 PHASE=06 DISK SECTOR
816		DC	320*FR06 WORD COUNT
817		DC	FRX07 PHASE=07 DISK SECTOR
818		DC	320*FR07 WORD COUNT
819		DC	FRX08 PHASE=08 DISK SECTOR
820		DC	320*FR08 WORD COUNT
821		DC	FRX09 PHASE=09 DISK SECTOR
822		DC	320*FR09 WORD COUNT
823		DC	FRX10 PHASE=10 DISK SECTOR
824		DC	320*FR10 WORD COUNT
825		DC	FRX11 PHASE=11 DISK SECTOR
826		DC	320*FR11 WORD COUNT
827		DC	FRX12 PHASE=12 DISK SECTOR
828		DC	320*FR12 WORD COUNT
829		DC	FRX13 PHASE=13 DISK SECTOR
830		DC	320*FR13 WORD COUNT
831		DC	FRX14 PHASE=14 DISK SECTOR
832		DC	320*FR14 WORD COUNT
833		DC	FRX15 PHASE=15 DISK SECTOR
834		DC	320*FR15 WORD COUNT
835		DC	FRX16 PHASE=16 DISK SECTOR
836		DC	320*FR16 WORD COUNT
837		DC	FRX17 PHASE=17 DISK SECTOR
838		DC	320*FR17 WORD COUNT
839		DC	FRX18 PHASE=18 DISK SECTOR

840	DC	320*FR18	WORD COUNT
841	DC	FRX19	PHASE=19 DISK SECTOR
842	DC	320*FR19	WORD COUNT
843	DC	FRX20	PHASE=20 DISK SECTOR
844	DC	320*FR20	WORD COUNT
845	DC	FRX21	PHASE=21 DISK SECTOR
846	DC	320*FR21	WORD COUNT
847	DC	FRX22	PHASE=22 DISK SECTOR
848	DC	320*FR22	WORD COUNT
849	DC	FRX23	PHASE=23 DISK SECTOR
850	DC	320*FR23	WORD COUNT
851	DC	FRX24	PHASE=24 DISK SECTOR
852	DC	320*FR24	WORD COUNT
853	DC	FRX25	PHASE=25 DISK SECTOR
854	DC	320*FR25	WORD COUNT
855	DC	FRX26	PHASE=26 DISK SECTOR
856	DC	320*FR26	WORD COUNT
857	DC	FRX27	PHASE=27 DISK SECTOR
858	DC	320*FR27	WORD COUNT
859	DC	FRX28	PHASE=28 DISK SECTOR
860	DC	320*FR28	WORD COUNT
861	ORG	FCOM	FORTRAN COMMUNICATIONS ARE
862	FX001	BSS	1 START OF STRING
863	FX002	BSS	1 END OF STRING
864	FX003	BSS	1 START OF SYMBOL TABLE
865	FX004	BSS	1 START OF NON-STATEMENT NUM
866	FX005	BSS	1 START OF SUBSCRIPT TEMPORA
867	FX006	BSS	1 START OF GENERATED TEMPORA
868	FX007	BSS	1 END OF SYMBOL TABLE
869	FX008	BSS	1 NEXT AVAILABLE COMMON
870	FX009	BSS	1 SIZE OF COMMON
871	FX010	BSS	1 OVERLAP ERROR
872	FX011	BSS	1 PROGRAM NAME
873		BSS	1 2ND WORD OF NAME
874	FX012	BSS	1 SUBROUTINE(=) OR FUNCTION(
875	FX013	BSS	1 CONTROL CARD WORD
876	*		BIT 15 TRANSFER TRACE
877	*		BIT 14 ARITHMETIC TRACE
878	*		BIT 13 EXTENDED PRECISION
879	*		BIT 12 LIST SYMBOL TABLE
880	*		BIT 11 LIST SUBPROGRAM NAMES
881	*		BIT 10 LIST SOURCE PROGRAM
882	*		BIT 9 ONE WORD INTEGERS
883	*		BIT 8 PUNCH
884	*		BIT 7 NONPROCESS PROGRAM
885	FX014	BSS	1 IOCS CONTROL CARD WORD
886	*		BIT 15 CARD
887	*		BIT 14 PAPER TAPE
888	*		BIT 13 TYPEWRITER
889	*		BIT 12 1443 PRINTER
890	*		BIT 11 MAGNETIC TAPE
891	*		BIT 10 KEYBOARD
892	*		BIT 9 1442 PUNCH
893	*		BIT 8 DISK
894	*		BIT 7 NOT USED
895	*		BIT 6 NOT USED
896	*		BIT 5 NOT USED
897	*		BIT 4 NOT USED
898	*		BIT 3 PLOTTER
899	*		BIT 2 NOT USED

900	*			BIT 1	UNFORMATTED DISK
901	*			BIT 0	UNFORMATTED TAPE
902	FX015	BSS	1		FILE COUNT
903	LCOMN	BSS	2		INSKEL COMMON SIZE
904	ICCR	DC			IOCS CONTROL CARD ERROR
905	SQFS	EQU	FX001		FCOM EQUIVALENCES
906	EQFS	EQU	FX002		*
907	SQFST	EQU	FX003		*
908	SQFNS	EQU	FX004		*
909	SQFXT	EQU	FX005		*
910	SQFGT	EQU	FX006		*
911	EQFST	EQU	FX007		*
912	COMON	EQU	FX008		*
913	CSIZE	EQU	FX009		*
914	ERROR	EQU	FX010		*
915	FNAME	EQU	FX011		*
916	SORF	EQU	FX012		*
917	CCWD	EQU	FX013		*
918	IOCS	EQU	FX014		*
919	DFCNT	EQU	FX015		*
920		ORG	OVERL		
921	*				CARD ROUTINE
922	CARD	DC	***		
923		LD	ENDSW		
924		BSC	L F1035,Z		LAST WAS END
925		BSI	L LIO		READ NEXT CARD
926		DC	/1001		
927		DC	CAREA		
928		DC	0		
929		BSI	L LIO		WAIT FOR TRANSFER
930		DC	/F001		
931		LD	L FX013		
932		SLA	10		
933		BSC	L CAR01,-		NO SOURCE LIST
934		BSI	L LIO		LIST SOURCE
935		DC	/2005		
936		DC	CAREA		
937		DC	0		
938		BSI	L LIO		WAIT FOR TRANSFER
939		DC	/F005		
940	CAR01	STX	1 CAR05+1		SAVE X1
941		LDX	1 72		OUTPUT CHARACTERS
942		LDX	2 36		WORDS TO CONVERT
943	CAR02	LD	L2 CAREA		
944		AND	CARX1		GET RIGHT BYTE
945		BSI	CHVRT		CONVERT TO EBCDIC
946		LD	L2 CAREA		
947		SRA	8		
948		BSI	CHVRT		CONVERT TO EBCDIC
949		MDX	2 -1		
950		MDX	CAR02		LOOP
951		S	CARX2		TEST FOR ,C
952		BSC	L CAR05,+		NO END IF COMMENT
953		LDX	1 -4		
954		LDX	2 -67		
955	CAR03	LD	L2 CAREB+73		TEST FOR END CARD
956		S	CARX3		TEST FOR BLANK
957		BSC	L CAR04,+		PASS BLANKS
958		S	L1 CARX4+4		E,N,D
959		BSC	L CAR05,Z		NOT END CARD

```

960          MDX      1 1
961  CAR04  MDX      2 1
962          MDX      CAR03      TEST LOOP
963          MDX      1 1
964          MDX      CAR05      NOT END CARD
965          STX      ENDSW      SET END FLAG
966  CAR05  LDX      L1 *-*      RESTORE X1
967          LDX      L2 CAREB+1  X2 POINTS TO IMAGE
968          BSC      I  CARD      EXIT
969  ENDSW  DC        *-*      END SWITCH
970  CARX1  DC        /FF
971  CARX2  DC        .C
972  CARX3  DC        .
973  CARX4  DC        .E-,
974          DC        .I-,
975          DC        .D-,
976          DC        /1FF
977  *          ROUTINE TO CONVERT ASCII TO EBCDIC
978  CNVRT  DC        *-*      ENTRANCE (A) IS CHAR
979          S          CNVX1      /A0
980          BSC      L  CNV03,+Z  FORCE SPACE FOR ) /A0
981          CMP      CNVX2      /DA=/A0
982          MDX      CNV03      FORCE SP. FOR /DA
983          MDX      *
984          SRT      1
985          A          CNVX3      =CNVTB
986          STO      CNV01
987          SLT      17          CARRY = LOW ORDER
988          LD        L  *-*      BET EBCDIC WORD
989  CNV01  EQU      *-1
990          BSC      C          EVEN IS RIGHT HALF
991          SRA      8          GET LEFT HALF
992          AND      CARX1      GET CHARACTER
993  CNV02  STO      L1 CAREB      STORE EBCDIC CHAR.
994          MDX      1 -1
995          MDX      *
996          BSC      I  CNVRT      EXIT
997  CNV03  LD        CARX3      SPACE
998          MDX      CNV02
999  CNVX1  DC        /A0
1000  CNVX2  DC        /DA=/A0
1001  CNVX3  DC        CNVTB
1002  *          CONVERSION TABLES
1003  CNVTB  EBC      . = ($**)(***/.,
1004          EBC      .1032547698 =) ,
1005          EBC      .A*CBEDQFIHKJMLON,
1006          EBC      .QPSRUTWVYX Z,
1007  *          INPUT AREAS
1008  CAREA  DC        40
1009          BSS      40          ASCII SOURCE IMAGE
1010  CAREB  DC        72
1011          BSS      72          EBCIDIC IMAGE
1012  *          MAIN LINE PROGRAM
1013  F1000  BSI      L  CARD      READ A CARD
1014  F1001  EQU      F1000      READ A CARD
1015          STX      2 F1003+1  SAVE XR2
1016          LD        L  F1951    LOAD SWITCH 1
1017          BSC      L  F1005,Z  BRANCH IF SW1 NOT ZER
1018          LD        2 0          LOAD COLUMN 1
1019          S          L  F1963    IS IT AN *

```

1020		BSC	L	F2003,+	BRANCH IF YES (CONTPOL CD
1021		STX	L	F1951	TAG SWITCH 1
1022		BSC	L	F1005	
1023	F1003	LDX	L2	*-*	
1024		LD		F1951	LOAD SWITCH 1
1025		BSC	L	F1001,+	GO READ A CARD IF NOT SET
1026	F1005	LD	2	0	IS THIS A COMMENT CARD
1027		S		F1967	
1028		BSC	L	F1006,Z	BRANCH IF NO
1029		STO		F1952	NORMALIZE SWITCH 2
1030		MDX		F1001	GET NEXT CARD
1031	F1006	LD	2	5	TEST FOR CONTINUATION CAR
1032		S		F1965	
1033		BSC	L	*+3,+	BRANCH IF BLANK
1034		S		F1966	
1035		BSC	L	F1016,Z	BRANCH IF NON-ZERO
1036		STO		F1954	NORMALIZE SWITCH 4
1037		STO	L	F1959	NORMALIZE SWITCH 9
1038		STO	L	F1984	CLEAR DATA AREA
1039		LDX	3	72	INITIALIZE TO TEST FOR
1040		STX	2	*+1	BLANK CARD,
1041		LD	L	*-*	LOAD CARD COLUMN
1042	F1007	EQU		*+1	LABEL FOR STANDARDS
1043		S		F1965	
1044		BSC	L	*+5,Z	BRANCH IF COLUMN NOT BLAN
1045		MDX	L	F1007,1	INCREMENT TO NEXT COLUMN
1046		MDX	3	-1	KICK COUNT, SKIP IF ZERO
1047		MDX		F1007-1	CHECK NEXT COLUMN
1048		MDX		F1001	BLANK CARD, GET NEXT
1049		STX		F1952	SET SWITCH 2
1050	F1008	SLT		32	INITIALIZE
1051		STO		F1969	TO COLLECT
1052		LD		F1962	STATEMENT
1053		STD		F1972	NUMBERS,
1054		LDX	3	5	
1055	F1009	LD	2	0	CHECK IF COLUMN BLANK
1056		S		F1965	
1057		BSC	L	F1012,+	BRANCH IF BLANK
1058		S		F1966	IS COLUMN ZERO
1059		BSC	L	F1010,Z	BRANCH IF NON-ZERO
1060		LD		F1969	IS IT A LEADING ZERO
1061		BSC	L	F1011,Z	BRANCH IF NO
1062		MDX		F1012	YES, LEADING ZERO
1063	F1010	BSC	L	F1030,+	BRANCH IF NON-NUMERIC
1064		S		F1968	
1065		BSC	L	F1030,-Z	BRANCH IF NON-NUMERIC
1066	F1011	STX		F1969	INDICATE VALID DIGIT
1067		LDD		F1972	ADD
1068		RTE		26	DIGIT
1069		STO		F1972	TO
1070		LD	2	0	STATEMENT
1071		AND		F1961	NUMBER
1072		OR		F1972	*
1073		STD		F1972	*
1074	F1012	MDX	2	1	GO TO NEXT CHARACTER
1075		MDX	3	-1	SKIP IF THROUGH
1076		MDX		F1009	GET NEXT DIGIT IF NOT
1077		LD		F1969	IS THERE A STATEMENT NO,
1078		BSC	L	F1013,+	BRANCH IF NOT
1079		LDD		F1972	NORMALIZE

1080	RTE		16	STATEMENT
1081	SLT		1	NUMBER
1082	BSC	L	*-3,-	(PACKED
1083	STO		F1972	6 BITS PER
1084	SRA		15	DECIMAL CHARACTER)
1085	SLT		15	*
1086	STO		F1972+1	*
1087	F1013 MDX	2	1	GET NEXT CHARACTER
1088	SLA		16	NORMALIZE
1089	STO		F1971	CHARACTER COUNT
1090	LD		F1953	LOAD SWITCH 3
1091	BSC	L	F1014,+-	BRANCH IF SWITCH 3 NOT SE
1092	BSI	L	F1029	COMPUTE NORM
1093	F1014 STX		F1953	SET SWITCH 3
1094	*			(XR1 POINTS AT SOFS)
1095	STX	1	F1970	
1096	SLA		16	
1097	BSI	L	F1050	GO TO PUT ID WORD
1098	LD		F1969	LOAD STATEMENT NO, IND
1099	BSC	L	F1015,+-	BRANCH IF NO STATEMENT NO
1100	LD	I	F1970	INDICATE
1101	OR		F1962	STATEMENT
1102	STO	I	F1970	NUMBER
1103	LD		F1972	PUT
1104	BSI	L	F1050	STATEMENT
1105	LD		F1972+1	NUMBER,
1106	BSI	L	F1050	*
1107	F1015 SLA		16	
1108	STO		F1955	NORMALIZE SWITCH 5
1109	STO		F1956	NORMALIZE SWITCH 6
1110	STO		F1957	NORMALIZE SWITCH 7
1111	STO	L	F1991	NORMALIZE FCOT
1112	MDX		F1017	CONTINUE
1113	*			CONSTANTS, STORAGE, AND SWITCHES
1114	F1951 DC		0	SWITCH 1
1115	F1952 DC		0	SWITCH 2
1116	F1953 DC		0	SWITCH 3
1117	F1954 DC		0	SWITCH 4
1118	F1955 DC		0	SWITCH 5
1119	F1956 DC		0	SWITCH 6
1120	F1957 DC		0	SWITCH 7
1121	F1961 DC		/003F	MASK FOR STM NO, PACKING
1122	F1962 DC		1	USEFUL CONSTANT
1123	F1963 DC		.*	* TEST
1124	F1965 DC		/0040	EBC BLANK TEST
1125	F1966 DC		/F0-/40	EBC ZERO TEST
1126	F1968 DC		/F9-/F0	DIGIT TEST
1127	F1969 DC		0	STATEMENT NO, TEST
1128	F1970 DC		0	ID SAVE WORD
1129	F1967 DC		.C	C TEST
1130	F1971 DC		0	CHARACTER COUNT
1131	BSS	E	0	
1132	F1972 DC		0	STATEMENT NUMBER
1133	DC		0	COLLECTION AREA
1134	F1016 LD		F1954	LOAD SWITCH 4
1135	BSC	L	F1001,Z	BRANCH IF SW 4 SET
1136	LD		F1952	LOAD SWITCH 2
1137	BSC	L	F1031,+-	BRANCH TO ERROR IF NOT SE
1138	MDX	L	F1971,1	INCREMENT CHARACTER COUNT
1139	LD		F1971	ERROR IF COUNT IS GREATER

1140		S		F1985	*THAN 6,
1141		BSC	L	F1054,-	*
1142		MDX	2	6	MV PT TO CD COLUMN SEVEN
1143	F1017	LDX	3	66	INITIALIZE TO PACK RECORD
1144	F1018	LD		F1956	LOAD SWITCH 6
1145		BSC	L	F1021,Z	BRANCH IF SWITCH 6 SET
1146	F1019	LD	2	0	IS COLUMN BLANK
1147		S		F1965	
1148		BSC	L	F1021,Z	BRANCH IF NOT BLANK
1149	F1020	MDX	2	1	INCR TO NEXT COLUMN
1150		MDX	3	-1	SKIP IF WHOLE CARD CHECKE
1151		MDX		F1018	CHECK NEXT COLUMN
1152		BSC	L	F1001	READ NEXT CARD
1153	F1021	LD		F1959	LOAD SWITCH 9
1154		BSC	L	F1028,Z	BRANCH IF SWITCH 9 SET
1155	F1022	MDX	L	F1984,/10	CDAIA = CDAIA + /10
1156		LD		F1957	LOAD SWITCH 7
1157		BSC	L	F1023,Z	BRANCH IF SWITCH 7 SET
1158		LD	2	0	PJT
1159		SLA		8	LEFT
1160		OR		F1965	CHARACTER,
1161		BSI	L	F1050	*
1162		STX		F1957	SET SWITCH 7
1163		MDX		F1020	GET NEXT CHARACTER
1164	F1023	SLA		16	NORMALIZE
1165		STO		F1957	SWITCH 7
1166		LD	1	-1	ADD
1167		AND		F1995	RIGHT
1168		OR	2	0	CHARACTER
1169		STO	1	-1	*
1170		LD		F1984	LD CDAIA
1171		S		F1965	
1172		BSC	L	F1027,+-	BRANCH IF EQ TO /40
1173	F1024	LD		F1955	LOAD SWITCH 5
1174		BSC	L	F1020,Z	BRANCH IF SWITCH 5 SET
1175		MDX	L	F1991,1	INCREMENT FORMAT COUNT
1176		LD		F1991	IS CHARACTER
1177		A		F1993	PART OF
1178		STO		**+1	FORMAT
1179		LD	L	**	*
1180		S	1	-1	*
1181		BSC	L	F1025,+-	BRANCH IF YES
1182		STX		F1955	TAG SWITCH 5
1183		MDX		F1020	CONTINUE TO PACK
1184	F1025	LD		F1991	LOAD FORMAT COUNT
1185		S		F1992	
1186		BSC	L	F1020,Z	BRANCH IF NOT EQUAL TO 3
1187	F1026	MDX	1	-3	XR1 = XR1#3
1188		LD	I	F1970	SET
1189		OR		F1990+3	FORMAT
1190		STO	I	F1970	STATEMENT ID,
1191		STX		F1956	TAG SWITCH 6
1192		MDX		F1020	CONTINUE, DON T PACK FMT
1193	*			DATA STATEMENT	
1194	F1027	LD	1	-1	ARE SECOND PAIR OF
1195		S		F1982	CHARACTERS EQ TO (TA)
1196		BSC	L	F1024,Z	BRANCH IF NO
1197		LD	1	-2	ARE FIRST PAIR OF
1198		S		F1981	CHARACTERS EQ TO (DA)
1199		BSC		+-	SKIP IF NO

1200		STX		F1959	TAG SWITCH 9
1201		MDX		F1024	CONTINUE WITH PACK RTN.
1202	*				CONSTANTS, WORK AREAS AND SWITCHES
1203	F1981	DC		/C4C1	DA TEST
1204	F1982	DC		/E3C1	TA TEST
1205	F1983	DC		/7D	QUOTE TEST
1206	F1984	DC		***	DATA WORK AREA
1207	F1959	DC		***	SWITCH 9
1208	F1028	LD	2	0	IS CHARACTER A QUOTE
1209		S		F1983	
1210		BSC	L	F1022,Z	BRANCH IF NO
1211		LD		F1956	LOAD SWITCH 6
1212		EOR		*	REVERSE SWITCH 6
1213		STO		F1956	STORE
1214		MDX		F1022	CONTINUE
1215	F1029	DC		0	BSI ADDRESS
1216		LD	1	-1	ELIMINATE
1217		S		F1994	TRAILING
1218		BSC	L	**2,Z	BLANKS
1219		MDX	1	-1	*
1220		MDX		F1029+1	LOOP BACK
1221		STX	1	F1986	COMPUTE
1222		LD		F1986	NORM
1223		S		F1970	*
1224		SLA		2	*
1225		OR	I	F1970	*
1226		STO	I	F1970	STORE NORM IN IDSAVY
1227		BSC	I	F1029	RETURN
1228	*				CONSTANTS AND WORK AREAS
1229	F1985	DC		6	USEFUL CONSTANT
1230	F1986	DC		0	WORK WORD
1231	F1987	DC		0	ERNO
1232	F1988	DC		/A008	ERROR ID
1233	F1989	DC		0	OVERLAP COUNTER
1234	F1990	DC		/C6D6	FO
1235		DC		/D9D4	RM
1236		DC		/C1E3	AT
1237		DC		/6000	FORMAT ID WORD
1238	F1991	DC		0	FORMAT COUNT
1239	F1992	DC		3	USEFUL CONSTANT
1240	F1993	DC		F1990-1	COMPARE POINTER FOR FORMA
1241	F1994	DC		/4040	BLANK TEST
1242	F1995	DC		/FF00	MASK WORD
1243	F1996	DC		/1000	END STATEMENT ID
1244	F1030	LDX	3	1	SET ERROR NUMBER
1245		STX	3	F1987	EQUAL TO 1
1246		MDX		F1032	GO TO ERROR
1247	F1031	LDX	3	2	SET ERROR NUMBER
1248		STX	3	F1987	EQUAL TO 2
1249	F1032	LD	L	F1953	LOAD SWITCH 3
1250		BSC	L	F1033,+-	BRANCH IF SWITCH 3 NOT SE
1251		BSI		F1029	COMPUTE NORM
1252	F1033	LD		F1988	PUT ERROR ID
1253		BSI		F1050	GO PUT ON STRING
1254		LD		F1987	PUT ERROR NUMBER
1255		BSI		F1050	GO PUT ON STRING
1256	F1034	STX	L	F1954	TAG SWITCH 4
1257		SLA		16	NORMALIZE SWITCH 3
1258		STO	L	F1953	
1259		BSC	L	F1001	GET NEXT CARD

1260	F1035	LD	L	F1953	LOAD SWITCH 3
1261		BSC	L	F1036,+	BRANCH IF NOT SET
1262		BSI		F1029	COMPUTE NORM
1263	F1036	LD		F1996	LOAD END ID
1264		STX	L1	FX002	SET END OF STRING POINTER
1265		BSI		F1050	PUT END STATEMENT ID
1266	*	ERROR			ROUTINE FOR SPECIFICATION OF BOTH
1267	*				UNFORMATTED TAPE AND DISK
1268		LD	L	IOCS	GET THE IOCS WORD
1269		SRA		14	SAVE ONLY BITS 0 AND 1
1270		S		H0003	COMPARE TO CONSTANT
1271		BSC	L	**2,Z	BR IF BITS 0,1 NOT BOTH ON
1272		STX	L0	ICCR	ELSE SET IND FOR LIST ERRO
1273		MDX		EXIT	GO CALL DOWN PHASE 2
1274	H0003	DC		0003	CONSTANT WITH BITS 14,15 0
1275	EXIT	BSI	L	ROLRX	GO GET PHASE 2 FROM DISK
1276		DC		02	NEXT PHASE NUMBER
1277	*				THE FOLLOWING SUBROUTINE PUTS THE
1278	*				WORD THAT IS IN THE A REGISTER ON TH
1279	*				STRING THAT IS INDICATED BY INDEX
1280	*				REGISTER 1, ENTRY IS A BSI
1281	*				INSTRUCTION TO THE LABEL (F1050),
1282	*				INDEX REGISTER 1 IS INCREMENTED BY
1283	*				1 UPON EXITING,
1284	F1050	DC		0	BSI ADDRESS
1285	F1051	STO	1	0	STORE A REGISTER IN STRIN
1286	F1052	MDX	1	1	MOVE POINTER (XR1)
1287		STX	1	F1989	CHECK FOR
1288		LD		F1989	OVERLAP
1289		S	L	FX007	ERROR,
1290		BSC	L	F1053,+Z	BRANCH IF NO ERROR
1291		LD	L	F1962	SET OVERLAP
1292		STO	L	FX010	ERROR INDICATOR
1293		LD		F1051	SET UP TO NO LONGER
1294		STO		F1052	INCREMENT XR1
1295	F1053	BSC	1	F1050	RETURN
1296	F1054	LDX	3	2	SET ERROR NUMBER
1297		STX	3	F1987	EQUAL TO 2,
1298		LDX	11	F1970	XR1 = CONTENTS OF IDSAV
1299		LD	1	0	
1300		BSC	L	**1,E	BRANCH IF STATEMENT NUMBR
1301		MDX		F1033	BRANCH IF NOT
1302		LD		F1999	PUT OUT ERROR ID FOR
1303		STO	1	0	STM NO,
1304		LD		F1987	PUT OUT ERROR NO
1305		STO	1	3	
1306		MDX	1	4	INCREMENT POINTER
1307		MDX		F1034	GO SET SWITCH 4
1308	F1999	DC		/A011	ERROR ID
1309		BSS		ROLRX-++3*320	PHASE=01 PATCH AREA
1310		END		FZZ50	