DEMOPAK
A Functional Demonstration of the IBM 1620

DEMOPAK
A Functional Demonstration of the IBM 1620

Ray Peck
Bill Olmo
Dave Montgomery

Direct Inquiries to: W. J. Olmo
IBM Corporation
340 Market Street
San Francisco
California

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

2

DECK KEY

1. Object Deck
2. Input Data Sec. 8
* 3. Source Deck

* This deck will be forwarded only when specifically requested.

3

TABLE OF CONTENTS

DEMOPAK - A Functional Demonstration of the IBM 1620

Author: Ray Peck
Bill Olmo
Dave Montgomery

Direct Inquiries to: W. J. Olmo
IBM Corporation
340 Market Street
San Francisco, California

A. Purpose/Description: DEMOPAK is designed to be used as an introductory demonstration of various 1620 functions and to provide a background for the demonstration of programming systems and production programs.

B. Method: The program is divided into sections, each of which is a complete program which will type a heading and, if program switch 3 is on, a set of operating instructions. The section then comes to a halt and is executed by depressing START. Upon completion the program returns to the initial halt and is ready to be executed again. Program switches 1 and 2 are used in some sections to stop the operation, alter logic, or provide optional input or output.

When the demonstration of any section is complete the operator depresses RESET, INSERT, RELEASE, and START to proceed to the next demonstration section. By INSERTING the correct branch the operator can return to any section or change the sequence of section.

C. Restrictions and Range: N/A

D. Accuracy: N/A

E. Machine Configuration: DEMOPAK requires only the basic 1620 card system, no special features.

F. Program Requirements: N/A

G. Source Language: The program is written in 1620/1710 SPS (1620-SP-020).

H. Program Execution Time: It requires 30 to 50 minutes to run.

I. Check-Out Status: N/A

J. Sample Problem Running Time: N/A

K. Comments: In order to intelligently present DEMOPAK, the demonstrator should read:
Operating Instructions Page 33
Demonstration Narrative Page 42
This program and its documentation were written by an IBM employee. It was developed for a specific purpose and submitted for general distribution to interested parties in the hope that it might prove helpful to other members of the data processing community. The program and its documentation are essentially in the author's original form. IBM serves as the distribution agency in supplying this program. Questions concerning the use of the program should be directed to the author's attention.

PROGRAM DESCRIPTION

DEMOPAK is designed to demonstrate various features in the 1620 system. Eight sections are included in this package. The reader may add a ninth to meet the needs of the group to whom the demo is being presented, or delete any section. Each section is written in 1620/1710 SPS (1620-SP-020). The SPS condensed deck is then added to DEMOPAK after removing the first two cards and the last seven cards. Each section is allotted 1800 positions of memory as shown:

Section	1000 Positions Starting at	and	800 Positions Starting at
1	1000		10,000
2	2000		10,800
3	3000		11,600
4	4000		12,400
5	5000		13,200
6	6000		14,000
7	7000		14,800
8	8000		15,600
9	9000		16,400

CONTENTS OF THE DECK

DECK I

		Seq. No.
Pre-Demo Instructions		100 - 124
Initialization		200 - 211
Subroutines		300 - 347
Card Reading	Sec. 1	1,000 - 1,043
Card Output	Sec. 2	2,000 - 2,025
Typewriter I/O	Sec. 3	3,000 - 3,025
Addition	Sec. 4	4,000 - 4,027
Digits to Alphabetic	Sec. 5	5,000 - 5,056
Rapid Successive Addition	Sec. 6	6,000 - 6,046
Successive Multiplication	Sec. 7	7,000 - 7,047
Combination of Effects	Sec. 8	8,000 - 8,047
Math Tables		400 - 406

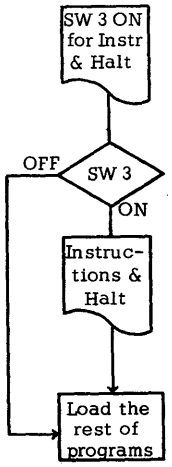
It should be noted that the Pre-Demo Instructions may be removed if desired.

DECK II

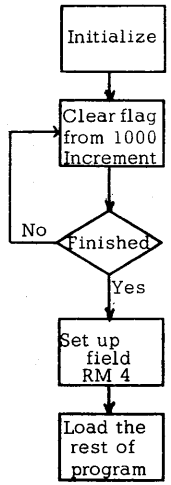
Typical input data for Section 8

DEMOPAK contains two pre-programs which are overloaded by the main program.

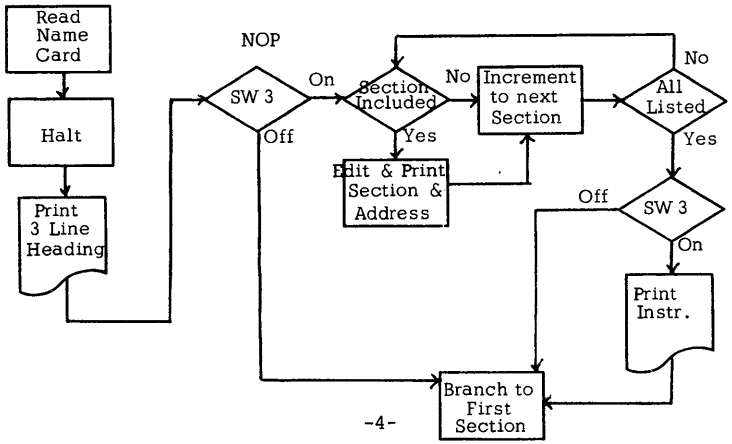
The first is the section giving pre-demo instructions. After the first few cards have been read a message is typed saying SW 3 ON for pre-demo instructions. The operator then has the option of getting margin and tab stop information.



The second is the initialization program. This does two things. First, it clears flags from all the thousands positions and second, it sets up a field of four record marks.



DEMOPAK starts by executing the Demonstration Heading subprogram. This program prints the heading, operating instructions and the starting position of the sections being demonstrated.



There are several subroutines used in DEMOPAK. They are:

ALPHA	Print Alpha and Skip Subroutine
EDIT	Numeric to Alpha Edit Subroutine
NEXT	Branch to Next Function Subroutine
LD 80	Prepare I/O Area Subroutine
LD 160	Prepare I/O Area Subroutine

ALPHA Subroutine:

The purpose of this subroutine is to print messages and return the carriage of the typewriter.

The linkage is:

```
          BTM       ALPHA,XXXXX
```

where XXXXX is the label of the first line to be printed.

The instructions are in the form:

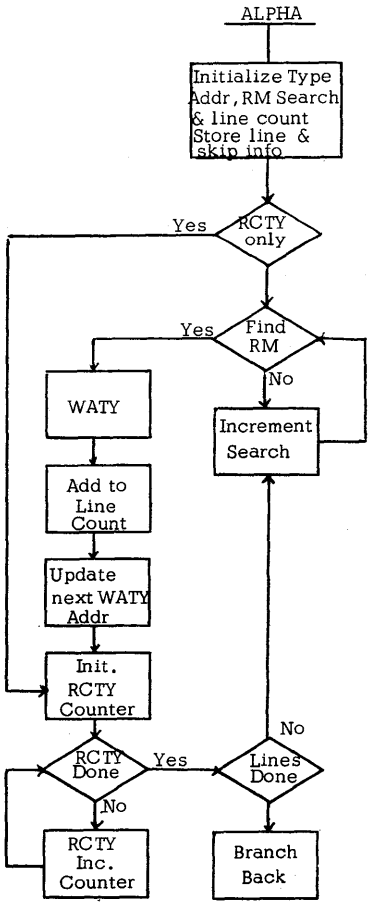
```
          BTM       ALPHA,XXXXX
                    .
                    .
                    .
          DC        4,YYZZ
XXXXX     DAC
                    .
                    .
                    .
          DAC
ALPHA     DS        ,00694
```

where YY is the number of lines to be typed and ZZ is the number of times the carriage is to be returned after each line. Each line as defined by its DAC must be terminated by a record mark.

This subroutine can also be used to return the carriage of the typewriter in this manner:

```
          BTM       ALPHA,XXXXX+2
                    .
                    .
                    .
XXXXX     DC        4,00ZZ
ALPHA     DS        ,00694
```

where ZZ is again the number of carriage returns.



EDIT Subroutine:

The purpose of this subroutine is to simulate the Transfer Numerical Fill command and to edit out high order zeroes.

The linkage is:

```
          BTM       EDIT,XXXXX
```

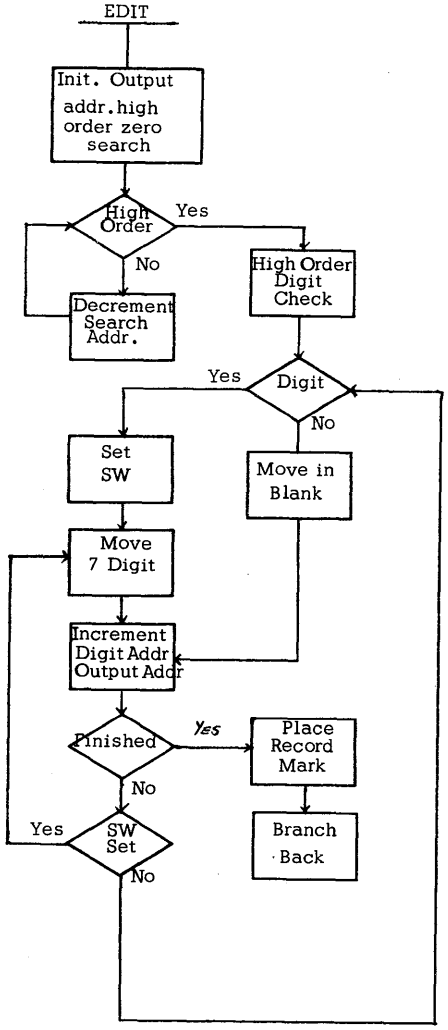
where XXXXX is the address of the field to be edited. The instruction will be in the form:

```
          BTM       EDIT,XXXXX
          .
          .
          .
EDIT      DS        ,17206
EDNUMB    DS        ,19001
```

The results, with a leading minus sign for negative data and a trailing record mark, will be stored at EDNUMB. Typical instructions will then be:

```
          WATY      EDNUMB
          TR        ZZZZZ,EDNUMB-1
```

The number of spaces required for the edited number is 2 * (Y+1), where Y is the length of the field being edited.



NEXT Subroutine:

The purpose of this subroutine is to give an automatic branch to the next section when RESET, INSERT, RELEASE and START are depressed in that order.

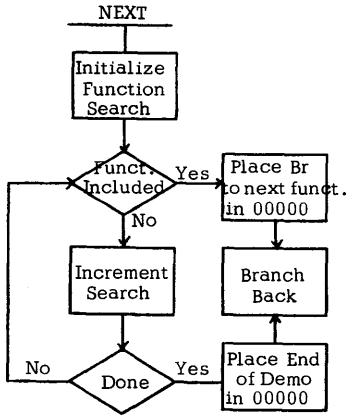
The linkage is:

```
        BTM     NEXT,XXXX
```

where XXXX is the thousands position where the present section starts.

The instruction will be in the form:

```
        BTM     NEXT,XXXX
        .
        .
        .
NEXT    DS      ,17608
```



LD 80 Subroutine:

The purpose of this subroutine is to set the input/output area, the last eighty positions of storage, to any desired character.

The linkage is:

```
          BT        LD80,XXXX,8
```

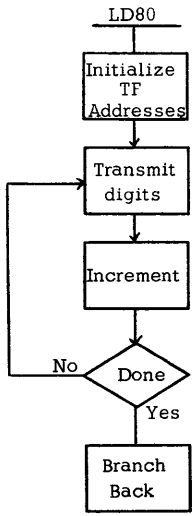
where XXXX are the digits to be set in storage. The instructions will be in the form:

```
          BT        LD80,XXXX,8
           .
           .
           .
LD80      DS        ,17798
BAREA     DS        ,19920
```

where BAREA is the label used for the last 80 storage positions. If it is desired to set the area to record marks we use:

```
          BT        LD80,RM4
           .
           .
           .
LD80      DS        ,17798
RM4       DS        ,405
```

where RM4 is a field of four record marks.



LD 160 Subroutine:

This is actually the same subroutine as the LD 80 except it sets the last one hundred sixty positions of storage.

The linkage is:

```
          BTM       LD160,XXXX,8
```

where XXXX are the digits to be set in storage. The instructions will be in the form:

```
          BTM       LD160,XXXX,8
           .
           .
           .
LD160     DS        ,17834
AAREA     DS        ,19841
```

where AAREA is the label given to the last 160 positions of storage.

DEMOPAK contains eight sections. The description of these sections follows, with the format:

A. Purpose

B. Input

C. Output

D. Processing

E. Demonstration Procedure and Significant Points

F. Block Diagram

SECTION I - READ CARDS

A. Purpose - to illustrate

1) Card read speed

2) Typewriter as message station

B. Input - cards in read unit

C. Output - typed message stating number of cards read

D. Processing

1) Cards are read.

2) Switches are interrogated to determine when card reading should end.

E. Demonstration Procedure and Significant Points

1) Place over 1000 cards in the card reader, SW 1 and 2 OFF.

2) Depress START and READER START.

3) Identify the various components of the 1620 system, explain the idea of a stored program and answer any questions.

4) After 1000 cards have been read reading will stop and a message will be typed out.

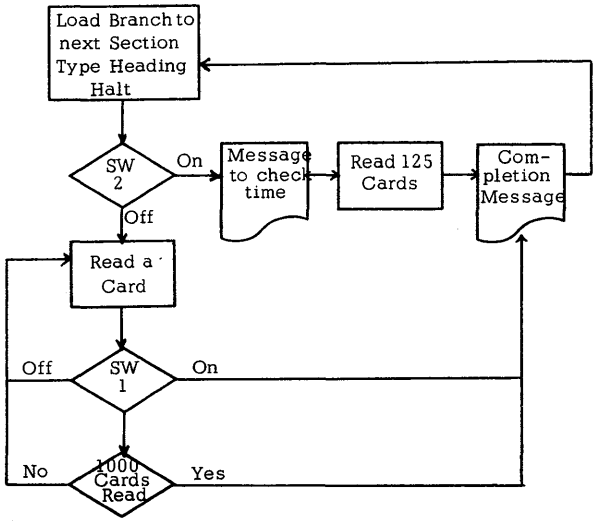
5) To show that the cards are counted as read depress START and after several cards have been read turn SW 1 ON.

6) Reload the card read hopper; point out the read speed is 250 cards per minute. Turn Switch 2 on and depress START.

7) A message will be typed out to check the time when ready depress START and get a timed run of 125 cards.

8) Any of the above steps may be repeated.

F.



SECTION II - PUNCH CARDS

A. Purpose - to illustrate

1) Punch speed

2) Typewriter as message station

3) Ability to generate random numbers

B. Input - blank cards in punch unit

C. Output

1) With SW 2 ON 125 data cards containing two random numbers each

2) When SW 1 is turned ON data cards, up to time of turning SW on, with two random numbers

3) Error message

D. Processing

1) Test for correct switch settings.

2) Multiply two numbers picking parts of the product as random numbers.

3) Punch cards containing random numbers.

E. Demonstration Procedure and Significant Points

1) Depress START and PUNCH START.

2) Cards are now being punched at the rate of 125 cards per minute.

3) Turn SW 1 ON to stop punching.

4) Depress NON-PROCESS RUNOUT on punch unit but do not remove the two blank cards.

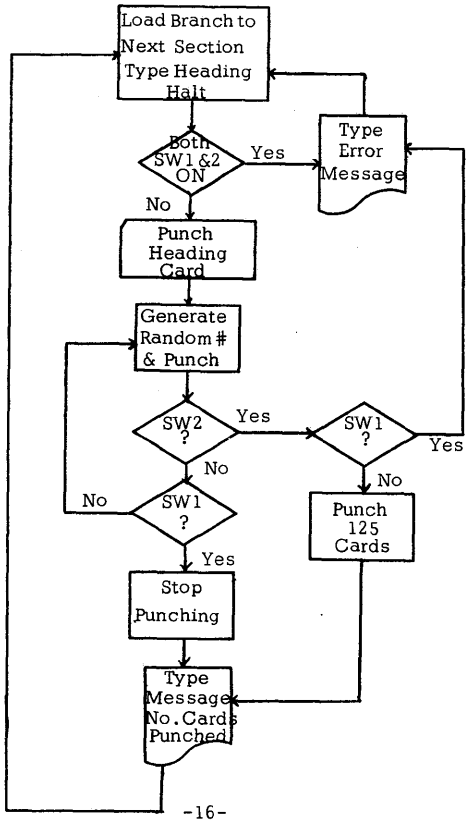
5) Repeat procedure with SW 2 ON, SW 1 OFF and time output.

6) Repeat procedure with both SW 1 and SW 2 ON.

7) 80-80 list the output cards and point out that two random numbers have been generated in each card.

8) Point out that the punch unit runs at top speed while these calculations are going on; this demonstrates the buffered punch unit.

F.



SECTION III - TYPEWRITER INPUT/OUTPUT

A. Purpose - to illustrate

1) The Alphameric and Numeric modes of typewriter input/output

2) The ability of the 1620 to handle variable length information in typewriter input/output operations

3) The entry of numeric information in the Alphameric mode by depression of the numeric shift key

4) The ability to select the input/output mode based on program switches

5) Procedure for correction of errors in entered from the typewriter

B. Input

1) SW 1 ON, SW 2 OFF - A number up to 150 digits in length may be entered from the typewriter.

2) SW 1 OFF, SW 2 ON - Up to 75 characters of alphameric information may be entered from the typewriter.

C. Output

1) The typewriter prints an "A", a number telling how many inputs have been made to this section, an equals sign, and then the input data. For example: If data such as

I AM AN IBM 1620

were typed as input (SW 1 OFF, SW 2 ON for alphameric mode), then the typewriter would print the following message:

A01 = I AM AN IBM 1620

If SW 1 is now turned ON and SW 2 turned OFF (numeric mode) and the data

1 2 3 4 5 6 7

is entered from the keyboard, the output will be

A02 = 1 2 3 4 5 6 7

2) If 28 or fewer digits are entered, the typewriter will tabulate and print the variable data that was entered.

3) If more than 28 digits are entered, the typewriter will return the carriage two times and then print the variable data that was entered.

D. Processing

1) Input filled with record marks prior to entry.

2) Test for valid switch settings and mode of input.

3) Increments the counter which accumulates the number of entries.

4) Prints the Output. (See Section C for the format of the output.)

E. Demonstration Procedure and Significant Points

1) Instructions are typed out if SW 3 is ON.

2) Depress START leaving both SW 1 and SW 2 OFF an error message will be typed.

3) SW 1 and SW 2 both ON and depress START. This will cause a second error message to type.

4) SW 1 ON, SW 2 OFF for numeric input.

5) Depress START.

6) Typewriter selected in the numeric mode.

7) Point out the 36 (Read Numeric) in the OP register with the 01 sense code.

8) Enter digits - about 10.

9) Point out that MARS (Memory Address Register) indicates locations in memory where digits are stored.

10) Discuss variable word length.

11) Depress RELEASE and START.

12) Typewriter tabulates and prints a variable name (A01 =) followed by the entered data.

13) Depress START and typewriter will again be selected.

14) Enter more than 28 digits.

15) Depress RELEASE and START.

16) The carriage will return and the variable name (A02 =) will print followed by the entered data.

17) Point out the detection of a long record with the tabulation and carriage return under stored program control.

18) Depress START and typewriter will be selected.

19) Have members of the group enter data.

20) If an error occurs or the question of a typing error comes up - describe and demonstrate typewriter error correction. If an error is made, turn ON SW 4 before depressing RELEASE and START. Depress RELEASE and START. Turn OFF SW 4. Type in correct data. If RELEASE and START have been depressed prior to detecting the error and turning SW 4 ON, the error cannot be corrected.

21) SW 1 OFF and SW 2 ON for alphameric input/output.

22) Depress START and typewriter will be selected in the alphameric mode.

23) Type in up to 28 characters. Use the numeric shift key to enter a few numbers, e.g.

I AM AN IBM 1620

24) Depress RELEASE and START.

25) Typewriter will tabulate and print the variable name followed by the entered data.

26) Depress START and typewriter will again be selected.

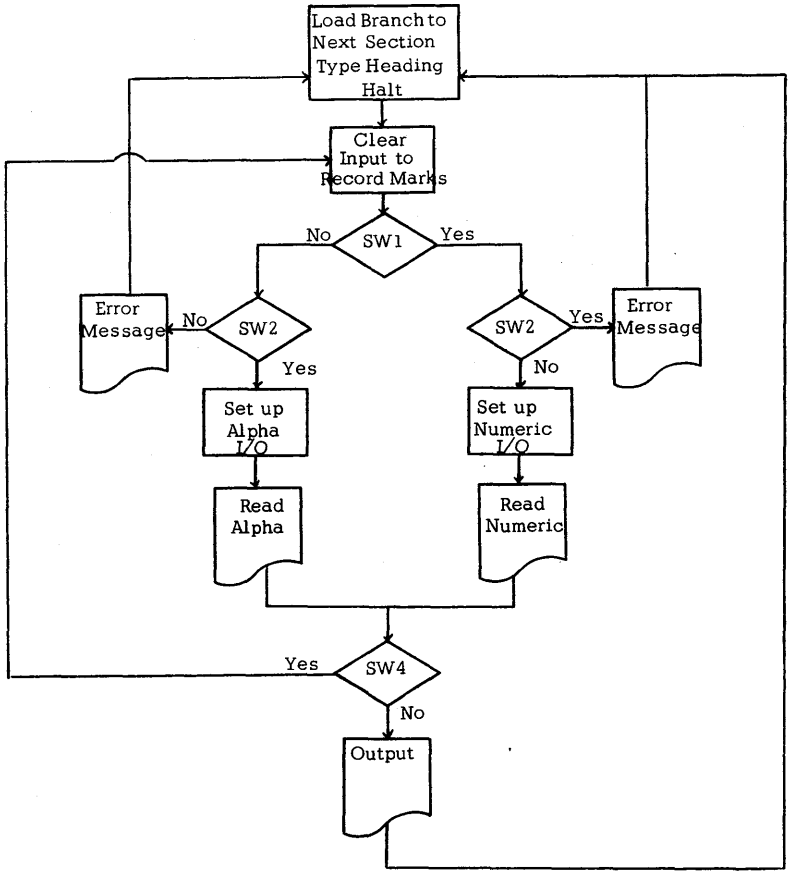
27) Type in more than 28 characters.

28) Depress RELEASE and START.

29) Typewriter will return carriage and print the variable name followed by the entered data.

30) Have the members of the group enter alphameric information.

F.



SECTION IV - ADD A COLUMN OF FIGURES

A. Purpose - to illustrate

1) The ability to change a job's procedure based on a program switch

2) Typewriter entry error procedure

3) Calculation speed

B. Input

1) A number up to 8 positions, entered on console typewriter

2) A two position number telling how many times to add the number to itself thus simulating the addition of a column of figures

C. Output

1) SW 1 OFF the sum of the column of figures

2) SW 1 ON each subtotal and the final total.

3) Error messages

D. Processing

1) Test for correct number of digits entered.

2) Add the number to itself the specified number of times.

E. Demonstration Procedure and Significant Points

1) Depress START.

2) A message is typed out to enter an eight digit field, and the typewriter is selected in the numeric mode.

3) Enter up to eight digits and depress RELEASE and START.

4) A message is typed out to enter the number of times the above number is to be added to itself and the typewriter is selected in the numeric mode.

5) Enter two digits and depress RELEASE and START or R-S key on typewriter.

6) With Program SW 1 OFF the final sum will be typed out.

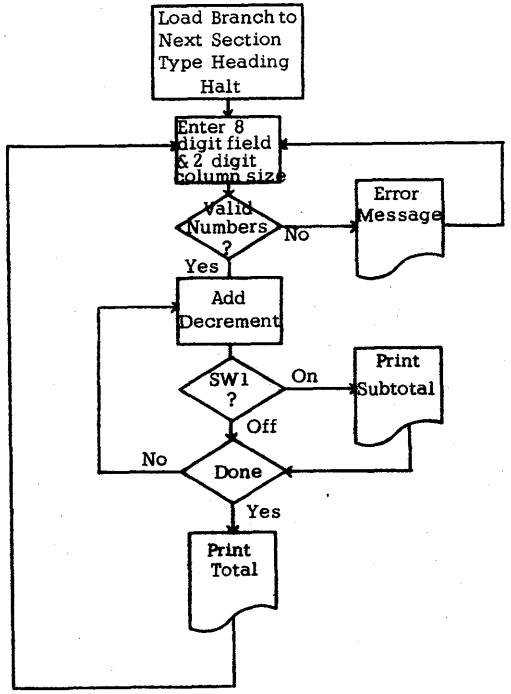
7) With Program SW 1 ON each subtotal will be printed and then the final total.

8) Point out the use of Program Switches to show how a basic program can be altered to give different results.

9) Enter an eight digit number with leading zeros and point out that the program tests for high order zeros and eliminates them. This will not be done in Section VI.

10) In case of typing errors see Sec.III, Part E 20.

F.



28

SECTION V - SIMPLE LOGICAL DECISION

A. Purpose - to illustrate

1) The ability to make a selection based on an input number code

2) The ability to recognize input errors

3) Typewriter entry error procedure

B. Input - numbers from 0 to 99 entered on console typewriter

C. Output

1) Spells out the name of the number entered

2) Error message if more than two digits are entered

D. Processing

1) Test for the number of digits entered.

2) Branch to one of three loops testing for 0-9, 10-19, 20-90.

3) If 20-90, we next go through the 0-9 loop.

E. Demonstration Procedure and Significant Points

1) Depress START.

2) Enter message is typed out and the typewriter is selected to the numeric mode.

3) The operator or a member of the group enters one or two digits.

4) Explain that the computer will now use this information to find the location where the number name is stored and after finding it will type it out.

5) Depress RELEASE and START or R-S key on typewriter.

6) The typewriter tabs, the name is printed, and the carriage returns.

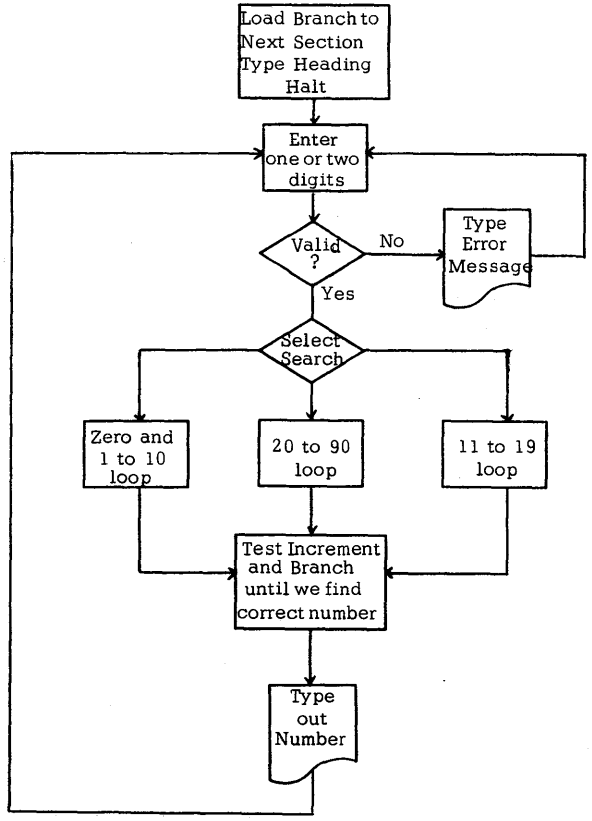
29

7) To illustrate an error condition, enter three digits and then no digits.

8) See Sec.III, Part E 20 for typewriter input error condition.

9) If no one has already done so enter the number 13 before going on to the next section.

F.



SECTION VI - SIMPLE ADD SPEED

A. Purpose - to illustrate

1) Calculation Speed (simple add)

2) Single Instruction Execute and the use of the console

3) The ability to change conditions for switch settings

B. Input - none

C. Output

1) The contents of the counter (6 positions with high order zeros) when switch two is changed after start is depressed

2) Error message if switch two is changed before start is depressed

D. Processing

1) Switch two setting initializes the test conditions for error message and addition completed.

2) One is added to a 6 digit counter.

E. Demonstration Procedure and Significant Points

1) Depress START.

2) Explain that the computer has looked at switch two and will continue to add one to the counter until the switch is changed. Point out that it is the change in switch two that stops addition, not just turning it ON or OFF.

3) Change switch two.

4) The contents of the counter are typed plus a comment.

5) Point out that this time, as opposed to section IV, the output number is not edited and high order zeros are typed.

6) To illustrate an error condition change switch one then depress START.

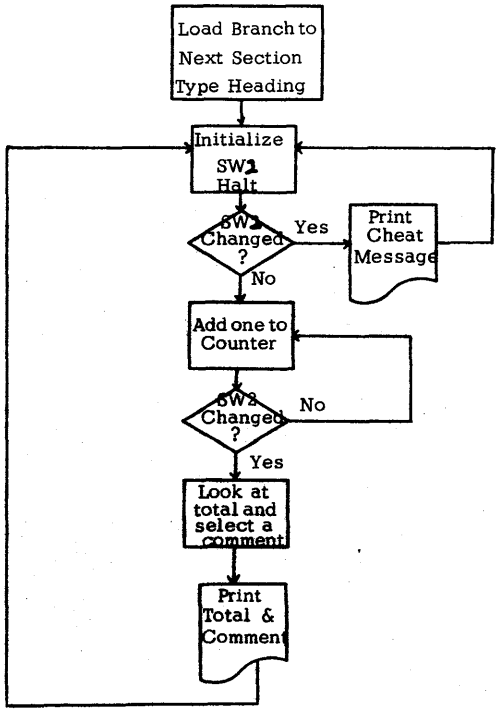
7) It is assumed that all members of the group will want to try their luck on this section.

8) To illustrate just what is occurring depress SIE to show how this section works, first the test to see if SW 2 has been changed too soon, then the addition of $\overline{01}$, then the test to see if SW 2 has been changed, then add, test, add, etc.

9) Point out how SIE is useful in debugging programs.

10) Show that by depressing SCE we can see just how the computer accesses and executes an instruction. Point out I cycles and E cycles.

F.



SECTION VII - SUCCESSIVE MULTIPLICATION

A. Purpose - to illustrate

1) Multiplication speed

2) Ability to develop floating point products

3) Ability to accept variable length input

4) Typewriter entry error correction

B. Input

1) NUMBER is printed on the typewriter and 1 to 7 digits may be entered. Leading zeros cause an error message to type out.

2) POWER is printed and a power from 1 to 99 may be entered.

C. Output

1) Since any number raised to the zero power is one, the program will type one whenever a zero power has been entered.

2) Since any number to the first power is that number itself, the program will type

Answer =

and then type the number that was entered.

3) Under any other conditions the program will develop a floating point product. An example of this output for twelve squared ($(12)^2$) is as follows:

Answer 0.14400000000000 times ten to the power 003

D. Processing

1) Input filled with record marks prior to entry

2) Test for illegal entry

a) No entry

b) Number greater than seven digits or power greater than two digits

c) Leading zero on number
(This restriction preserves the significant digits as the number is raised to successively higher powers in the floating point form.)

3) Tests for power of zero (See Output, Sec. C.1.)

4) Tests for power of one (See Output, Sec. C.2.)

5) Converts number into a floating point number and raises it to the power which was entered. It does so by successively multiplying the number by itself until the appropriate power is reached. The mantissa of the answer will always be 14 digits.

E. Demonstration Procedure and Significant Points

1) Instructions are typed if SW 3 is ON.

2) Depress START.

3) Number will type out and the 1620 will then be prepared to read from the typewriter in the numeric mode. Up to a 7 digit number may now be entered. Point out the 36 in the Op Register and the 1 in the Sense Branch Register indicating a read numeric from the typewriter.

4) Depress RELEASE and START.

5) POWER will type out and the 1620 will then be prepared to accept the entry of a power from 0 to 99.

6) SW 4 may be used for error correction from the typewriter as discussed in Section III, Part E 20. (For both number and power)

7) Depress RELEASE and START.

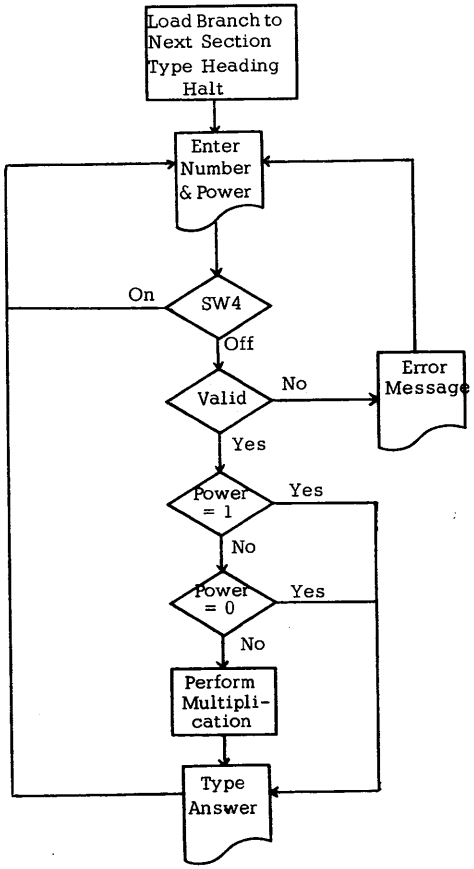
8) The answer will type out and the 1620 will halt.

9) Depress START to continue the multiplication demonstration.

10) Messages will type out on all the error conditions mentioned in Processing, Sec. D.2. The comment will change the second time either too long a number or too long a power is entered.

11) Demonstrate the difference in processing time between $(12)^2$ and $(9999999)^{99}$.

F.



SECTION VIII - COMBINATION OF EFFECTS

A. Purpose - to illustrate

1) Reading, processing and punching

2) Development of a numeric polynomial. This numeric polynomial is punched into a card.

3) Processing with prepared data

4) Testing for last card

5) Ability to alter processing via the sense switches

B. Input

Prepunched data cards. Columns 1 and 2 of each card is the number. Columns 4 and 5 represents the degree of the numeric polynomial which is to be developed based on the number. The other columns in the card are ignored.

C. Output

1) A heading card is punch out.

NUMBER POWER TRUNCATED ANSWER

2) The input number and power are punched in a card along with the answer and an X if the polynomial was truncated.

3) The card count, number of multiplications performed, and an end of job message will type out at the end of each job.

4) SW 1 ON stops reading and causes type out.

5) SW 2 ON causes 125 cards to be read and punched.

D. Processing

1) Develop a numeric polynomial, A, of the form

$$A = \sum_{i=1}^{PR} (NR)^i$$

where NR is from card columns 1 and 2 of data card and PR is from card columns 4 and 5.

For example: suppose NR = 2, PR = 4.

Then $A = 2 + (2)^2 + (2)^3 + (2)^4 = 30$

2) If the power entered in card columns 4 and 5 of the data card is greater than 15, the program will truncate the numeric polynomial. An X will be punched in card column 17 of the output card to signal the fact that the polynomial has been truncated. Thus the program will develop polynomials only through the 15th degree.

3) Punch an output card.

4) Test for SW 1, SW 2, and last card.

5) Type card count, number of multiplications performed, and end of job.

E. Demonstration Procedure and Significant Points

1) Depress PUNCH START, READER START, and START.

2) A heading card will punch.

3) Data cards will be read, the numeric polynomial developed, and the output punched out.

4) Turn SW 1 ON to stop the read/process/punch cycle.

5) Depress NON-PROCESS RUNOUT on punch unit but do not remove the two blank cards.

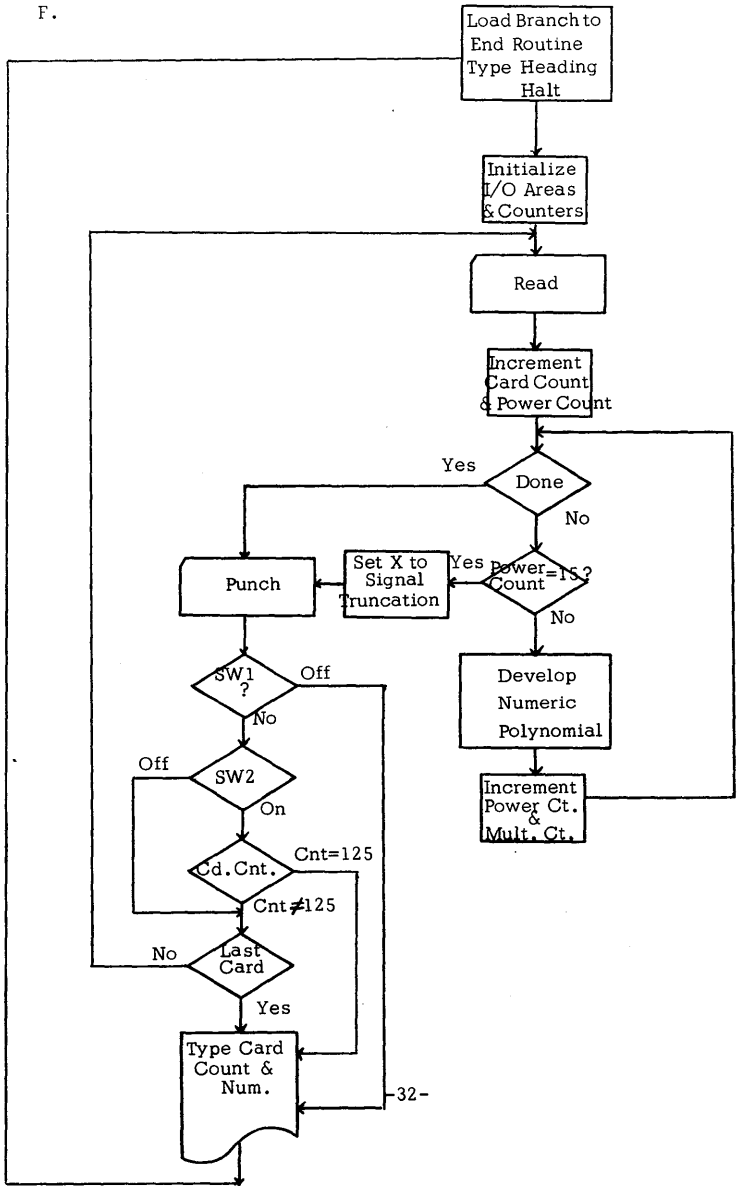
6) Repeat procedure with SW 2 ON.

7) 80/80 list the output cards.

8) Be sure to include a few data cards with a power greater than 15 (in columns 4 and 5) so that the program will truncate the polynomial and place an X in column 17 of the output card.

9) If a 402, 403 or 407 isn't conveniently available, the demonstrator could show a previously listed output from this input deck.

F.



OPERATING INSTRUCTIONS

Before the demonstration is given the operator must prepare the typewriter and the demo heading.

The typewriter should have the margins set at 15 and 90 and a tab stop at 50.

One card should be placed behind the deck, on it should be the name of the group to whom the demo is being given. This card may contain up to 60 columns of information and should be centered at card column 30.

If Switch 3 is ON operating instructions will be typed as the demonstration proceeds.

To load the program:

1. Depress RESET and INSERT.
2. Type in 160001000000
3. Depress RELEASE and START.
4. Depress INSTANT STOP and RESET.
5. Depress LOAD.

The first part of the program will load and a message will be typed:

SW 3 ON FOR PRE-DEMO INSTRUCTIONS

The operator should set SW 3 and depress START. SW 4 is normally OFF.

When the program is completely loaded the program halts. The operator should set the paper in the typewriter to a clean page and place a supply of cards in the card punch and card reader.

When the group arrives the operator depresses START and the demonstration begins. To repeat any section depress START. To go on to the next section depress RESET, INSERT, RELEASE and START.

The cards read in section one are not used and may therefore be punched or blank. The cards read in section 8 require input data, for the format see section 8, typical input is included as Deck II of this package.

All punched output is listable. Therefore, it is desirable to have a 407 with an 80-80 board ready so that punched output may be listed.

If the operator is familiar with DEMOPAK he may want to remove the pre-demo instructions. To do this remove the first twenty-five cards, the first card in the deck will then be numbered 200.

Any section may be deleted by removing that section's cards from the deck (see page 2) or after the program is loaded clear the flag from that section's thousand position.

The next five pages show a type out of the section headings first with SW 3 ON and then OFF.

40

1600010000000RS

SW3 ON FOR PRE-DEMO INSTRUCTIONS
MARGINS SHOULD BE SET AT 15 AND 90
SET ONLY ONE TAB STOP AT 50
PROGRAM CALLS FOR 1 DATA CARD, CC 1 TO 60 (CENTERED AT CC 30)
 CONTAINING NAME OF GROUP FOR WHOM DEMO IS BEING GIVEN
SW3 ON WILL GIVE INSTRUCTIONS AS YOU GO
ERROR MESSAGES WILL PRINT ON TYPEWRITER INPUT ERRORS
TURN TO CLEAN PAGE, PUSH START

41

1620 FUNCTIONAL DEMONSTRATION

FOR

(NAME OF GROUP GOES HERE)

DEMONSTRATING
ADDRESS FUNCTION
1000 CARD READING-PROGRAM AND DATA INPUT
2000 CARD OUTPUT
3000 VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O
4000 ADDITION OF A COLUMN OF FIGURES
5000 DIGITS TO ALPHABETIC
6000 RAPID SUCCESSIVE ADDITION
7000 SUCCESSIVE MULTIPLICATION
8000 COMBINATION OF EFFECTS

DEPRESS
START TO EXECUTE OR REPEAT A FUNCTION
RESET, INSERT, RELEASE, START TO EXECUTE THE NEXT FUNCTION
RESET AND INSERT 490X000 TO EXECUTE FUNCTION AT ADDRESS 0X000
PROG SW3 ON - INSTRUCTIONS, SW1, AND SW2 SETTINGS ARE PRINTED
PROG SW4 - TYPEWRITER ERROR CORRECTION

CARD READING-PROGRAM AND DATA INPUT

READ CARDS - SW1, SW2, OFF
STOP READING -- SW1, ON
READ 125 CARDS -- SW2, ON
I/O - PROGRAM OR STOP

CARD OUTPUT

PUNCH DATA CARDS CONTAINING RANDOM NUMBERS
SW1 ON TO STOP PUNCHING
SW2 ON TO PUNCH 125 CARDS
I/O TO STOP

VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O

SW1 ON SW2 OFF FOR NUMERIC
SW1 OFF SW2 ON FOR ALPHA-NUMERIC
TYPE IN UP TO 65 DIGITS OR LETTERS

ADDITION OF A COLUMN OF FIGURES

SW1 ON TO SHOW EACH SUBTOTAL
OVERFLOW TO STOP

DIGITS TO ALPHABETIC

SW1, SW2 -- ON OR OFF
OVERFLOW TO STOP
TYPE IN ANY TWO DIGIT NUMBER

RAPID SUCCESSIVE ADDITION

PUSH START THEN CHANGE SW 2 TO STOP ADDITION
OVERFLOW TO STOP

SUCCESSIVE MULTIPLICATION

TYPE UP TO 7 DIGITS
THEN TYPE A 2 DIGIT POWER

COMBINATION OF EFFECTS

SW2 ON READS 125 CARDS
SW1 ON STOPS READER
OFLOW TO STOP
LOAD DATA, PUSH START

END OF DEMONSTRATION

1600010000000

SW3 ON FOR PRE-DEMO INSTRUCTIONS

1620 FUNCTIONAL DEMONSTRATION

FOR

(NAME OF GROUP GOES HERE)

DEMONSTRATING

ADDRESS	FUNCTION
1000	CARD READING-PROGRAM AND DATA INPUT
2000	CARD OUTPUT
3000	VARIABLE LENGTH, ALPHA OR NUMERIC TYPEWRITER I/O
4000	ADDITION OF A COLUMN OF FIGURES
5000	DIGITS TO ALPHABETIC
6000	RAPID SUCCESSIVE ADDITION
7000	SUCCESSIVE MULTIPLICATION
8000	COMBINATION OF EFFECTS

CARD READING-PROGRAM AND DATA INPUT

CARD OUTPUT

VARIABLE LENGTH, ALPHA OR NUMERIC TYPEWRITER I/O

ADDITION OF A COLUMN OF FIGURES

DIGITS TO ALPHABETIC

RAPID SUCCESSIVE ADDITION

SUCCESSIVE MULTIPLICATION

COMBINATION OF EFFECTS

END OF DEMONSTRATION

TO ADD A SECTION

It is recognized that in order to make the demonstration more meaningful to certain groups, an additional section or sections should be added.

When adding a section certain procedures must be observed:

1) A section must start at an even thousands position, one to nine.

2) A section may not use more than 1800 storage positions, see table page 2.

3) A section must have a flag at its thousands position.

4) The NEXT subroutine must be used.

5) The section must be written in 1620/1710 SPS.

6) To include the section in the deck remove the first two and the last seven cards from the SPS condensed deck and insert it between Section 8 (card 8047) and the math tables (card 400).

A typical section might start something like this:

```
            DORG    9000
            B       BEGIN,0103,08,SET FLAG OVER 9000
HEADIN      DAC     17,AN EXTRA SECTION@
BEGIN       BTM     NEXT,9000,,SET UP LINKAGE TO END OF
                    DEMONSTRATION
            BTM     ALPHA,RCTY+2,,RETURN CARRIAGE THREE
                    TIMES
            BTM     ALPHA,HEADIN,,PRINT HEADING
            BNC 3   HALT,,,BRANCH AROUND OPERATING
                    INSTRUCTIONS
            BTM     ALPHA,INSTR,,TYPE OPERATING
                    INSTRUCTIONS
HALT        H
RCTY        DC      4,0003,HALT+11
NEXT        DS      ,17608
ALPHA       DS      ,00694
```

The writer, of course, should read over the section on the subroutines very carefully and take advantage of them.

The title of the section must have its high order address at location 9012 and 13 so that it will be printed by the demonstration heading subprogram. The field in position 9011 (i.e. 0103) is used in the Alpha subprogram. The heading may be any length.

DEMONSTRATION NARRATIVE

Prior to demo

Margin and tabs are set and a new page is ready for typing. The program has been loaded and the read hopper is full of blank or punched cards. A demo started at this point and using a narrative as follows, can be given in 30 to 40 minutes.

Introduction

This is the IBM 1620 computer. Stored in its memory at this moment is a series of instructions called a stored program. These instructions are designed to demonstrate to you the various functions of this computer system.

When I push START we will begin to execute the instructions in sequence. The first portion of this program is typing out a title for our demonstration. The system is now typing out what it will demonstrate, including the starting location for each program section. For example, starting in computer location 1000 there is a program to demonstrate card reading. In location 2000 a card output program begins, and so on. The typewriter is now typing 7000, a program to show multiplication speed. And starting at 8000, there is a section to show the various functions working together on an actual problem.

```
          1620 FUNCTIONAL DEMONSTRATION
                        FOR
          BASIC COMPUTER SYSTEMS CLASS

DEMONSTRATING
ADDRESS   FUNCTION
   1000   CARD READING-PROGRAM AND DATA INPUT
   2000   CARD OUTPUT
   3000   VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O
   4000   ADDITION OF A COLUMN OF FIGURES
   5000   DIGITS TO ALPHABETIC
   6000   RAPID SUCCESSIVE ADDITION
   7000   SUCCESSIVE MULTIPLICATION
   8000   COMBINATION OF EFFECTS
```

The program is also designed to type out a series of instructions to the operator. These instructions will be typed when this switch (Sw 3) is in the ON position.

45

```
DEPRESS
   START TO EXECUTE OR REPEAT A FUNCTION
   RESET, INSERT, RELEASE, START TO EXECUTE THE NEXT FUNCTION
   RESET AND INSERT 490X000 TO EXECUTE FUNCTION AT ADDRESS 0X000
PROG SW3 ON - INSTRUCTIONS, SW1, AND SW2 SETTINGS ARE PRINTED
PROG SW4 - TYPEWRITER ERROR CORRECTION
```

Card Reading

We have now arrived at the portion of program which demonstrates the reading of cards. The typewriter is busy typing out a set of instructions for this section of the program, and finally the computer stops in a manual status. When I depress START it will begin to read cards at the rate of 250 cards a minute.

```
CARD READING-PROGRAM AND DATA INPUT

READ CARDS -- SW1, SW2, OFF
STOP READING -- SW1, ON
READ 125 CARDS -- SW2, ON
I/O - PROGRAM OR STOP
```

While the computer is busy reading cards, I would like to point out the various features of the 1620 Data Processing System. You see in front of you the console of the machine. On the console there is a row of keys which allow the operator to interrupt the automatic sequence of events in the program and to communicate with the computer by using the typewriter. The operator may be interested in looking into the memory of the machine or in changing the sequence of program steps.

Above this row of keys is a set of lights that indicate the status of the machine at any moment. Right now we see that the machine is in the automatic mode, which means it is following a sequence of program steps. Depressing this key (SIE) will place the computer in a manual mode -- notice the "MANUAL" light. -- This will allow the operator to execute each instruction individually. We return to the automatic mode by pushing START. The computer can also be programmed to stop in the manual mode. This "MANUAL" light, then, indicates to the operator that the execution of the program has come to a halt and operator attention is required to continue.

This "READER NO FEED" light indicates that the program has come to a card input command and that the computer is waiting for the card to mechanically pass through the card reader. Since the reader contains a buffer, reading and computing can occur simultaneously. If there is a large computation job between cards then the computer would probably not wait for the reader.

47

The lights here on the face of the machine indicate certain status to the operator (OP Register, MARS). Normally he will be interested in these lights when he has stopped the automatic operation of the program. For example, this particular register, the operation register, tells the operator which machine language instruction is being executed at any particular moment. During an input operation, such as we see now, the operation register will indicate an input type command (point to 37). These lights appear very bright because the computer is waiting for the mechanical operation of card reading. Under normal computations in the automatic mode these lights will be very dim and the computer will be executing commands in sequence at electronic speed.

An operator will also find this panel marked "Memory Address Register" extremely useful. This register will always indicate the address or memory location from which the computer is taking information. In this way it can be used to indicate to the operator where the next instruction is coming from or the location of data being used by any instruction.

The memory unit of the computer, is located just behind this console. It's a very small box, about a foot cube, but large enough to store 20,000 digits of information. The remainder of the space is occupied by electronic circuitry for addressing any particular digit and for executing the instructions of the machine.

Both the sequential program instructions and the data are stored in this memory unit. Each instruction is made up of 12 digits, which are accessed from the memory and executed in sequence. This sequential execution of instructions can be changed by a certain type of instruction known as "Test and Branch" commands. These are the commands we use to build logic into the program.

I see the computer has read a thousand cards. It has been counting the cards read as I talked, has typed out a message, and stopped in MANUAL mode.

```
STOP, STOP, STOP 1000 CARDS READ
```

To show you that the program has actually been counting, I would like to repeat this section. You see, each segment of this demonstration program is designed so that the operator can return to the beginning.

At any point I can alter the program by testing a switch, which appears here on the console. When I turn this switch ON the program will then branch to a different section and it can tell me how many cards have been read up to this point. While I have been talking we have read 67 cards, and I see we have a message from the program saying

```
67 CARDS READ, LITTLE JOB (NICE)
```


50

Let's try it again.

```
174 CARDS READ, GOOD READER,RIGHT
```

This time to demonstrate that the reader is reading at a rate of 250 cards a minute, I would like to have you check the time required to read 125 cards. (Depress START). We have on the typewriter at the moment a message saying

```
NOTE THE TIME AT START

125 CARDS READ, 30 SEC. ON THE NOSE
```

Let's go on to punched output.

Punched Output

By depressing some keys on the console I will be able to enter the next section of the program which will demonstrate card output.

```
CARD OUTPUT

PUNCH DATA CARDS CONTAINING RANDOM NUMBERS
SW1 ON TO STOP PUNCHING
SW2 ON TO PUNCH 125 CARDS
I/O TO STOP
```

When I depress START, the program will begin punching data cards containing random numbers. Again we will use a switch to stop the operation and we will have the opportunity to time the output (Depress START).

While these cards are punching we are also doing some multiplication and other manipulations to develop the random numbers which are being punched on the card. Later we can take these cards to a listing machine such as the 407 and print the output. Let's see how many cards we have punched.

```
90 DATA CARDS PUNCHED
```

To demonstrate that the punch is punching at the rate of 125 cards per minute, I will turn Switch 2 ON and depress START. Apparently I have violated an instruction and receive a message.

```
I FIND IT CONFUSING WHEN YOU HAVE BOTH SW1 AND SW2 ON, START AGAIN
```

So for this portion of the program I must have only either Sw 1 or Sw 2 ON, but not both. Time ready.

71

The 1620 card read punch contains what we call a buffer. This means that the data from the processing unit can be transmitted to the card read punch at electronic rates and the computer may continue calculations while the mechanical action of punching is taking place. This allows overlapping of computing and input/output operation. Therefore, we are doing not only computations at this time, but we are punching cards at the maximum rate of the punch. We have been punching for exactly one minute and the typewriter tells us we have punched 125 cards.

```
125 DATA CARDS PUNCHED
```

Typewriter I/O

Now let's go to the next portion of program which demonstrates the typewriter as in input/output device. We are also interested in demonstrating the fact that the data within this machine can be variable in length. If it requires three positions we can reserve three positions for each field. If the data requires 25 positions, we can program the machine for data 25 digits long. Regardless of the problem, we can make the most effective use of our storage.

```
VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O


SW1 ON   SW2 OFF  FOR NUMERIC
SW1 OFF  SW2 ON   FOR ALPHA-NUMERIC
TYPE IN UP TO 65 DIGITS OR LETTERS
```

First, let me demonstrate some numeric input/output. The computer has now selected the typewriter and is waiting for me to enter information. Notice that the keyboard has been shifted for numeric input. I must also tell the computer that I'm through entering information so I hit a key on the typewriter called "RELEASE START". Under program control, the typewriter tabulates, types out a name AO1 and prints back for me what was stored by typewriter input.

```
123456789                          AO1 = 123456789

12                                 AO2 = 12
```

Now I will enter information which is longer than half a page. Watch this console register (MAR) as I type. In the BCD code (explain) we see that the first digit enters memory location 19841. The second goes into location 19842. The third to 43 and etc.

```
22222222222222222222222222222222222222222222222222

AO3 = 22222222222222222222222222222222222222222222222222
```


The program recognizes that we have entered more information than can be printed on half a page width. Therefore a program step returns the carriage rather than performing a tabulation. So the program recognized the fact that the field we entered was longer than half a page width.

Now let's try some alphabetic information. I will turn switch 1 OFF and switch 2 ON. The typewriter is now selected in the alphanumeric mode. Why don't you enter your name.

```
RAY PDCK
```

You made a mistake. That's OK. I will place switch 4 ON, then depress RS and we can re-enter the correct information.

```
RAY PECK                           AO4 = RAY PECK

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID
AO5 = NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID
```

Addition

Let's proceed to the next section. The 1620 and I are now prepared to demonstrate the addition of a column of figures. It's typing out input instructions.

```
ADDITION OF A COLUMN OF FIGURES

SW1 ON TO SHOW EACH SUBTOTAL
OVERFLOW TO STOP
ENTER AN EIGHT DIGIT NUMBER
12345678
ENTER COLUMN SIZE, TWO DIGITS

22
  271604916
```

There's the result. Let's give it something a little bit more strenuous to do. How about all nines?

```
ENTER AN EIGHT DIGIT NUMBER
99999999
ENTER COLUMN SIZE, TWO DIGITS

99
 9899999901
```

Let's see if it's doing it correctly. Let's add 9 to itself three times. Apparently, it must be an 8 digit number. There I entered a nine with leading zeros. Now 03. Correct.

```
ENTER AN EIGHT DIGIT NUMBER
9
THIS JOB IS TOO SMALL, START AGAIN
ENTER AN EIGHT DIGIT NUMBER
09
ENTER COLUMN SIZE, TWO DIGITS

03

   27
```

By turning switch 1 on, I can see the intermediate sub-totals as we go along. Let's take a big number and add it to itself five or six times. This illustrates that it's actually doing the addition.

```
ENTER AN EIGHT DIGIT NUMBER
12345678
ENTER COLUMN SIZE, TWO DIGITS

06
   24691356
   37037034
   49382712
   61728390
   74074068

   74074068
```

Let me show it to you again at electronic speed.

```
ENTER AN EIGHT DIGIT NUMBER
12345678
ENTER COLUMN SIZE, TWO DIGITS

22

   271604916
```

Logic

In this section we will demonstrate that a computer can perform logical decisions. It says "Type in any two digit number". Would you (one of audience) like to enter a number. Try some more.

```
DIGITS TO ALPHABETIC
SW1, SW2 -- ON OR OFF
OVERFLOW TO STOP
TYPE IN ANY TWO DIGIT NUMBER
10                              TEN
18                              EIGHTEEN
99                              NINTY NINE
84                              EIGHTY FOUR
40                              FORTY
```


Try entering a 1. Ah, the program is designed to detect single digit numbers. Zero? How about 13? Perhaps the machine is superstitious. That's what I thought. What happens with a three digit number. An error message requesting two digits. I wonder what happens if we neglect to enter information.

```
1                               ONE
0                               ZERO
13                              I DO NOT LIKE THE NUMBER THIRTEEN
123                             ENTER ONLY TWO CHARACTERS PLEASE
                                GO AHEAD AND ENTER TWO DIGITS
23                              TWENTY THREE
56                              FIFTY SIX
39                              THIRTY NINE
```

So the computer has the ability to examine a number in its memory, make a logic decision based on that number, find the correct alphabetic words stored in its memory and print them out in their proper order.

Successive Addition (Depress START)

Again, we take the opportunity to show the speed of the 1620. The computer has examined switch 2 and will continue to add a 1 to a counter until I change the status of switch 2. Notice that the operation register contains an 11 which is the code for addition. However, it is executing two commands in what we call a loop. After adding one to the counter, it will test the status of switch 2 and return to the add instruction. When I turn switch 2 ON, the operation will stop and the contents of the counter will be printed out. How many additions have we done? What message did I get? Let's test the reaction of a few of you. Depress START, then with the same hand change switch 2.

```
RAPID SUCCESSIVE ADDITION

PUSH START THEN CHANGE SW2 TO STOP ADDITION
OVERFLOW TO STOP
017773  FASTER FASTER FASTER
001622  YOU ARE TOO SLOW
000417  SPEED IT UP CHARLIE
000181  GET THE LEAD OUT
```

Maybe I can get a good score by changing the switch first.

```
 YOU MOVED SW2 TOO SOON

000198  GET THE LEAD OUT

000037  TRY IT ONE MORE TIME
```

Ha! I'm gettin' good.

```
SW2 TOO SOON
```

Let's see what else the program has to say when Sw 2 is moved before START is depressed.

```
PLAY FAIR
WATCH THAT
DIRTY PLAYER
DONT CHEAT
```

If I move the switch and depress START at the same time I can do pretty well.

```
000080  YOU CAN IMPROVE THIS
000041  TRY IT ONE MORE TIME
000062  FASTER ON THE SWITCH
000041  TWO DIGITS IS FAST
000054  YOU CAN IMPROVE THIS
```

If I use the single instruction Execute Key (SIE) I can do much better. In the OPeration Register you see the step which tests Sw 2 and branches to the "Cheat" messages if it has been changed. Now you see the 11, an add, followed by a test of Sw 2, then an add, etc. (change Sw 2, depress START).

```
000008  THAT WAS FAST ON SW2
```

By going through the loop only one time I can really get a good comment.

```
000001  YOU ARE NOW AN EXPERT
```

Multiplication

Here we will demonstrate the successive multiplication of a number containing up to seven digits. The number will be multiplied successively times itself according to a 2 digit input number or "Power". Let's try 99 cubed.


```
SUCCESSIVE MULTIPLICATION

TYPE UP TO 7 DIGITS
THEN TYPE A 2 DIGIT POWER
NUMBER 99

POWER 3

ANSWER 0.97029900000000  TIMES TEN TO THE POWER 006
```

Notice that the answer is printed in what we call floating point form. Using this technique we can work with numbers of any size. The result is printed as a decimal x some power of 10. Here the .970,299 x 10^6 is another way of writing nine hundred seventy thousand two hundred ninety nine. We have 6 significant digits. 99 x 99 gives us a 4 place result, which when multiplied by 99 gives us 6 places.

Let's try one with a simple answer. Give me a number. 12. Now, a power. 2.

```
NUMBER 12

POWER 2

ANSWER 0.14400000000000  TIMES TEN TO THE POWER 003
```

The answer is .144 x 10^3 or 144. Here's something more strenuous.

```
NUMBER 9999999

POWER 99

ANSWER 0.99999010004754  TIMES TEN TO THE POWER 693
NUMBER
```

The computer just performed 99 multiplications of a seven digit number. Notice that this number would have 693 significant digits, but we are keeping only 14. Each successive multiplication, then, consists of the 7 digit input number times the 14 digit result of former multiplication.

Since this machine uses a variable word length concept we could develop all 693 digits of this multiplication if required by our problem. In this way a variable word length machine allows you to carry out your arithmetic to any required degree of accuracy.

Polynomial Solution

In this section we will demonstrate the solution of a complete problem. We will input a card containing two values, the variable called NR and the degree of a polynomial called PR. The polynomial is of the form

$NR + NR^2 + NR^3 + NR^4 \ldots$ etc. on up to NR^{PR}.

COMBINATION OF EFFECTS

SW2 ON READS 125 CARDS
SW1 ON STOPS READER
OFLOW TO STOP
LOAD DATA, PUSH START

When I push START we will read an input card, perform the multiplications and additions of the polynomial and punch the result.

The program is designed to calculate all polynomials to the 15th degree or less. If the degree is greater than 15, the punched result is for the 15th degree problem and the output is flogged by an X in column 17.

On the console you see the flashing of lights that is typical of the 1620 operating at electronic speed.

NUMBER OF CARDS READ 158

NUMBER OF MULTIPLICATIONS PERFORMED 1799

END OF JOB, LOAD DATA AND PUSH START

We have read the last data card and typewriter has given us the number of multiplications performed.

For this next run I will turn switch 2 to the ON position so that the computer will stop after 125 cards have been read. Watch your watches. (Depress START). Since the reader and the punch are buffered we should be able to overlap all of the processing with the reading and punching. We should, therefore, punch at the maximum rate of 125 cards per minute and still perform all of the computations. If it takes more than one minute to punch 125 cards, then some of the computations require more time than is required to punch a card.

NUMBER OF CARDS READ 125

NUMBER OF MULTIPLICATIONS PERFORMED 1360

END OF JOB, LOAD DATA AND PUSH START

How did we do?

Conclusion

END OF DEMONSTRATION

That was an introduction to the IBM 1620 computer and a relatively simple demonstration of some typical computer functions. The kind of logic, the Input Output capabilities and the computation speeds demonstrated here can be put to use to solve both the simple and the complex engineering, scientific, mathematical or data processing problems of your company. Experience has proven that the 1620 computer in any or all of these roles produces money saving results.

SAMPLE OUTPUT

1620 FUNCTIONAL DEMONSTRATION

FOR

RAY PECK, BILL OLMO AND DAVE MONTGOMERY

```
DEMONSTRATING
ADDRESS   FUNCTION
   1000   CARD READING-PROGRAM AND DATA INPUT
   2000   CARD OUTPUT
   3000   VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O
   4000   ADDITION OF A COLUMN OF FIGURES
   5000   DIGITS TO ALPHABETIC
   6000   RAPID SUCCESSIVE ADDITION
   7000   SUCCESSIVE MULTIPLICATION
   8000   COMBINATION OF EFFECTS

DEPRESS
  START TO EXECUTE OR REPEAT A FUNCTION
  RESET, INSERT, RELEASE, START TO EXECUTE THE NEXT FUNCTION
  RESET AND INSERT 490X000 TO EXECUTE FUNCTION AT ADDRESS 0X000
PROG SW3 ON - INSTRUCTIONS, SW1, AND SW2 SETTINGS ARE PRINTED
PROG SW4 - TYPEWRITER ERROR CORRECTION


CARD READING-PROGRAM AND DATA INPUT

READ CARDS - SW1, SW2, OFF
STOP READING  --  SW1, ON
READ 125 CARDS --  SW2, ON
I/O - PROGRAM OR STOP

STOP, STOP, STOP 1000 CARDS READ

  27 CARDS READ, LITTLE JOB (NICE)

  43 CARDS READ, LET ER RUN CHARLIE

 124 CARDS READ, GOOD READER,RIGHT

   8 CARDS READ, GOOD READER,YES

  12 CARDS READ, TRY TIMING, SW 2 ON

   9 CARDS READ, TRY TIMING, SW 2 ON

   7 CARDS READ, TRY TIMING, SW 2 ON
```


60

```
NOTE THE TIME AT START

 125 CARDS READ, 30 SEC. ON THE NOSE


CARD OUTPUT

PUNCH DATA CARDS CONTAINING RANDOM NUMBERS
SW1 ON TO STOP PUNCHING
SW2 ON TO PUNCH 125 CARDS
I/O TO STOP

   12 DATA CARDS PUNCHED

    7 DATA CARDS PUNCHED

I FIND IT CONFUSING WHEN YOU HAVE BOTH SW1 AND SW2 ON, START AGAIN

   10 DATA CARDS PUNCHED

  125 DATA CARDS PUNCHED


VARIABLE LENGTH ALPHA OR NUMERIC TYPEWRITER I/O


SW1 ON  SW2 OFF FOR NUMERIC
SW1 OFF SW2 ON  FOR ALPHA-NUMERIC
TYPE IN UP TO 65 DIGITS OR LETTERS

COME,COME,TURN ON EITHER SW1 OR SW2

1111122222333334444455555666RS      A01 = 1111122222333334444455555666

1111122222333334444455555666RS

A02 = 1111122222333334444455555666

MAKE UP YOUR MIND,TURN OFF EITHER SW1 OR SW2

I AM AN IBM 1620RS                  A03 = I AM AN IBM 1620

HOW DO LIKE THIS FOR A DEMONSTRATION OF THE 1620RS

A04 = HOW DO LIKE THIS FOR A DEMONSTRATION OF THE 1620

THIS IS AN EXAMPLE OF ERROR CORRECTION, SW 4 ON THEN RS

THIS IS A NEW MESSAGE, SW 4 OFF AND THIS WILL PRINTRS

A05 = THIS IS A NEW MESSAGE, SW 4 OFF AND THIS WILL PRINT
```


51

ADDITION OF A COLUMN OF FIGURES

SW1 ON TO SHOW EACH SUBTOTAL
OVERFLOW TO STOP
ENTER AN EIGHT DIGIT NUMBER
11111111RS
ENTER COLUMN SIZE, TWO DIGITS
10RS
22222222
33333333
44444444
55555555
66666666
77777777
88888888
99999999
111111110

111111110
ENTER AN EIGHT DIGIT NUMBER
00001234RS
ENTER COLUMN SIZE, TWO DIGITS
12RS
2468
3702
4936
6170
7404
8638
9872
11106
12340
13574
14808

14808
ENTER AN EIGHT DIGIT NUMBER
11111111RS
ENTER COLUMN SIZE, TWO DIGITS
10RS

111111110
ENTER AN EIGHT DIGIT NUMBER
123456789RS
I SAID ONLY EIGHT DIGITS
ENTER AN EIGHT DIGIT NUMBER
12345678RS
ENTER COLUMN SIZE, TWO DIGITS
5RS
THIS JOB IS TOO SMALL, START AGAIN

ENTER AN EIGHT DIGIT NUMBER
12345678RS
ENTER COLUMN SIZE, TWO DIGITS
05RS

61728390

DIGITS TO ALPHABETIC

SW1, SW2 -- ON OR OFF
OVERFLOW TO STOP
TYPE IN ANY TWO DIGIT NUMBER

0RS	ZERO
00RS	ZERO
6RS	SIX
06RS	SIX
8RS	EIGHT
08RS	EIGHT
123RS	ENTER ONLY TWO CHARACTERS PLEASE
12RS	TWELVE
14RS	FOURTEEN
29RS	TWENTY NINE
RS	GO AHEAD AND ENTER TWO DIGITS
99RS	NINTY NINE
83RS	EIGHTY THREE
13RS	I DO NOT LIKE THE NUMBER THIRTEEN

RAPID SUCCESSIVE ADDITION

PUSH START THEN CHANGE SW 2 TO STOP ADDITION
OVERFLOW TO STOP

018890 FASTER FASTER FASTER

015183 YOU ARE TOO SLOW

009612 FASTER ON SWITCH2, TO REDUCE SUM

003696 SPEED IT UP CHARLIE

004441 THE 1620 IS FASTER THAN YOU ARE

003971 GET THE LEAD OUT

004875 I KNOW YOU CAN DO BETTER CHARLIE

004113 FASTER ON SWITCH2 TO REDUCE SUM

002738 FASTER FASTER FASTER

000485 YOU ARE TOO SLOW

000490 SPEED IT UP CHARLIE

000227 GET THE LEAD OUT

000016 YOU CAN IMPROVE THIS

000021 TRY IT ONE MORE TIME

000025 FASTER ON THE SWITCH

000018 TWO DIGITS IS FAST

YOU MOVED SW2 TOO SOON

SW2 TOO SOON

PLAY FAIR

WATCH THAT

DIRTY PLAYER

DONT CHEAT

YOU MOVED SW2 TOO SOON

000003 THAT WAS FAST ON SW 2

000005 YOU ARE NOW AN EXPERT

000001 GOOD SHOW OLD BOY

000001 THAT WAS FAST ON SW 2

000002 YOU ARE NOW AN EXPERT

031517 FASTER FASTER FASTER

003644 YOU ARE TOO SLOW

64

SUCCESSIVE MULTIPLICATION

TYPE UP TO 7 DIGITS
THEN TYPE A 2 DIGIT POWER
NUMBER 12RS

POWER 2RS

ANSWER 0.14400000000000 TIMES TEN TO THE POWER 003

NUMBER 10RS

POWER 99RS

ANSWER 0.10000000000000 TIMES TEN TO THE POWER 100

NUMBER RS

GO AHEAD

NUMBER 9999999RS

POWER 99RS

ANSWER 0.9999910004754 TIMES TEN TO THE POWER 693

NUMBER 12345678RS

ENTER ONLY SEVEN DIGITS HOW OFTEN MUST I TELL YOU

NUMBER 9RS

POWER 123RS

THE POWER MAY NOT EXCEED TWO DIGITS

POWER 123RS

THE POWER MAY NOT EXCEED TWO DIGITS CHARLIE

POWER 99RS

65

ANSWER 0.2951266543O603 TIMES TEN TO THE POWER 095

NUMBER 0123456RS

NO HIGH ORDER ZERO PLEASE

NUMBER 123456RS

POWER 1RS

ANSWER = 123456

NUMBER 123456RS

POWER ORS

ONE

NUMBER 1111111RS

POWER 25RS

ANSWER 0.13929520867090 TIMES TEN TO THE POWER 152

COMBINATION OF EFFECTS

SW2 ON READS 125 CARDS
SW1 ON STOPS READER
OFLOW TO STOP
LOAD DATA, PUSH START

NUMBER OF CARDS READ 8

NUMBER OF MULTIPLICATIONS PERFORMED 54

END OF JOB, LOAD DATA AND PUSH START

NUMBER OF CARDS READ 21

NUMBER OF MULTIPLICATIONS PERFORMED 258

END OF JOB, LOAD DATA AND PUSH START

NUMBER OF CARDS READ 125

NUMBER OF MULTIPLICATIONS PERFORMED 1406

END OF JOB, LOAD DATA AND PUSH START

END OF DEMONSTRATION

THIS SECTION GENERATES A TABLE OF RANDOM NUMBERS

1	92114	349.490
2	21016	352.917
3	94726	142.714
4	4370	242.520
5	41659	618.923
6	74512	871.733
7	82333	55.766
8	78338	429.050
9	9327	77.824
10	62092	264.623
11	29009	86.641
12	27649	967.407

THIS SECTION GENERATES A TABLE OF RANDOM NUMBERS

1	63804	378.349
2	10368	331.175
3	74733	395.192
4	40490	584.327
5	58798	557.854
6	68911	907.696
7	76980	693.300

THIS SECTION GENERATES A TABLE OF RANDOM NUMBERS

1	39873	115.255
2	55561	983.753
3	40476	656.638
4	93675	549.219
5	10324	47.721
6	52122	952.450
7	86550	589.318
8	81781	586.781
9	25398	793.782
10	5178	581.157

THIS SECTION GENERATES A TABLE OF RANDOM NUMBERS

1	14305	123.948
2	36133	609.736
3	39028	226.888
4	91462	240.160
5	3165	523.630
6	313	586.724
7	70786	841.894
8	64252	381.749
9	11762	979.738
10	23057	254.027
11	55985	774.942
12	22940	602.761
13	70024	166.685
14	77424	568.031
15	8682	849.912
16	95640	744.056
17	69296	854.651
18	74243	321.077
19	84758	774.630
20	67373	170.449
21	53820	969.179
22	10223	958.197
23	98558	950.755
24	60493	969.956
25	23373	664.576
26	6521	306.598
27	9634	986.537
28	72924	68.730
29	93116	298.932
30	5812	574.090
31	31823	413.205
32	61126	690.913
33	11316	134.245
34	58795	212.433
35	45710	906.181
36	225	465.511
37	93711	49.061
38	96064	767.183
39	98305	560.688
40	14190	609.283
41	77340	463.020

42	52125	707.044
43	52352	943.877
44	5668	152.650
45	39034	658.494
46	78179	138.436
47	25463	156.790
48	96969	920.242
49	31348	918.012
50	39678	33.263
51	47073	223.504
52	54440	138.480
53	5512	371.580
54	38838	465.967
55	13270	712.875
56	22904	547.271
57	10356	340.304
58	45831	299.288
59	96731	279.991
60	68107	363.286
61	35276	725.367
62	91582	905.148
63	1276	660.032
64	98140	432.698
65	55552	721.659
66	78832	899.799
67	16051	473.287
68	85206	961.149
69	25616	528.987
70	58403	512.114
71	79171	90.567
72	9224	298.001
73	62868	570.012
74	59759	915.190
75	44555	52.397
76	74970	498.377
77	20866	533.853
78	39678	299.791
79	90409	828.730
80	37293	376.803
81	35330	515.877
82	41012	87.044

- 64 -

70

83	27796	928.099
84	28213	94.603
85	32606	622.622
86	83287	931.725
87	19214	633.635
88	83448	773.750
89	76259	870.780
90	1788	971.750
91	53684	375.692
92	18461	408.328
93	20446	422.953
94	63833	486.691
95	71841	848.544
96	5953	253.695
97	5160	90.620
98	82320	688.783
99	68949	921.878
100	79667	358.438
101	10787	722.205
102	21580	952.015
103	21874	469.980
104	24281	677.167
105	23197	537.615
106	92731	724.294
107	12160	495.715
108	84032	251.108
109	67685	267.342
110	86242	946.432
111	38532	317.696
112	8484	520.261
113	33636	157.129
114	59951	881.054
115	51198	236.735
116	61137	865.294
117	49681	104.624
118	39656	998.813
119	8909	807.961
120	71129	922.632
121	20572	925.404
122	35241	422.197
123	80033	41.360
124	44088	322.924
125	28693	226.132

- 65 -

71

NUMBER	POWER	TRUNCATED	ANSWER
2	5		62
6	6		55986
2	5		62
2	6		126
10	10		11111111110
11	10		28531167060
12	10		67546215516
13	10		149346699502

NUMBER	POWER	TRUNCATED	ANSWER
14	10		311505013050
15	10		617839704240
16	10		1172812402960
99	99	X	868834460300893509841851440799
15	99	X	469172025408063615
3	66	X	21523359
74	74	X	1107604357274159198243841997 4
85	85	X	883941502810285086513700939 35
96	96	X	54779255228049757475828437 1040
3	10		88572
24	25	X	526807599606308980824
26	26	X	1744349715977154962390
28	27	X	5286457405432405435884
45	78	X	64261009523282172463156 95
12	17	X	16807659899548764
18	18	X	7143501829211426574
78	45	X	24379394659122672740198893434
2	2		6
56	89	X	17007706105645980566266 1240
18	18	X	7143501829211426574
13	12		25239592216020

NUMBER	POWER	TRUNCATED	ANSWER
9	5		66429
99	15		868834460300893509841851440799
99	45	X	868834460300893509841851440799
6	10		72559410
9	20	X	231627523606479
15	1		15
20	1		20

- 66 -

1	1		1
4	2		20
4	12		22369620
64	97	X	125758988117879900942133 2544
12	45	X	16807659899548764
6	3		258
16	15		1229782938247303440
15	16	X	469172025408063615
17	10		2141993519226
18	10		3780494710542
18	17	X	7143501829211426574
17	18	X	3041324492229179279
18	19	X	7143501829211426574
19	10		6471681049900
11	10		28531167060
4	3		84
20	20	X	34492631578947368420
21	20	X	71528434512099266415
20	21	X	34492631578947368420
27	28	X	30679401183412503793 59
30	30	X	14843696896551724137930
23	32	X	27875501889459084767 9
23	56	X	27875501889459084767 9
45	12		70519626362998268820
32	23	X	38997607084342876603424
12	11		810554586204
56	23	X	17007706105645980566266 1240
25	36	X	97012768189112345377 5
13	11		194150709 3539
74	14		14967626449650799976268 1350
12	12		97266550 34460
13	13		3281146988082 73
34	12		24587364619868313 90
21	43	X	71528434512099266415
54	78	X	98633464405394656724626914
23	23	X	27875501889459084767 9
25	24	X	97012768189112345377 5
67	45	X	24983478599430934075115636 79
14	14		11966776581370170
12	15		16807659899548764
54	76	X	98633464405394656724626914
96	78	X	54779255228049757475828437 1040
12	16	X	16807659899548764

- 67 -

15	17	X	469172025408063615
76	98	X	165179525077992479960963621 40
14	17	X	167534872139182394
13	17	X	55451384098598319
1	1		1
99	45	X	868834460300893509841851440799
4	12		22369620
4	12		22369620
1	1		1
1	1		1
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
1	1		1
4	12		22369620
99	45	X	868834460300893509841851440799
1	1		1
99	45	X	868834460300893509841851440799
4	12		22369620
4	12		22369620
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
3	66	X	21523359
6	10		72559410
6	10		72559410
6	10		72559410
6	10		72559410
99	45	X	868834460300893509841851440799
99	45	X	868834460300893509841851440799
4	12		22369620

99	45	X	868834460300893509841851440799
4	12		22369620
99	45	X	868834460300893509841851440799
6	10		72559410
6	10		72559410
6	10		72559410
6	10		72559410
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
99	45	X	868834460300893509841851440799
4	12		22369620
99	45	X	868834460300893509841851440799
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
4	12		22369620
99	45	X	868834460300893509841851440799
99	45	X	868834460300893509841851440799
99	45	X	868834460300893509841851440799
4	12		22369620
4	12		22369620
4	12		22369620
99	45	X	868834460300893509841851440799
99	45	X	868834460300893509841851440799

```
                                *          *******
                                *          DEMOPAK
                                *          *******
                                *
                                *          A PROGRAM FOR DEMONSTRATING VARIOUS FUNCTIONS
                                *                    OF THE IBM 1620
                                *
                                *          **********************
                                *          PRE-DEMO INSTRUCTIONS
                                *          **********************
00402 39 00667 00100            DSTART WATY MESS1
00414 34 00000 00102                   RCTY
00426 48 00000 00000                   H
00438 47 00642 00300                   BNC3 DEINAL
00450 39 00733 00100                   WATY MESS2
00462 34 00000 00102                   RCTY
00474 39 00803 00100                   WATY MESS3
00486 34 00000 00102                   RCTY
00498 39 00859 00100                   WATY MESS4
00510 34 00000 00102                   RCTY
00522 39 00983 00100                   WATY MESS5
00534 34 00000 00102                   RCTY
00546 39 01093 00100                   WATY MESS6
00558 34 00000 00102                   RCTY
00570 39 01235 00100                   WATY CLOWN
00582 34 00000 00102                   RCTY
00594 39 01173 00100                   WATY MESS7
00606 34 00000 00102                   RCTY
00618 34 00000 00102                   RCTY
00630 48 00000 00000                   H
00642 36 00000 00500            DEINAL RNCD 00000
00654 49 00000 00000                   B    00000
00667     00033                MESS1  DAC  33,SW3 ON FOR PRE-DEMO INSTRUCTIONS@
00733     00035                MESS2  DAC  35,MARGINS SHOULD BE SET AT 15 AND 90@
00803     00028                MESS3  DAC  28,SET ONLY ONE TAB STOP AT 50@
00859     00041                MESS4  DAC  41,PROGRAM CALLS FOR 1 DATA CARD, CC 1 TO 60
00941     00021                       DAC  21, %CENTERED AT CC 30¤@
00983     00039                MESS5  DAC  39, CONTAINING NAME OF GROUP FOR WHOM DEMO
01061     00016                       DAC  16, IS BEING GIVEN@
01093     00040                MESS6  DAC  40,SW3 ON WILL GIVE INSTRUCTIONS AS YOU GO@
01173     00031                MESS7  DAC  31,TURN TO CLEAN PAGE, PUSH START@

01235     00045                CLOWN  DAC  45,ERROR MESSAGES WILL PRINT ON TYPEWRITER INPUT
01325     00008                       DAC  8, ERRORS@
00402                                 DEND DSTART
                                *          **********************
                                *          INITIALIZATION PROGRAM
                                *          **********************
00500                                 DORG 500
00500 16 00518  1000            DINIT  TFM  *-18,1000
00512 33 99999 00000                   CF   99999,,,          CLEAR FLAG FROM THOUSANDS
00524 11 00515 000-1                   AM   *-9,1,10
00536 14 00515 000J0                   CM   *-21,10,10
00548 47 00512 01200                   BNE  DINIT-12
00560 25 00405 00400                   TD   405,400
00572 25 00404 00400                   TD   404,400
00584 25 00403 00400                   TD   403,400
00596 25 00402 00400                   TD   402,400
00608 32 00402 00000                   SF   402
00620 36 00000 00500                   RNCD ,,,               PROGRAM LOADER
00632 49 00000 00000                   B
00500                                 DEND DINIT
                                *          **********************
                                *          DEMONSTRATION HEADING
                                *          **********************
00406                                 DORG 406
00406 37 18687 00500                   RACD HEAD,,,           INPUT HEADING LINE 3
00418 34 00000 00102                   RCTY
00430 48 0000  10100                   H    0,10100,6
00442 25 18807 00400            START  TD   HEADRM,400
00454 17 00694 18533                   BTM  ALPHA,DEMO
00466 41 00670 00300                   BNC3 HDEND ,,,         THIS HAS BEEN CHANGED TO NOP
00478 17 00694 17913                   BTM  ALPHA,FUNCT,,,    FOLLOWING LISTS
00490 16 00513  1000                   TFM  TESTE-11,1000,,   INCLUDED FUNCTIONS
00502 44 00508 99999            TESTE  BNE  NEXTE,99999,,     IS FUNCTION INCLUDED
00514 17 17206  0513                   BTM  EDIT,*-1,,,       YES -EDIT AND PRINT
00526 26 18999 00441                   TF   EDNUMB-2,C1,,     ADDR AND TITLE
00538 16 19015 0 000                   TFM  EDNUMB-14,00000,8
00550 26 00585 00513                   TF   *-35,TESTE-11
00562 11 00585 000J2                   AM   *-23,12,10
00574 31 19016 99999                   TR   EDNUMB-15,99999,,  TITLE
00586 17 00694 18999                   BTM  ALPHA,EDNUMB-2,,   PRINT
00598 11 00510 000-1            NEXTE  AM   TESTE-8,1,10,     INCR FOR NEXT FUNCT
```

```
00610 14 00510 00000          CM   TESTF&8,10,10
00622 47 00502 01200          BNE  TESTF,,,          ALL FUNCTIONS LISTED
00634 34 00000 00102          RCTY ,,,               YES
00646 47 00670 00400          BNC3 HDEND
00658 17 00694 17981          BTM  ALPHA,INSTR,,     LISTS OPERATING INSTR
00670 17 17608  0000  HDEND   BTM  NEXT,0
00682 49 00000 00000          B    0,,,              GOES TO FIRST FUNCT
00689                         DORG *-4
                      *       ********************************
                      *       PRINT ALPHA AND SKIP SUBROUTINE
                      *       ********************************
00693     00005               DS   5
00694 26 00888 00693  ALPHA   TF   WATY&6,ALPHA-1,,  INIT TYPE ADDR
00706 26 00881 00693          TF   FINDRM&11,ALPHA-1,,INIT RM SEARCH
00718 26 00753 00693          TF   *&35,ALPHA-1,,    STORE LINE AND SKIP
00730 12 00753 000-2          SM   *&23,2,10,        INFORMATION
00742 26 00844 99999          TF   LINES,99999
00754 32 00843 00000          SF   LINES-1
00766 16 00949 000-0          TFM  LINECM&11,0,10,   INIT LINE PRINTED
00778 43 00870 00842          BD   FINDRM,LINES-2
00790 43 00870 00841          BD   FINDRM,LINES-3
00802 16 00825 000-0  SKIP    TFM  *&23,0,10,        INIT LINES SKIPPED
00814 14 00844 99999  SKIPCM  CM   LINES,99999,10
00826 46 00938 01200          BE   LINECM
00838 34 00000 00102  RCTY    RCTY
00850 11 00825 000-1          AM   SKIPCM&11,1,10,   INCR LINES SKIPPED
00862 49 00814 00000          B    SKIPCM
00870                         DORG *-3
00870 45 00964 99999  FINDRM BNR  UPRM,99999
00882 39 99999 00100  WATY    WATY 99999
00894 11 00949 000-1          AM   LINECM&11,1,10,   INCR LINES PRINTED
00906 26 00888 00881          TF   WATY&6,FINDRM&11,, UP DATE TYPE ADDR
00918 11 00888 000-2          AM   WATY&6,2,10
00930 49 00802 00000          B    SKIP
00938                         DORG *-3
00938 14 00842 99999  LINECM  CM   LINES-2,99999,10, ALL LINES PRINTED
00950 47 00964 01200          BNE  UPRM
00962 42 00000 00000          BB   ,,,               YES
00964                         DORG *-9
00964 11 00881 000-2  UPRM    AM   FINDRM&11,2,10,   INCR RM SEARCH
00976 49 00870 00000          B    FINDRM
00844     00004               LINES DS  4,RCTY&6
```

```
17200                         DORG 17200
17204     00005               DS   5
17206 16 17344 19001  EDIT    TFM  DIGIT&18,EDNUMB,, OUTPUT ADDR
17218 16 17392 17326          TFM  EDEND&6,DIGIT,,   SET HI ORD ZERO SW
17230 16 17459 000-0          TFM  SIGN&11,00,10,,   SET SIGN PLUS
17242 26 17265 17205          TF   *&23,EDIT-1
17254 44 17278 99999          BNF  PLUS,99999
17266 16 17459 000K0          TFM  SIGN&11,20,10,    SET SIGN MINUS
17278 26 17313 17205  PLUS    TF   FLG&11,EDIT-1
17290 12 17313 000-1  DECFLG  SM   FLG&11,1,10
17302 44 17290 99999  FLG     BNF  DECFLG,99999,,    FIND HIGH ORDER FLAG
17314 26 17337 17313          TF   DIGIT&11,*-1
17326 43 17424 99999  DIGIT   BD   NUMB,99999
17338 16 99999 000-0          TFM  99999,0,10,       BLANK HI ORD ZEROS
17350 11 17344 000-2  UPA     AM   *-6,2,10
17362 11 17337 000-1          AM   DIGIT&11,1,10
17374 24 17205 17337          C    EDIT-1,DIGIT&11
17386 46 99999 01300  EDEND   BI   99999,01300,,     HI ORDER ZERO SWITCH
17398 26 17416 17344          TF   *&18,DIGIT&18,,   FINISHED
17410 25 99999 00400          TD   99999,400
17422 42 00000 00000          BB
17424                         DORG *-9
17424 16 17392 17472  NUMB    TFM  EDEND&6,GOON ,,   CHNG HI ORD ZERO SW
17436 26 17454 17344          TF   SIGN&6,DIGIT&18
17448 16 99999 999R9  SIGN    TFM  99999,99999,10
17460 11 17344 000-2          AM   DIGIT&18,2,10
17472 26 17495 17337  GOON    TF   *&23,DIGIT&11
17484 25 17519 99999          TD   MOVE&11,99999,,   MOVE DIGIT
17496 26 17514 17344          TF   *&18,DIGIT&18
17508 16 99999 000P0  MOVE    TFM  99999,70,10,
17520 49 17350 00000          B    UPA
17528                         DORG *-3
17602                         DORG 17602
                      *       ********************************
                      *       BRANCH TO NEXT FUNCTION SUBROUTINE
                      *       ********************************
17606     00005               DS   5
17608 26 17681 17607  NEXT    TF   ISFUN&11,NEXT-1,, INIT FLAG SEARCH
17620 11 17678 000-1          AM   ISFUN&8,1,10
```

```
17632 14 17678 00030            CM   ISFUN&8,10,10,     LAST FUNCTION
17644 47 17670 01200            BNE  ISFUN
17656 31 00000 17708            TR   0,STOP
17668 42 00000 00000            BB
17670                           DORG *-9
17670 44 17620 99999      ISFUN BNF  NEXT&12,99999,,   FUNCTION INCLUDED
17682 26 00006 17681            TF   6,ISFUN&11,,      YES
17694 16 00001 000M9            TFM  1,49,10
17706 42 00000 00000            BB
17708                           DORG *-9
17708 34 00000 00102      STOP  RCTY
17720 39 17753 00100            WATY END
17732 48 00000 00000            H
17744 49 00670 00000            B    HDEND
17751                           DORG *-4
17751     00001                 DC   1,@
17753     00021                 DAC  21,END OF DEMONSTRATION@
                        *       ****************************
                        *            PREPARE I/O AREA SUBROUTINE
                        *       ****************************
17797     00004                 DS   4
17798 26 17881 17797      LD80  TF   LOAD&11,*-1
17810 16 17876 19919            TFM  LOAD&6,BAREA-1
17822 49 17858 00000            B    LOAD-12
17834 26 17881 17833      LD160 TF   LOAD&11,*-1
17846 16 17876 19839            TFM  LOAD&6,AAREA-2
17858 11 17876 000-4            AM   LOAD&6,4,10
17870 16 99999 99999      LOAD  TFM  99999,99999
17882 14 17876 19999            CM   LOAD&6,19999
17894 47 17858 01200            BNE  LOAD-12
17906 42 00000 00000            BB
17908                           DORG *-9
19841     00160           AAREA DAS  160,19841
19920                     BAREA DS   ,19920
00441                     C1    DS   ,START-1
19001                     EDNUMB DS  ,19001
17911     00004                 DC   4,0201
17913     00014           FUNCT DAC  14,DEMONSTRATING@
17941     00018                 DAC  18,ADDRESS  FUNCTION@
17979     00004                 DC   4,0601
17981     00008           INSTR DAC  8,DEPRESS@
17997     00040                 DAC  40,  START TO EXECUTE OR REPEAT A FUNCTION@
```

```
18077     00042                 DAC  42,  RESET, INSERT, RELEASE, START TO EXECUTE
18161     00019                 DAC  19, THE NEXT FUNCTION@
18199     00046                 DAC  46,  RESET AND INSERT 490X000 TO EXECUTE FUNCTION
18291     00018                 DAC  18, AT ADDRESS 0X000@
18327     00049                 DAC  49,PROG SW3 ON - INSTRUCTIONS, SW1, AND SW2 SETTINGS
18425     00013                 DAC  13, ARE PRINTED@
18451     00039                 DAC  39,PROG SW4 - TYPEWRITER ERROR CORRECTION@
18531     00004                 DC   4,0302
18533     00045           DEMO  DAC  45,                1620 FUNCTIONAL DEMONSTRATION@
18623     00032                 DAC  32,                          FOR@
18687     00060           HEAD  DAS  60
18807                     HEADRM DAS
00406                           DEND START-36
                        *       **********************
                        *            CARD INPUT, SECTION 1
                        *       **********************
01000                           DORG 1000
01000 M9 01084 0 102            B    BEG,0102,08
01013     00036           HD    DAC  36,CARD READING-PROGRAM AND DATA INPUT@
01084 17 17608  1000      BEG   BTM  NEXT,1000
01096 17 00694  1178            BTM  ALPHA,RCTY&2,,    RET.CARR. 3 TIMES
01108 17 00694  1013            BTM  ALPHA,HD,,        PRINT HEADING
01120 47 01156 00300            BNC3 INTCM
01132 17 00694 10521            BTM  ALPHA,INSTR,,     PRINT OP INSTR
01144 34 00000 00102            RCTY
01156 16 01799 10727      INTCM TFM  WATIM&11,NOTE,,   INIT. TIME COMM.
01168 48 00000 00000      HALT  H
01180 16 01179 00-00            TFM  CDCNT,0,9,        INIT CARD COUNTER
01192 15 01265 00001            TDM  CHK2&1,1,,        SET 125 CDS SW OFF
01204 46 01764 00200            BC2  TIMCM
01216 37 19841 00500      READCD RACD INPUT
01228 11 01179 000-1            AM   CDCNT,1,10
01240 46 01744 01400            BV   OVER
01252 46 01676 00600            BI   ERROR,600
01264 46 01464 00200      CHK2  BC2  RD250,,,          SW2 ON, READ 125 CDS
01276 47 01216 00100            BNC1 READCD,,,         SW1 ON,STOPS READING
01288 17 17206  1179      PRINT BTM  EDIT,CDCNT
01300 39 19001 00100            WATY EDNUMB
01312 16 01426 10001            TFM  REMARK&6,WORDS,,  INIT FOR COMMENT
01324 46 01632 00200            BC2  TIME
01336 43 01496 01177            BD   MIDDLE,CDCNT-2,,  COUNT 100-999
```

```
01348 11 01426  0000      LITTLE AM   REMARK&6,0,,        COUNT UNDER 100
01360 11 01426  000M0            AM   *-1,40,10
01372 31 10104 10436             TR   ENOUGH-17,YES-1
01384 14 01426 10121      CHK    CM   REMARK&6,ENOUGH
01396 46 01612 01100             BH   LASTCM,,,          TO LAST COMMENT
01408 39 10347 00100             WATY CDSRD
01420 39 99999 00100      REMARK WATY 99999,,,           COMMENT
01432 34 00000 00102             RCTY
01444 34 00000 00102             RCTY
01456 49 01168 00000             B    HALT
01464                            DORG *-3
01464 14 01179 00J25      RD250  CM   CDCNT,125,9,       TEST FOR 250
01476 47 01216 01200             BNE  READCD
01488 49 01288 00000             B    PRINT
01496                            DORG *-3
01496 14 01179 00N00      MIDDLE CM   CDCNT,500,9
01508 46 01564 01100             BH   BIG,,,            BIG IF OVER 500 CDS
01520 31 10104 10424             TR   ENOUGH-17,RIGHT-1
01532 11 01426  0080             AM   REMARK&6,80,,     SELECT COMMENT
01544 11 01543 000M0             AM   *-1,40,10
01556 49 01384 00000             B    CHK
01564                            DORG *-3
01564 11 01426  0160      BIG    AM   REMARK&6,160,,    SELECT COMMENT
01576 12 01575 000M0             SM   *-1,40,10
01588 14 01426 10121      CHKB   CM   REMARK&6,ENOUGH
01600 46 01408 01100             BH   REMARK-12
01612 16 01426 10121      LASTCM TFM  REMARK&6,ENOUGH
01624 49 01408 00000             B    REMARK-12
01632                            DORG *-3
01632 31 10120 10200      TIME   TR   ENOUGH-1,FINAL-1,, CHANGE TIMED COMMENT
01644 11 01426  0280             AM   REMARK&6,280,,     SELECT COMMENT
01656 12 01655 000M0             SM   *-1,40,10
01668 49 01588 00000             B    CHKB
01676                            DORG *-3
01676 17 17206  1179      ERROR  BTM  EDIT,CDCNT
01688 39 10361 00100             WATY BAD,,,
01700 16 01694 10321             TFM  ERROR&18,ERRORS,, CHANGE COMMENT
01712 39 19001 00100             WATY EDNUMB
01724 34 00000 00102             RCTY
01736 49 01264 00000             B    CHK2
01744                            DORG *-3
01744 17 00694 10449      OVER   BTM  ALPHA,POVER,,     1000 CDS READ
```

```
01756 49 01168 00000             B    HALT
01764                            DORG *-3
01764 15 01265 00006      TIMCM  TDM  CHK2&1,6,,         SET 125 CDS SW ON
01776 34 00000 00102             RCTY
01788 17 00694 99999      WATIM  BTM  ALPHA,99999,,      TIME MESSAGE
01800 16 01799 10777             TFM  WATIM&11,WDTIME
01812 48 00000 00000             H
01824 49 01216 00000             B    READCD
01832                            DORG *-3
10000                            DORG 10000
10001     00018          WORDS  DAC  18,LITTLE JOB %NICE@
10037     00002                 DAS  2
10041     00020                 DAC  20,LET ER RUN CHARLIE @
10081     00040          REFA   DAC  40,GOOD READER,      TRY TIMING, SW 2 ON@
10161     00018                 DAC  18,BIG DATA JOB - OK@
10197     00002                 DAS  2
10201     00020          FINAL  DAC  20,ENOUGH - LETS GO ON@
10241     00019                 DAC  19,WHAT - TIMED AGAIN@
10279     00001                 DAS  1
10281     00020          TIMED  DAC  20,30 SEC. ON THE NOSE@
10321     00018          ERRORS DAC  18,MORE ERRORS, CARD@
10357     00002                 DAS  2
10361     00018          BAD    DAC  18,BAD STUFF IN CARD@
10397     00014          CDSRD  DAC  14, CARDS READ, @
10425     00006          RIGHT  DAC  6,RIGHT@
10437     00004          YES    DAC  4,YES@
10121                    ENOUGH DS   ,REFA&40
10447     00004                 DC   4,0102
10449     00034          POVER  DAC  34,STOP, STOP, STOP 1000 CARDS READ @
10519     00004                 DC   4,0401
10521     00027          INSTR  DAC  27,READ CARDS - SW1, SW2, OFF@
10575     00026                 DAC  26,STOP READING -- SW1, ON@
10627     00026                 DAC  26,READ 125 CARDS -- SW2, ON@
10679     00022                 DAC  22,I/O - PROGRAM OR STOP@
10725     00004                 DC   4,0102
10727     00023          NOTE   DAC  23,NOTE THE TIME AT START@
10775     00004                 DC   4,0102
10777     00005          WDTIME DAC  5,TIME@
19841     00080          INPUT  DAS  80,19841
01179     00003          CDCNT  DS   3,HALT&11
01176     00004          RCTY   DC   4,0003,HALT&8
17608                    NEXT   DS   ,17608
```

```
00694                          ALPHA   DS    ,00694
17206                          EDIT    DS    ,17206
19001                          EDNUMB  DS    ,19001
                               DEND
                       *       **********************
                       *       CARD OUTPUT SECTION 2
                       *       **********************
02000                          DORG  2000
02000 MS 02036 0 102           B     BEG,0102,08
02013    00012         HD      DAC   12,CARD OUTPUT@
02036 17 17608  2000   BEG     BTM   NEXT,2000
02048 17 00694 10813           BTM   ALPHA,RCTY &2
02060 17 00694  2013           BTM   ALPHA,HD
02072 47 02108 00300           BNC3  HALT
02084 17 00694 10829           BTM   ALPHA,INSTR,,
02096 34 00000 00102           RCTY
02108 16 02227 0 000   HALT    TFM   COUNT,0,8,          INITIALIZE CARD COUNT
02120 17 17834 0 000           BTM   LD160,0,8,         CLEAR OUTPUT AREA
02132 48 00000 00000           H
02144 47 02180 00100           BNC1  WOW
02156 47 02180 00200           BNC2  WOW
02168 49 02500 00000           B     GOON&12,,,         BOTH SWITCHES ON
02180 39 11039 00400   WOW     WACD  HEAD
02192 23 10807 10823   GO      M     VALUE,WJO,,        GENERATE RANDOM NUMBERS
02204 32 00084  0000   WK1     SF    84,,7
02216 32 00091 0 000   WK2     SF    91,,8
02228 26 10805 00096           TF    VALUE-2,96,,       SET UP NEW MULTIPLIERS
02240 33 10801 00000           CF    VALUE-6
02252 26 10819 00090           TF    WJO-4,90
02264 17 17206  0095           BTM   EDIT,95
02276 31 19910 19000           TR    OUTPUT&70,EDNUMB-1
02288 16 19923 000-0           TFM   OUTPUT&83,,10,     FIRST RANDOM NUMBER
02300 17 17206  0090           BTM   EDIT,90
02312 31 19944 19000           TR    OUTPUT&104,EDNUMB-1
02324 16 19961 000-0           TFM   OUTPUT&121,,10,    SECOND RANDOM NUMBER
02336 16 19953 000-3           TFM   OUTPUT&113,03,10,  WITH A DECIMAL POINT
02348 11 02227 000-1           AM    COUNT,1,10,        CARD COUNT
02360 17 17206  2227           BTM   EDIT,COUNT
02372 31 19870 19000           TR    OUTPUT&30,EDNUMB-1
02384 16 19881 000-0           TFM   OUTPUT&41,,10,
02396 39 19841 00400           WACD  OUTPUT&1
02408 46 02488 00200           BC2   GOON
```

```
02420 47 02192 00100           BNC1  GO
02432 39 19001 00100   DONE    WATY  EDNUMB,,,         TYPE OUT CARDS PUNCHED
02444 34 00000 00101           SPTY
02456 39 11201 00100           WATY  MESS1
02468 34 00000 00102           RCTY
02480 49 02096 00000           B     HALT-12
02488                          DORG  *-3
02488 47 02520 00100   GOON    BNC1  AOK
02500 17 00694 11243           BTM   ALPHA,MESS2,,     BOTH SWITCHES ON
02512 49 02108 00000           B     HALT
02520                          DORG  *-3
02520 14 02227 0 125   AOK     CM    COUNT,125,8,       PUNCH 125 CARDS
02532 47 02192 01200           BNE   GO
02544 49 02432 00000           B     DONE
10800                          DORG  10800
10807    00008         VALUE   DC    8,16895419
02227                  COUNT   DS    ,WK2&11
17206                  EDIT    DS    ,17206
19001                  EDNUMB  DS    ,19001
19840                  OUTPUT  DS    ,19840
10811    00004         RCTY    DC    4,0003
17834                  LD160   DS    ,17834
10823    00012         WJO     DC    12,326331030377
10827    00004                  DC    4,0401
10829    00043         INSTR   DAC   43,PUNCH DATA CARDS CONTAINING RANDOM NUMBERS@
10915    00024                  DAC   24,SW1 ON TO STOP PUNCHING@
10963    00026                  DAC   26,SW2 ON TO PUNCH 125 CARDS@
11015    00012                  DAC   12,I/O TO STOP@
11039    00039         HEAD    DAC   39,          THIS SECTION GENERATES A
11117    00042                  DAC   42, TABLE OF RANDOM NUMBERS          @
11201    00019         MESS1   DAC   19,DATA CARDS PUNCHED@
11241    00004                  DC    4,0102
11243    00038         MESS2   DAC   38,I FIND IT CONFUSING WHEN YOU HAVE BOTH
11319    00029                  DAC   29, SW1 AND SW2 ON, START AGAIN@
17608                  NEXT    DS    ,17608
00694                  ALPHA   DS    ,00694
                               DEND
                       *       ************************
                       *       TYPEWRITER I/O, SECTION 3
                       *       ************************
03000                          DORG  3000
03000 MS 03108 0 102           B     BEG,0102,08
```

```
03013     00048                HD3    DAC  48,VARIABLE LENGTH,ALPHA OR NUMERIC TYPEWRITER I/O@
03108 17 17608  3000           BEG    BTM  NEXT,3000
03120 17 00694 11605                  BTM  ALPHA,RCTY&2
03132 17 00694  3013                  BTM  ALPHA,HD3
03144 16 11977 000-0                  TFM  CTR-1,00,10
03156 34 00000 00102                  RCTY
03168 47 03204 00300                  BNC3 HALT3
03180 17 00694 11609                  BTM  ALPHA,INSTR
03192 34 00000 00102                  RCTY
03204 48 00000 00000           HALT3  H
03216 27 17834 00405                  BT   LD160,RM4
03228 46 03272 00100                  BC1  GO
03240 46 03560 00200                  BC2  ALFA
03252 17 00694 11807                  BTM  ALPHA,ERROR1
03264 49 03204 00000                  B    HALT3
03272                                 DORG *-3
03272 47 03304 00200           GO     BNC2 NUM
03284 17 00694 11683                  BTM  ALPHA,ERROR2
03296 49 03204 00000                  B    HALT3
03304                                 DORG *-3
03304 15 03501 00008           NUM    TDM  OUT&1,8,,      PREPARE TO WRITE NUMERIC
03316 36 19841 00100                  RNTY OUTPUT
03328 47 03360 00400                  BNC4 *&32
03340 34 00000 00102           GOOFER RCTY
03352 49 03216 00000                  B    HALT3&12
03360                                 DORG *-3
03360 45 03392 19869                  BNR  GOOUT,OUTPUT&28
03372 34 00000 00108           TAB    TBTY
03384 49 03416 00000                  B    GOOUT&24
03392                                 DORG *-3
03392 34 00000 00102           GOOUT  RCTY
03404 34 00000 00102                  RCTY
03416 11 11977 000-1                  AM   CTR-1,1,10
03428 33 11976 00000                  CF   CTR-2
03440 39 11973 00100                  WATY A
03452 38 11976 00100                  WNTY CTR-2
03464 34 00000 00101                  SPTY
03476 39 11981 00100                  WATY EQ
03488 34 00000 00101                  SPTY
03500 38 19841 00100           OUT    WNTY OUTPUT
03512 32 11976 00000                  SF   CTR-2
03524 34 00000 00102                  RCTY
```

```
03536 34 00000 00102                  RCTY
03548 49 03204 00000                  B    HALT3
03560 15 03501 00009           ALFA   TDM  OUT&1,9,,      PREPARE TO WRITE ALPHA-NUMERIC
03572 37 19841 00100                  RATY OUTPUT
03584 46 03340 00400                  BC4  GOOFER
03596 45 03392 19897                  BNR  GOOUT,OUTPUT&56
03608 49 03372 00000                  B    TAB
11600                                 DORG 11600
11603     00004               RCTY   DC   4,0003
11607     00004                      DC   4,0301
11609     00028               INSTR  DAC  28,SW1 ON  SW2 OFF FOR NUMERIC@
11665     00034                      DAC  34,SW1 OFF SW2 ON  FOR ALPHA-NUMERIC@
11733     00035                      DAC  35,TYPE IN UP TO 65 DIGITS OR LETTERS@
11805     00004                      DC   4,0102
11807     00036               ERROR1 DAC  36,COME,COME,TURN ON EITHER SW1 OR SW2@
11881     00004                      DC   4,0102
11883     00045               ERROR2 DAC  45,MAKE UP YOUR MIND,TURN OFF EITHER SW1 OR SW2@
11973     00002               A      DAC  2,A@
11978     00003               CTR    DC   3,00@
11981     00002               EQ     DAC  2,#@
00405                         RM4    DS   ,405
19841                         OUTPUT DS   ,19841
17608                         NEXT   DS   ,17608
00694                         ALPHA  DS   ,00694
17834                         LD160  DS   ,17834
                              DEND
                              *      ********************
                              *      ADDITION, SECTION 4
                              *      ********************
04000                                 DORG 4000
04000 MG 04076 0 102                  B    BEG,0102,08
04013     00032               HD     DAC  32,ADDITION OF A COLUMN OF FIGURES@
04076 17 17608  4000          BEG    BTM  NEXT,4000
04088 17 00694 12501                  BTM  ALPHA,RCTY&2
04100 17 00694  4013                  BTM  ALPHA,HD
04112 47 04136 00300                  BNC3 HALT
04124 17 00694 12405                  BTM  ALPHA,INSTR
04136 48 00000 00000           HALT   H
04148 27 17798 00405                  BT   LD80,RM4
04160 36 12501 00100                  WATY NUMB
04172 34 00000 00102                  RCTY
04184 36 19920 00100                  RNTY BAREA
```

```
04196 34 00000 00102              RCTY
04208 46 04148 00400              BC4   *-60
04220 45 04240 19921              BNR   *&20,BAREA&1
04232 49 04548 00000              B     SMALL
04240                             DORG  *-3
04240 16 19919 000-0              TFM   BAREA-1,0,10
04252 45 04772 19928              BNR   CLOWN,BAREA&8
04264 16 04426 12606              TFM   GO&6,COUNT
04276 45 04420 19927              BNR   GO,BAREA&7
04288 11 04426 000-1              AM    GO&6,1,10
04300 45 04420 19926              BNR   GO,BAREA&6
04312 11 04426 000-1              AM    GO&6,1,10
04324 45 04420 19925              BNR   GO,BAREA&5
04336 11 04426 000-1              AM    GO&6,1,10
04348 45 04420 19924              BNR   GO,BAREA&4
04360 11 04426 000-1              AM    GO&6,1,10
04372 45 04420 19923              BNR   GO,BAREA&3
04384 11 04426 000-1              AM    GO&6,1,10
04396 45 04420 19922              BNR   GO,BAREA&2
04408 11 04426 000-1              AM    GO&6,1,10
04420 31 99999 19918        GO    TR    99999,BAREA-2
04432 26 12625 12615              TF    CTWO,CTR
04444 39 12627 00100              WATY  TIMES
04456 34 00000 00102              RCTY
04468 36 19940 00100              RNTY  BAREA&20
04480 34 00000 00102              RCTY
04492 47 04524 00400              BNC4  *&32
04504 26 19942 00405              TF    BAREA&22,RM4
04516 49 04444 00000              B     GO&24
04524                             DORG  *-3
04524 45 04804 19942              BNR   CUTUP,BAREA&22
04536 45 04584 19941              BNR   OK,BAREA&21
04548 39 12759 00100        SMALL WATY  PIKER
04560 34 00000 00102              RCTY
04572 49 04136 00000              B     HALT
04584 32 19940 00000        OK    SF    BAREA&20
04596 14 19941 000-2              CM    BAREA&21,2,10
04608 47 04548 01300              BL    SMALL
04620 21 12615 12625        ADD   A     CTR,CTWO
04632 47 04680 00100              BNC1  GOON
04644 17 17206 12615              BTM   EDIT,CTR
04656 39 19001 00100              WATY  EDNUMB
```

```
04668 34 00000 00102              RCTY
04680 12 19941 000-1        GOON  SM    BAREA&21,1,10
04692 14 19941 000-1              CM    BAREA&21,1,10
04704 47 04620 01200              BNE   ADD
04716 17 17206 12615              BTM   EDIT,CTR
04728 34 00000 00102              RCTY
04740 39 19001 00100              WATY  EDNUMB
04752 34 00000 00102              RCTY
04764 49 04136 00000              B     HALT
04772                             DORG  *-3
04772 39 12557 00100        CLOWN WATY  ERRONE
04784 34 00000 00102              RCTY
04796 49 04136 00000              B     HALT
04804                             DORG  *-3
04804 39 12687 00100        CUTUP WATY  ERRTWO
04816 34 00000 00102              RCTY
04828 49 04136 00000              B     HALT
12400                             DORG  12400
12403     00004                   DC    4,0201
12405     00029             INSTR DAC   29,SW1 ON TO SHOW EACH SUBTOTAL@
12463     00017                   DAC   17,OVERFLOW TO STOP@
12499     00004             RCTY  DC    4,0003
17608                       NEXT  DS    ,17608
00694                       ALPHA DS    ,00694
00405                       RM4   DS    ,00405
17798                       LD80  DS    ,17798
19920                       BAREA DS    ,19920
17206                       EDIT  DS    ,17206
19001                       EDNUMB DS   ,19001
12501     00028             NUMB  DAC   28,ENTER AN EIGHT DIGIT NUMBER@
12557     00025             ERRONE DAC  25,I SAID ONLY EIGHT DIGITS@
12615     00010             CTR   DS    10
12606                       COUNT DS    ,CTR-9
12625     00010             CTWO  DS    10
12627     00030             TIMES DAC   30,ENTER COLUMN SIZE, TWO DIGITS@
12687     00036             ERRTWO DAC  36,I SAID ONLY TWO DIGITS, START AGAIN@
12759     00035             PIKER DAC   35,THIS JOB IS TOO SMALL, START AGAIN@
                            DEND
                      *           **************************
                      *           LOGICAL DECISION, SECTION 5
                      *           **************************
05000                             DORG  5000
```

```
05000 M9 05054 0 102            B     BEG5,0102,08
05013    00021          HD5     DAC   21,DIGITS TO ALPHABETIC@
05054 17 17608  5000    BEG5    BTM   NEXT,5000
05066 17 00694 13913            BTM   ALPHA,RCTY&2
05078 17 00694  5013            BTM   ALPHA,HD5
05090 47 05114 00300            BNC3  HALT5
05102 17 00694 13773            BTM   ALPHA,INSTR
05114 48 00000 00000    HALT5   H
05126 27 17798 00405    CLR     BT    LD80,RM4,,          SET INPUT TO RECORD MARKS
05138 16 05397 19922            TFM   NUMB&11,19922,,     INITIALIZATION
05150 16 05468 19922            TFM   NAVT&6,19922
05162 16 05485 19922            TFM   NAVT&23,19922
05174 15 19920 0000-            TDM   19920,0,11
05186 36 19921 00100            RNTY  19921,,,            READ TWO DIGITS
05198 46 05650 00400            BC4   GOOF,,,             GOOF SWITCT
05210 34 00000 00108            TBTY
05222 2L  5238  5K38            M     *&16,*&16,7 ,       TIME DELAY FOR TABULATING
05234 2L  525   5K50            M     *&16,*&16,7 ,       TIME DELAY FOR TABULATING
05246 2L  5262  5K62            M     *&16,*&16,7 ,       TIME DELAY FOR TABULATING
05258 16 13756 13201            TFM   CTRO,ONE,,          INITIALIZATION
05270 16 13761 13307            TFM   CTRT,TEN
05282 16 13766 13503            TFM   CTRE,TWTY
05294 45 05638 19923            BNR   BAD,19923,,         TEST FOR NUMBER OF DIGITS
05306 45 05418 19922            BNR   GO,19922
05318 45 05350 19921            BNR   GOA,19921
05330 39 13693 00100            WATY  ERTWO ,,,           NOTHING ENTERED
05342 49 05650 00000            B     GOOF
05350                           DORG  *-3
05350 12 05397 000-1    GOA     SM    NUMB&11,01,10
05362 12 05468 000-1            SM    NAVT&6,01,10
05374 12 05485 000-1            SM    NAVT&23,01,10
05386 43 05462 99999    NUMB    BD    NAVT,99999,,        THIS LOOP CHECKS 1 TO 9
05398 39 13315 00100            WATY  ZERO
05410 49 05650 00000            B     GOOF
05418                           DORG  *-3
05418 14 19922 00-13    GO      CM    19922,013,9,        SPECIAL COMMENT WHEN NUMBER IS 13
05430 46 05778 01200            BE    FUN
05442 43 05538 19921            BD    GOC,19921
05454 49 05386 00000            B     NUMB
05462                           DORG  *-3
05462 12 99999 000-1    NAVT    SM    99999,01,10
05474 43 05518 99999            BD    ROUT,99999
05486 26 05504 13756            TF    *&18,CTRO

05498 39 09999 00100            WATY  9999
05510 49 05650 00000            B     GOOF
05518                           DORG  *-3
05518 11 13756 000J2    ROUT    AM    CTRO,12,10
05530 49 05462 00000            B     NAVT
05538                           DORG  *-3
05538 12 19921 000-1    GOC     SM    19921,1,10
05550 43 05670 19921            BD    GOD,19921,,         10 TO 19 RANGE
05562 43 05606 19922    UP      BD    GOE,19922,,         20 OR GREATER
05574 26 05592 13761            TF    *&18,CTRT
05586 39 99999 00100            WATY  99999
05598 49 05650 00000            B     GOOF
05606                           DORG  *-3
05606 12 19922 000-1    GOE     SM    19922,1,10,         THIS LOOP CHECKS 10 TO 19
05618 11 13761 000K0            AM    CTRT,20,10
05630 49 05562 00000            B     UP
05638                           DORG  *-3
05638 39 13627 00100    BAD     WATY  ERONE
05650 17 00694 13213    GOOF    BTM   ALPHA,OLMO&2,,      RCTY TWO TIMES
05662 49 05126 00000            B     CLR
05670                           DORG  *-3
05670 12 19921 000-1    GOD     SM    19921,1,10,         THIS LOOP CHECKS 20 TO 90
05682 43 05738 19921            BD    GOF,19921
05694 26 05712 13766            TF    *&18,CTRE
05706 39 99999 00100            WATY  99999
05718 43 05758 19922            BD    GOG,19922
05730 49 05650 00000            B     GOOF
05738                           DORG  *-3
05738 11 13766 000J6    GOF     AM    CTRE,16,10
05750 49 05670 00000            B     GOD
05758                           DORG  *-3
05758 34 00000 00101    GOG     SPTY  ,,,                 SPACE THE TYPEWRITER AND THEN
05770 49 05462 00000            B     NAVT,,,             GO TO THE 1 TO 9 LOOP
05778                           DORG  *-3
05778 39 13913 00100    FUN     WATY  HAHA
05790 49 05650 00000            B     GOOF
13200                           DORG  13200
13201    00004          ONE     DAC   4,ONE@
13211    00004          OLMO    DC    4,0002
13213    00004                  DAC   4,TWO@
13223    00004                  DS    4,
```

```
13225  00006                    DAC  6,THREE@
13237  00005                    DAC  5,FOUR@
13247  00002                    DS   2,
13249  00005                    DAC  5,FIVE@
13259  00002                    DS   2,
13261  00004                    DAC  4,SIX@
13271  00004                    DS   4,
13273  00006                    DAC  6,SEVEN@
13285  00006                    DAC  6,EIGHT@
13297  00005                    DAC  5,NINE@
13307  00004              TEN   DAC  4,TEN@
13315  00005              ZERO  DAC  5,ZERO@
13325  00002                    DS   2,
13327  00008                    DAC  8,ELEVEN @
13345  00004                    DS   4,
13347  00007                    DAC  7,TWELVE@
13365  00006                    DS   6,
13367  00009                    DAC  9,THIRTEEN@
13385  00002                    DS   2,
13387  00009                    DAC  9,FOURTEEN@
13405  00002                    DS   2,
13407  00008                    DAC  8,FIFTEEN@
13425  00004                    DS   4,
13427  00008                    DAC  8,SIXTEEN@
13445  00004                    DS   4,
13447  00010                    DAC  10,SEVENTEEN@
13467  00009                    DAC  9,EIGHTEEN@
13485  00002                    DS   2,
13487  00008                    DAC  8,NINTEEN@
13503  00007              TWTY  DAC  7,TWENTY@
13517  00002                    DS   2,
13519  00007                    DAC  7,THIRTY@
13533  00002                    DS   2,
13535  00006                    DAC  6,FORTY@
13549  00004                    DS   4,
13551  00006                    DAC  6,FIFTY@
13565  00004                    DS   4,
13567  00006                    DAC  6,SIXTY@
13581  00004                    DS   4,
13583  00008                    DAC  8,SEVENTY@
13599  00007                    DAC  7,EIGHTY@
13613  00002                    DS   2,
```

98

```
13615  00006                    DAC  6,NINTY@
13627  00033              ERONE DAC  33,ENTER ONLY TWO CHARACTERS PLEASE@
13693  00030              ERTWO DAC  30,GO AHEAD AND ENTER TWO DIGITS@
13756  00005              CTRO  DS   5
13761  00005              CTRT  DS   5
13766  00005              CTRE  DS   5
00694                     ALPHA DS   ,00694
17798                     LD80  DS   ,17798
13767  00001                    DS   1,
13771  00004                    DC   4,0301
13773  00022              INSTR DAC  22,SW1, SW2 -- ON OR OFF@
13817  00017                    DAC  17,OVERFLOW TO STOP@
13851  00029                    DAC  29,TYPE IN ANY TWO DIGIT NUMBER@
00405                     RM4   DS   ,00405
13911  00004              RCTY  DC   4,0003
17608                     NEXT  DS   ,17608
13913  00034              HAHA  DAC  34,I DO NOT LIKE THE NUMBER THIRTEEN@
                          DEND
                          *     ************************************
                          *     RAPID SUCCESSIVE ADDITION, SECTION 6
                          *     ************************************
06000                           DORG 6000
06000 M9 06064 0 102            R    REG6,0102,08
06013    00026          HD6     DAC  26,RAPID SUCCESSIVE ADDITION@
06064 17 17608  6000    REG6    RTM  NEXT,6000
06076 17 00694  6981            RTM  ALPHA,RCTY&2
06088 17 00694  6013            RTM  ALPHA,HD6
06100 47 06124 00300            RNC3 HALT6
06112 17 00694  6853            RTM  ALPHA,INSTR
06124 34 00000 00102    HALT6   RCTY
06136 47 06184 00200            RNC2 GOON,,,          TEST FOR THE PRESENT SW2 SETTING
06148 15 06257 00007            TDM  ADD-11,7,,        AND THEN INITIALIZE THE LATTER TESTS
06160 15 06281 00006            TDM  ADD&13,6,,        SO THAT CHANGING SW2 WILL BE
06172 49 06208 00000            R    CLR,,,           ENOUGH TO TEST FOR CHEATING AND
06184 15 06257 00006    GOON    TDM  ADD-11,6,,        STOP THE ADDITION
06196 15 06281 00007            TDM  ADD&13,7
06208 16 06254 0 000    CLR     TFM  SUM,0,8
06220 16 06251 00-00            TFM  SUM-3,0,9
06232 25 06255 00400            TD   SUM&1,400
06244 48 000 0 00003    H       H    ,,5
06256 46 06764 00200            RC2  ERRO,,,          CHEAT MESSAGE
06268 11 06254 000-1    ADD     AM   SUM,1,10,        ADD ONE TO THE COUNTER AN THEN
```

87

```
06280 47 06264 00200          BNC2 ADD,,,          TEST SW2 FOR STOP
06292 33 06249 00000          CF   SUM-5
06304 38 06249 00100          WNTY SUM-5
06316 34 00000 00101          SPTY
06328 34 00000 00101          SPTY
06340 43 06676 06250          BD   BIG,SUM-4,,     TEST FOR NUMBER OF DIGITS
06352 43 06552 06252          BD   MID,SUM-2,,     THEN SELECT A COMMENT
06364 43 06464 06253          BD   SMALL,SUM-1
06376 39 14001 00100    HERE  WATY COMA,,2
06388 34 00000 00102          RCTY
06400 11 06382  0044          AM   HERE&6,44,,     CHANGE COMMENT
06412 14 06382 14133          CM   HERE&6,COMA&132
06424 46 06444 01200          BE   OUT,,,          REINITIALIZE COMMENT
06436 49 06124 00000          B    HALT6
06444                         DORG *-3
06444 12 06382 00J32    OUT   SM   HERE&6,132,9
06456 49 06124 00000          B    HALT6
06464                         DORG *-3
06464 39 14125 00100    SMALL WATY COMB,,2
06476 34 00000 00102          RCTY
06488 11 06470 000M2          AM   SMALL&6,42,10,    CHANGE COMMENT
06500 14 06470 14293          CM   SMALL&6,COMB&168
06512 46 06532 01200          BE   OUT2,,,         REINITIALIZE COMMENT
06524 49 06124 00000          B    HALT6
06532                         DORG *-3
06532 12 06470 00J68    OUT2  SM   SMALL&6,168,9
06544 49 06124 00000          B    HALT6
06552                         DORG *-3
06552 25 06575 06251    MID   TD   CM&11,SUM-3
06564 14 06606 000-0    CM    CM   RC&6,0,10
06576 46 06676 01100          BH   BIG
06588 39 14289 00100          WATY COMC,,2
06600 34 000 4 00102    RC    RCTY 4,,5
06612 11 06594 00004          AM   RC-6,64,10,     CHANGE COMMENT
06624 14 06594 14481          CM   RC-6,COMC&192
06636 46 06656 01200          BE   OUT3,,,         REINITIALIZE COMMENT
06648 49 06124 00000          B    HALT6
06656                         DORG *-3
06656 12 06594 00J92    OUT3  SM   RC-6,192,9
06668 49 06124 00000          B    HALT6
06676                         DORG *-3
06676 39 14483 00100    BIG   WATY COMD,,2
```

```
06688 34 00000 00102          RCTY
06700 11 06682 000M2          AM   BIG&6,42,10,    CHANGE COMMENT
06712 14 06682 14651          CM   BIG&6,COMD&168
06724 46 06744 01200          BE   OUT4,,,         REINITIALIZE COMMENT
06736 49 06124 00000          B    HALT6
06744                         DORG *-3
06744 12 06682 00J68    OUT4  SM   BIG&6,168,9
06756 49 06124 00000          B    HALT6
06764                         DORG *-3
06764 39 14643 00100    ERRO  WATY CMT,,2
06776 34 00000 00102          RCTY
06788 11 06770 000K6          AM   *-18,26,10,     CHANGE COMMENT
06800 14 06770 14799          CM   *-30,END&26
06812 47 06124 01200          BNE  HALT6
06824 16 06770 14643          TFM  ERRO&6,CMT ,,   REINITIALIZE COMMENT
06836 49 06124 00000          B    HALT6
06851     00004               DC   4,0201
06853     00045     INSTR     DAC  45,PUSH START THEN CHANGE SW 2 TO STOP ADDITION@
06943     00017               DAC  17,OVERFLOW TO STOP@
06979     00004     RCTY      DC   4,0003
17608               NEXT      DS   ,17608
00694               ALPHA     DS   ,00694
14000                         DORG 14000
14001     00022     COMA      DAC  22,THAT WAS FAST ON SW 2@
14045     00022               DAC  22,YOU ARE NOW AN EXPERT@
14089     00018               DAC  18,GOOD SHOW OLD BOY@
14125     00021     COMB      DAC  21,YOU CAN IMPROVE THIS@
14167     00021               DAC  21,TRY IT ONE MORE TIME@
14209     00021               DAC  21,FASTER ON THE SWITCH@
14251     00019               DAC  19,TWO DIGITS IS FAST@
14289     00032     COMC      DAC  32,FASTER ON SWITCH2 TO REDUCE SUM@
14353     00032               DAC  32,THE 1620 IS FASTER THAN YOU ARE@
14417     00033               DAC  33,I KNOW YOU CAN DO BETTER CHARLIE@
14483     00021     COMD      DAC  21,FASTER FASTER FASTER@
14525     00021               DAC  21,YOU ARE TOO SLOW    @
14567     00020               DAC  20,SPEED IT UP CHARLIE@
14607     00001               DAC  1,A
14609     00017               DAC  17,GET THE LEAD OUT@
06254               SUM       DS   ,H&10
14643     00026     CMT       DAC  26, YOU  MOVED  SW2 TOO SOON@
14695     00010               DAC  10,PLAY FAIR@
14719     00006               DS   6
```

```
14721     00011                    DAC   11,WATCH THAT@
14745     00004                    DS    4,
14747     00013                    DAC   13,DIRTY PLAYER@
14773     00011          END       DAC   11,DON'T CHEAT@
                                   DEND
                         *         ***********************************
                         *         SUCCESSIVE MULTIPLICATION, SECTION 7
                         *         ***********************************
07000                              DORG  7000
07000 M9 07064 0 102               B     BEG7,0102,08
07013     00026          HD7       DAC   26,SUCCESSIVE MULTIPLICATION@
07064 17 17608  7000     BEG7      BTM   NEXT,7000
07076 17 00694 15549               BTM   ALPHA,RCTY&2
07088 17 00694  7013               BTM   ALPHA,HD7
07100 47 07124 00300               BNC3  HALT7
07112 17 00694 15021               BTM   ALPHA,INSTR
07124 48  0000 00000     HALT7     H     ,,2
07136 17 00694 15121     ENTER1    BTM   ALPHA,VAR1
07148 27 17834 00405               BT    LD160,RM4
07160 36 19847 00100               RNTY  NR-7
07172 17 00694 15549               BTM   ALPHA,RCTY&2
07184 46 07136 00400               BC4   ENTER1
07196 45 07228 19847               BNR   LOOP1,NR-7
07208 17 00694 15159               BTM   ALPHA,ERMES1,,      REQUEST AN ENTRY
07220 49 07136 00000               B     ENTER1
07228                              DORG  *-3
07228 45 14928 19854     LOOP1     BNR   ERROR1,NR,,         CHECK FOR TOO LONG A NUMBER
07240 43 07272 19847               BD    ENTER2,NR-7
07252 17 00694 15181               BTM   ALPHA,ERMES2
07264 49 07136 00000               B     ENTER1
07272                              DORG  *-3
07272 17 00694 15141     ENTER2    BTM   ALPHA,VAR2
07284 27 17798 00405               BT    LD80,RM4
07296 36 19920 00100               RNTY  PR-1
07308 17 00694 15549               BTM   ALPHA,RCTY&2
07320 46 07272 00400               BC4   ENTER2
07332 45 07364 19920               BNR   LOOP2,PR-1
07344 17 00694 15159               BTM   ALPHA,ERMES1,,      REQUEST AN ENTRY
07356 49 07272 00000               B     ENTER2
07364                              DORG  *-3
07364 45 07944 19922     LOOP2     BNR   ERROR2,PR&1,,       CHECK FOR TOO GREAT A POWER
07376 45 07412 19921               BNR   COMP,PR

07388 25 19921 19920               TD    19921,19920,,       CONVERSION OF A SINGLE DIGIT POWER X
07400 15 19920 0000-               TDM   19920,0,11,         TO A DIGIT POWER OX
07412 32 19847 00000     COMP      SF    NR-7
07424 32 19920 00000               SF    PR-1
07436 14 19921 000-1               CM    PR,01,10,           TEST FOR FIRST POWER
07448 46 14960 01200               BE    SELF
07460 14 19921 000-0               CM    PR,0,10
07472 46 07976 01200               BE    MESONE
07484 16 15581 000-0               TFM   CTR,00,10,          ZERO PROD. CTR.
07496 16 07135 000-0               TFM   LENGTH,0,10
07508 26 15559 07132               TF    FLNR,ZEROES-1
07520 45 07540 19847     SEARCH    BNR   LOOP3,NR-7,7,        FIND LENGTH OF NUMBER FIELD
07532 49 07572 00000               B     SET8
07540                              DORG  *-3
07540 11 07135 000-1     LOOP3     AM    LENGTH,1,10
07552 11 07531 000-1               AM    SEARCH&11,1,10
07564 49 07520 00000               B     SEARCH
07572                              DORG  *-3
07572 16 07531 19847     SET8      TFM   SEARCH&11,NR-7
07584 16 15584 00-00               TFM   EXPON,0,9
07596 21 15584 07135               A     EXPON,LENGTH
07608 45 07688 19848     ZOOM      BNR   FIND,NR-6,7
07620 26 07638 07619               TF    *&18,ZOOM&11
07632 15 99999 00000     GOG       TDM   99999,0,2
07644 14 07638 19854               CM    GOG&6,NR
07656 46 07708 01200               BE    REGEN1
07668 11 07638 000-1               AM    GOG&6,1,10
07680 49 07632 00000               B     GOG
07688                              DORG  *-3
07688 11 07619 000-1     FIND      AM    ZOOM&11,1,10
07700 49 07608 00000               B     ZOOM
07708                              DORG  *-3
07708 26 15559 19853     REGEN1    TF    FLNR,NR-1,7
07720 16 07619 19848               TFM   ZOOM&11,NR-6
07732 11 15581 000-2               AM    CTR,2,10
07744 26 15578 07133               TF    PROD,ZEROES,,       CLEAR PRODUCT AREA
07756 26 15571 07132               TF    PROD-7,ZEROES-1
07768 21 15571 15559               A     PROD-7,FLNR
07780 15 00079 00000     WOW       TDM   79,0
07792 23 15578 15559               M     PROD,FLNR
07804 26 15578 00092               TF    PROD,92
07816 21 15584 07135               A     EXPON,LENGTH
```

```
07828 43 07900 15565          BD   GOGO,PROD-13,,     IS THERE A HIGH ORDER ZERO
07840 12 15584 000-1          SM   EXPON,1,10,       IF YES, THEN DECREMENT THE EXPONENT
07852 26 19950 15578          TF   XPRODX,PROD
07864 26 15577 19900          TF   PROD-1,XPRODX
07876 15 15576 00000          TDM  PROD,0,,          THEN ADD A LOW ORDER ZERO
07888 32 15585 00000          SF   PROD-13
07900 24 19921 15581 GOGO     C    PR,CTR
07912 46 14800 01200          BE   PRINT
07924 11 15581 000-1          AM   CTR,1,10
07936 49 07780 00000          B    WOW
07944                         DORG *-3
07944 17 00694 15421 ERROR2   BTM  ALPHA,MESS2
07956 25 15491 15490          TD   MONTG-2,MONTG-3,,  CHANGE COMMENT SECOND TIME
07968 49 07272 00000          B    ENTER2
07976                         DORG *-3
07976 17 00694 15513 MESONE   BTM  ALPHA,MESS3
07988 49 07136 00000          B    ENTER1
14800                         DORG 14800
14800 17 00694 15237 PRINT    BTM  ALPHA,ANS
14812 27 17798 00405          BT   LD80,RM4
14824 31 19940 15565          TR   19940,PROD-13
14836 33 19940 00000          CF   19940
14848 38 19940 00100          WNTY 19940
14860 17 00694 15261          BTM  ALPHA,ANSEXP
14872 31 19970 15582          TR   19970,EXPON-2
14884 33 19970 00000          CF   19970
14896 38 19970 00100          WNTY 19970
14908 17 00694 15549          BTM  ALPHA,RCTY&2
14920 49 07136 00000          B    ENTER1
14928                         DORG *-3
14928 17 00694 15317 ERROR1   BTM  ALPHA,MESS1
14940 25 15363 15362          TD   OLMO-2,OLMO-3,,    CHANGE COMMENT SECOND TIME
14952 49 07136 00000          B    ENTER1
14960                         DORG *-3
14960 33 19847 00000 SELF     CF   NR-7
14972 17 00694 15525          BTM  ALPHA,MESS4
14984 38 19847 00100          WNTY NR-7
14996 17 00694 15549          BTM  ALPHA,RCTY&2
15008 49 07136 00000          B    ENTER1
15016                         DORG *-3
15019 00004                   DC   4,0201
15021 00020 INSTR             DAC  20,TYPE UP TO 7 DIGITS@

15061 00026                   DAC  26,THEN TYPE A 2 DIGIT POWER@
15113 00002                   DAC  2,@
15119 00004                   DC   4,0100
15121 00008 VAR1              DAC  8,NUMBER @
15139 00004                   DC   4,0100
15141 00007 VAR2              DAC  7,POWER @
15157 00004                   DC   4,0102
15159 00009 ERMES1            DAC  9,GO AHEAD@
15179 00004                   DC   4,0102
15181 00026 ERMES2            DAC  26,NO HIGH ORDER ZERO PLEASE@
15235 00004                   DC   4,0100
15237 00010 ANS               DAC  10,ANSWER 0.@
15259 00004                   DC   4,0100
15261 00026 ANSEXP            DAC  26,  TIMES TEN TO THE POWER @.
15315 00004                   DC   4,0102
15317 00024 MESS1             DAC  24,ENTER ONLY SEVEN DIGITS@
15365 00026 OLMO              DAC  26,HOW OFTEN MUST I TELL YOU@
15419 00004                   DC   4,0102
15421 00036 MESS2             DAC  36,THE POWER MAY NOT EXCEED TWO DIGITS@
15493 00008 MONTG             DAC  8,CHARLIE@
15511 00004                   DC   4,0102
15513 00004 MESS3             DAC  4,ONE@
15523 00004                   DC   4,0100
15525 00010 MESS4             DAC  1 ,ANSWER # @
15547 00004 RCTY              DC   4,0003
15549 00002                   DS   2
15551 00002 L2                DC   2,0
19854       NR                DS   ,19854
19921       PR                DS   ,19921
07133       ZEROES            DS   ,HALT7&9
15559 00008 FLNR              DS   8,
07135       LENGTH            DS   ,HALT7&11
15564 00005                   DS   5
15578 00014 PROD              DC   14,0
15579 00001                   DSC  1,@
17608       NEXT              DS   ,17608
00694       ALPHA             DS   ,00694
17798       LD80              DS   ,17798
00405       RM4               DS   ,405
17834       LD160             DS   ,17834
15581 00002 CTR               DC   2,0
```

```
19900 00014                XPRODX DC   14,0,19900
15584 00003                FXPON  DC   3,000
15585 00001                       DSC  1,@
                           *      DEND
                           *
                           *      ********************************
                           *      COMBINATION OF EFFECTS, SECTION 8
                           *      ********************************
08000                              DORG 8000
08000 M9 08058 0 103               B    BEG8,0103,08
08013    00023             HD8     DAC  23,COMBINATION OF EFFECTS@
08058 17 17608  8000       BEG8    BTM  NEXT,8000
08070 17 00694  8347               BTM  ALPHA,RCTY&2,,          RETURN CARR. 3 TIMES
08082 17 00694  8013               BTM  ALPHA,HD8,,             PRINT HEADING
08094 47 08142 00300               BNC3 HALT8
08106 17 00694 15865               BTM  ALPHA,INSTR,,           PRINT OP.INSTR9
08118 17 17834 0 000               BTM  LD160,,8,          CLEAR OUTPUT AREA
08130 46 08142 00900               BLC  HALT8
08142 48 00000 00000       HALT8   H
08154 17 17834 0 000               BTM  LD160,,8
08166 26 08998 16307               TF   MULTCT,ZEROES-25,,      INITIALIZE MULTIPLICATION AND
08178 26 08148 16302               TF   CDCNT,ZEROES-30,,       CARD COUNT COUNTERS
08190 26 08153 16299               TF   CNT1,ZEROES-33
08202 31 19840 16238               TR   AAREA,NBR-1,,      PUNCH HEADING CARD
08214 16 19853 000-0               TFM  AAREA&13,,10
08226 31 19854 16252               TR   AAREA&14,PR-1
08238 16 19865 000-0               TFM  AAREA&25,,10
08250 31 19866 16264               TR   AAREA&26,TRUNK-1
08262 16 19885 000-0               TFM  AAREA&45,,10
08274 31 19950 16284               TR   AAREA&110,ANS-1
08286 16 19963 000-0               TFM  AAREA&123,,10
08298 39 19841 00400               WACD AAREA&1
08310 36 19840 00500       READ    RNCD AAREA,,,           READ IN DATA CARDS
08322 32 19840 00000       ONE     SF   AAREA
08334 32 19843 00000       TWO     SF   AAREA&3
08346 26 08331 19841               TF   POWER,AAREA&1
08358 26 08333 19844               TF   NUMBER,AAREA&4
08370 11 08148 000-1       CONT    AM   CDCNT,1,10
08382 26 16362 16327               TF   SUM,ZEROES-5
08394 26 16392 16327               TF   INTER,ZEROES-5
08406 21 16362 08333               A    SUM,NUMBER
08418 11 08153 000-1               AM   CNT1,1,10

08430 26 16392 08333               TF   INTER,NUMBER
08442 24 08153 08331       TEST    C    CNT1,POWER,,                DEVELOP SUM
08454 46 08582 01200               BE   PUNCH
08466 14 08153 000J5               CM   CNT1,15,10
08478 46 08570 01200               BE   ERROR2
08490 26 00079 16307               TF   79,ZEROES-25
08502 23 16392 08333               M    INTER,NUMBER
08514 26 16392 00099               TF   INTER,99
08526 11 08998 000-1               AM   MULTCT,1,10
08538 21 16362 16392               A    SUM,INTER
08550 11 08153 000-1       GO      AM   CNT1,1,10
08562 49 08442 00000               B    TEST
08570                              DORG *-3
08570 32 08151 00000       ERROR2  SF   HALT8&9,,,        OVERFLOW CAUSES TRUNCATION
08582 17 17834 0 000       PUNCH   BTM  LD160,,8,         CLEAR PUNCH AREA
08594 16 08153 000-0               TFM  CNT1,,10,          REINITIALIZATION
08606 17 17206  8333               BTM  EDIT,NUMBER,,      SET UP PUNCH AREA TO
08618 31 19842 19000               TR   AAREA&2,EDNUMB-1,,  PUNCH OUT RESULTS
08630 16 19849 000-0               TFM  AAREA&9,,10
08642 17 17206  8331               BTM  EDIT,POWER
08654 31 19854 19000               TR   AAREA&14,EDNUMB-1
08666 16 19861 000-0               TFM  AAREA&21,,10
08678 17 17206 16362               BTM  EDIT,SUM
08690 31 19926 19000               TR   AAREA&86,EDNUMB-1
08702 16 19927 000-0               TFM  AAREA&87,,10
08714 16 19989 000-0               TFM  AAREA&149,,10
08726 44 08762 08151               BNF  GOGO,HALT8&9,,     TEST FOR TRUNCATION
08738 33 08151 000-0       OLMO    CF   HALT8&9,,10
08750 16 19873 00007               TFM  AAREA&33,67,10,    X IF TRUNCATED
08762 39 19841 00400       GOGO    WACD AAREA&1
08774 47 08810 00200               BNC2 LAST
08786 14 08148  0125               CM   CDCNT,125,7
08798 46 08842 01200               BE   PRINT
08810 46 08842 00900       LAST    BLC  PRINT
08822 46 08842 00100               BC1  PRINT
08834 49 08310 00000               B    READ
08842                              DORG *-3
08842 17 00694 16043       PRINT   BTM  ALPHA,MESS1,,           POSITION CARD COUNT
08854 27 17798 00405               BT   LD80,RM4
08866 43 08958  8144       GROY    BD   WRITE1,CDCNT-4,7
08878 34 00000 00101               SPTY
08890 11 08748 000-1               AM   CNT2,1,10
```

```
08902 11 08877 000-1          AM   OBOY&11,1,10
08914 14 08749 000-4          CM   CNT2,4,10
08926 47 08866 01200          BNE  OBOY
08938 25 19920 08148          TD   19920,CDCNT
08950 49 15600 00000          B    DAVE
08958                         DORG *-3
08958 26 08981 08877 WRITE1   TF   *&23,OBOY&11
08970 31 19920 99999          TR   19920,99999
08982 49 15600 00000          B    15600
08989                         DORG *-4
08998     00010     MULTCT    DC   10,0
08999     00001               DSC  1,@
15600                         DORG 15600
15600 16 08877  8144 DAVE     TFM  OBOY&11,CDCNT-4
15612 26 08749 16299 WRITE2   TF   CNT2,ZEROES-33
15624 38 19920 00100          WNTY 19920
15636 17 00694  8583          BTM  ALPHA,SPACE&2
15648 17 00694 16089          BTM  ALPHA,MESS2
15660 27 17798 00405          BT   LD80,RM4
15672 43 15768  8989 WOW      BD   WRITE3,MULTCT-9,7,          POSITION MULTIPLY COUNT
15684 11 08749 000-1          AM   CNT2,1,10
15696 11 15683 000-1          AM   WOW&11,1,10
15708 34 00000 00101          SPTY
15720 14 08749 000-9          CM   CNT2,9,10
15732 47 15672 01200          BNE  WOW
15744 25 19920 08998          TD   19920,MULTCT
15756 49 15792 00000          B    WRITE4-12
15768 26 15791 15683 WRITE3   TF   *&23,WOW&11
15780 31 19920 99999          TR   19920,99999
15792 16 15683  8989          TFM  WOW&11,MULTCT-9
15804 26 08749 16299 WRITE4   TF   CNT2,ZEROES-33
15816 38 19920 00100          WNTY 19920
15828 17 00694  8583          BTM  ALPHA,SPACE&2
15840 17 00694 16165          BTM  ALPHA,MESS3
15852 49 08142 00000          B    HALT8
15860                         DORG *-3
15863     00004               DC   4,0701
15865     00023     INSTR     DAC  23,SW2 ON READS 125 CARDS@
15911     00020               DAC  20,SW1 ON STOPS READER@
15951     00014               DAC  14,OFLOW TO STOP@
15979     00022               DAC  22,LOAD DATA, PUSH START@
16023     00002               DAC  2, @
```

96

102

```
16027     00002               DAC  2, @
16031     00002               DAC  2, @
16035     00001               DAC  1,@
16037     00001               DAC  1,@
16041     00004               DC   4,0100
16043     00021     MESS1     DAC  21,NUMBER OF CARDS READ@
16087     00004               DC   4,0100
16089     00036     MESS2     DAC  36,NUMBER OF MULTIPLICATIONS PERFORMED@
16163     00004               DC   4,0104
16165     00037     MESS3     DAC  37,END OF JOB, LOAD DATA AND PUSH START@
16239     00007     NBR       DAC  7,NUMBER@
16253     00006     PR        DAC  6,POWER@
16265     00010     TRUNK     DAC  10,TRUNCATED@
16285     00007     ANS       DAC  7,ANSWER@
08148               CDCNT     DS   ,HALT8&6
08149     00001               DC   1,@,HALT8&7
16332     00035     ZEROES    DC   35,0
16362     00030     SUM       DC   30,0
16392     00030     INTER     DC   30,0
08333               NUMBER    DS   ,ONE&11
08331               POWER     DS   ,ONE&9
08749               CNT2      DS   ,OLMO&11
08345     00004     RCTY      DC   4,0003,TWO&11
17608               NEXT      DS   ,17608
08581     00004     SPACE     DC   4,0002,ERROR2&11
00694               ALPHA     DS   ,00694
17798               LD80      DS   ,17798
00405               RM4       DS   ,405
08153               CNT1      DS   ,HALT8&11
17834               LD160     DS   ,17834
17206               EDIT      DS   ,17206
19840               AAREA     DS   ,19840
19001               FDNUMB    DS   ,19001
                              DEND
```

97

55

SAMPLE INPUT SECTION EIGHT

```
12 04
 5 02
 6 02
10 10
10 11
10 12
10 13
10 14
10 15
10 16
99 99
99 15
66 03
74 74
85 85
96 96
10 03
25 24
26 26
27 28
78 45
17 12
18 18
45 78
 2 02
89 56
18 18
12 13
15 99
45 99
10 06
20 09
 1 15
 1 20
 2 04
12 04
97 64
45 12
 3 06
```

```
45 67
14 14
15 12
76 54
78 96
16 12
17 15
98 76
17 14
17 13
 5 02
 6 02
99 99
66 03
96 96
10 10
10 11
15 16
16 15
10 17
10 18
17 18
18 17
19 18
10 19
10 11
 3 04
20 20
20 21
21 20
28 27
30 30
32 23
56 23
12 45
23 32
11 12
23 56
36 25
11 13
```

14	74
12	12
13	13
12	34
43	21
78	54
23	23
24	25
10	12
10	03
25	24
10	13
10	14
10	15
26	26
27	28
2	02
10	16
99	15
17	12
18	18
89	56
18	18
12	13
1	15
15	99
45	99
45	12
15	16
16	15
10	06
20	09
1	20
10	17
10	18
17	18
2	04
12	04
18	17
97	64

3	06
3	04
19	18
10	19
10	11
11	12
20	20
11	13
12	12
20	21
21	20
28	27
32	23
56	23
23	56
36	25
43	21
78	54
23	23
24	25
45	67
76	54
78	96
13	13
14	14
15	12
16	12
17	15
17	14
17	13
30	30
23	32
12	34
74	74
45	78
14	74
98	76
85	85
78	45
12	45