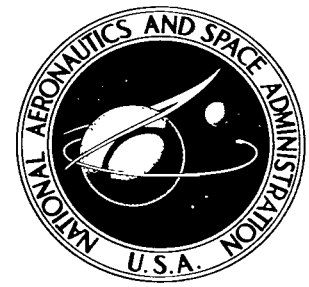
NASA TECHNICAL NOTE

NASA TN D-2850

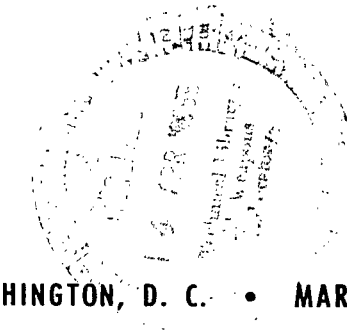


A DATA LOADING ROUTINE FOR THE IBM 7094 AND 7094 - 704X SYSTEMS

by Lawrence F. Hatakeyama

Goddard Space Flight Center

Greenbelt, Md.



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1966

NASA TN D-2850

A DATA LOADING ROUTINE FOR THE IBM 7094 AND 7094 - 704X SYSTEMS

By Lawrence F. Hatakeyama

Goddard Space Flight Center
Greenbelt, Md.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $0.75

CONTENTS

A DATA LOADING ROUTINE FOR THE IBM 7094 AND 7094-704X SYSTEMS

by
Lawrence F. Hatakeyama
Goddard Space Flight Center

INTRODUCTION

The routine discussed in this report resulted from a search for a versatile data loader to be used with FORTRAN IV and MAP coded programs to be excuted on an IBM 7094 digital computer. The routines available at the time this search was launched were programmed for the older IBM systems. It was hoped that this search would yield a routine that could be readily converted and extended for the projected use. After a short search, it was realized that the time and effort to be expended searching for, analyzing, and modifying a suitable routine would probably be more than that required to program a new one.

A characteristic designed into the projected routine was that it is easy to use. It is given control through a simple calling statement and does not require preconceived formats and lists or their equivalents for the conversion and loading of data. The programmer utilizing this routine is freed of the need to consider in detail the nature of the input data required by his program. The object time user is given the option of loading the required data in the format that is most convenient to him.

The calling sequence provides the routine with the initial loading point to a data storage area. All variables and arrays to be loaded during a given call on the routine would be placed in such an area by the programmer. The routine tends to load consecutive data words into successively higher storage locations beginning with the location at the initial loading point. The object time user has the means to shift the current loading point of the routine to any location at any time to achieve the pattern of loading desired. He may therefore load each location or any arbitrary combination of locations within the referenced area reserved for the data. This loading is achieved despite a lack of knowledge concerning the absolute locations of the initial loading point, variables, and arrays. It does require the familiarization of the user with the displacement of each variable and array from the initial loading point.

The decimal data items recognized by the routine include integers, single-precision floating point (or real) numbers, single-precision fixed point numbers, and double-precision floating point numbers. No restrictions are placed on the punching of any of these types of data within any given decimal data field. The number of items punched into any of these data fields is left to the object

time user. This data field originates at column 2 on each decimal data card and may extend across the card to column 80. Blanks are not allowed within the data field. The subfields of all except the last data item contained within a given data field are terminated by commas; the last subfield is terminated by a blank or the end of the card. These subfields are not fixed in length; they may be as long as necessary to formulate a given item.

Octal data recognized by the routine consist solely of octal integers. These are arranged in the octal data cards in the manner described above for the decimal data.

Hollerith data recognized by the routine are punched into a data field beginning at column 7 within each Hollerith data card. Each of these cards also contains a count in column 2 which specifies the number of words contained within the data field of the card. This count may specify up to 10 words. If more than 10 words are to be loaded, two or more of these cards must be used.

The routine is directed to return control to the calling program on processing a card devised for this purpose or by encounter with an end-of-file. In the latter case, the status of an indicator interrogated by the calling program is altered before the return of control is executed. This alteration is suspended if the error procedure of the routine has been executed.

CALLING THE ROUTINE

Passage of control to the loading routine is directed through an IBJOB subroutine calling sequence. This sequence is generated by the IBJOB processor from the following FORTRAN IV statement or its MAP equivalent:

CALL LOAD (N, A)

The first argument in the preceding statement, N, is a reference to an indicator. This indicator is always zeroed by the routine before it begins to process card images. The status of this indicator may remain unchanged throughout the subsequent processing of card images and return of control to the calling program. A plus one or a minus one, respectively, is loaded at the indicator location if a processing error or an end-of-file (hereafter referred to as an *eof*) is detected by the routine. The loading of the *eof* indication is suppressed if the *eof* is encountered after the error indication has been posted.

The second argument, A, is a reference for data storage. It may be the location of a variable, a location within an array or a common block, an entry point, or an absolute location. All except the last may be referenced through the FORTRAN calling statement. The absolute location must be referenced through the equivalent MAP instructions. Whatever the case may be, the referenced location will be the first location to be loaded with a data word if the routine is not directed to begin loading at another location. It is the base address used in the computation of effective addresses of locations loaded by the routine.

EXECUTION OF THE ROUTINE

On assuming control, the routine checks the calling sequence for the required arguments. Failure to provide the two arguments discussed in the preceding section will result in a diagnostic and premature job termination.

The routine then proceeds to the processing of card images after zeroing the indicator discussed above and an index-register quantity used in the computation of effective storage addresses. Successive card images are processed until a return of control to the calling program is directed. The number of card images processed may vary widely. It is governed by the needs of the calling program and by the manner in which the object time user chooses to satisfy these needs. Hollerith, octal, and decimal data may be loaded as required. There is no loader restriction on the quantity of any particular type of data. There is also no requirement that some fixed quantity of a particular combination of data be loaded. There is no preconceived list or format to be satisfied.

Eight types of card images are recognized by the routine. Each type is identified by its column one character. The identifying character may be C, D, E, H, P, R, S, or blank.

The C- and the blank cards contain no data to be loaded. These cards also have no significant effect on the execution of the routine. They may serve as spacer cards locatable anywhere within the deck and may contain comments punched into columns 2 through 80.

The D-cards contain the decimal data loaded by the routine. Each of these cards has a variable length data field which begins at column 2. This field may extend across the card to column 80 or be terminated at some intermediate column by a blank. Comments may be punched into the field following the blank terminating the data field. Each data item completely occupies a subfield within the data field. Each of these subfields is separated from an adjacent subfield by a comma. A data subfield may have zero length, in which case a zero word is specified, or it may be as long as required to formulate a data item. Four types of decimal data items are recognized by the routine and representations of all four types may be placed within a given data field. The formulation of these data items is described in a following section.

The E-cards contain the octal data loaded by the routine. The arrangement of the data in these cards is the same as described above for the D-cards. Only one type of octal data is recognized by the routine. Its formulation is discussed in a following section.

The H-cards contain the Hollerith data loaded by the routine. Each of these cards also contains a decimal digit punched into column 2. This digit is a count of the Hollerith words contained within the card. The maximum count of ten is indicated by a zero. If more than ten Hollerith words are to be loaded, two or more H-cards must be used. These words are punched into the H-card data field beginning at column 7. Each word consists of six characters including blanks. Since each word has the same constant length, commas are neither needed nor utilized to define the field occupied by each word. The field of each Hollerith word is therefore contiguous with the fields of neighboring words.

The P- and the S-cards are used to alter the placement of data words in storage. This is done by changing the value of an index register quantity by the amount specified by an unsigned left-adjusted decimal integer punched between columns 2 and 5 inclusive in each of these cards. The index register quantity is used in the computation of effective addresses of storage locations loaded by the routine. The base address in these address computations is the storage reference provided as the second argument in the call on the routine. The index register quantity is zeroed each time control is passed to the routine. It is decremented by one by the routine for each data word that is loaded. Hence, the routine tends to load consecutive data words into successively higher storage locations beginning with the base address. The change affected by an S-card is a shifting of the current loading point of the routine to a lower storage address. If this change is executed before any data are loaded, it extends the range of the storage area that can be loaded by the routine to locations below the base address. The P-cards have the opposite effect. The current loading point of the routine is shifted upward with each application of these cards. Data may be loaded between these shifts to generate any given pattern of loading desired.

An R-card is used to direct the routine to return control to the calling program. Only the column one character in this card is significant. Hence, the rest of the card may be used for comments. Unlike an *eof*, no other effects result from the use of an R-card.

An *eof*, created by placing a card containing a 7-8 punch in column one in the data deck, also causes the routine to return control to the calling program. Before this return is executed, however, the indicator location discussed above is loaded with a minus one. If a plus one, indicating a processing error, has been loaded before the *eof* is detected by the routine, the loading of the *eof* indication is suppressed.

A diagnostic is recorded on the system output with the card image of each card that contains a disallowed character or a data item that cannot be properly expressed in the required binary word form. Only one diagnostic per card is given because the processing of all subsequent items in each faulted card is deleted. The loading of all data from the point of the fault, including those from succeeding cards, is also suspended. The processing of succeeding cards for faults, however, is not affected. Therefore, the routine continues on through successive cards until a return of control to the calling program is executed.

DECIMAL DATA ITEMS

A description of the decimal data processed by the routine is given in the following paragraphs. This description incorporates those given in the IBM FORTRAN IV and MAP manuals for the applicable data items. Hence, users familiar with those computer languages should experience little if any difficulty in formulating decimal data items for this routine.

Four types of decimal data items are recognized by the routine. These have been designated decimal integers, single-precision floating point numbers, single-precision fixed point numbers, and double-precision floating point numbers.

The decimal integers have a simple format. Each integer is composed of a string of decimal digits written without a decimal point. A minus sign is required to characterize a negative integer; a plus sign is optional for a positive integer. The magnitude of the integer must not exceed 2^{35} minus one, i.e., 34359738367.

A floating point number may have two components, a principal part and an exponent part. The principal part is composed of a string of decimal digits. A decimal point must be embedded at some point within the principal part if the exponent part is absent. The decimal point may be omitted if the exponent part is present. When it is omitted, the decimal point is assumed to be located at the right-hand end of the principal part. The exponent part trails the principal part. It is composed of a letter (or two letters) and a trailing decimal integer which specifies the power of ten by which the principal part is multiplied. The exponential integers are not restricted to two significant digits. However, use of more than two significant digits will result in a conversion error and suspension of data loading.

A single-precision floating point number has a principal part which contains less than 10 decimal digits. The exponent part, if specified, is written with an 'E' and the appropriate decimal integer. A double-precision floating point number is specified when the principal part has 10 or more decimal digits. The principal part of a double-precision number may have less than 10 decimal digits if the exponent part is present and formulated with a 'D' or 'EE' and the appropriate decimal integer. A conversion error will be generated during the processing of a double-precision number if its principal part, taken as an integer by ignoring any decimal point, has a magnitude which exceeds 2^{62} minus one. A conversion error will also be indicated during the processing of both types of floating point numbers if an accumulator register floating point overflow or underflow is detected. To avoid this error, the magnitude of the floating point numbers must be limited to between 10^{38} and 10^{-38}. It should also be noted that the converted double-precision quantity will possess the significance associated with its type as long as its high-order biased exponent is equal to or larger than 33_8. When this exponent has a lesser value, the double quantity is reduced to single word significance; it low-order part is zeroed. In terms of more familiar values, this loss of significance occurs for magnitudes less than 10^{-30}.

A fixed point number may have three components - a principal part, an exponent part, and a binary place part. The latter component characterizes the fixed point number. This part trails the principal part if the exponent part is absent. It may precede or follow the exponent part when all three parts are present. It is formulated with a 'B' and a trailing decimal integer which specifies the location of the understood binary point within the converted word or its virtual extensions. The formulation of the other parts is as described in the two preceding paragraphs. It should be noted at this point that the meaning of any double-precision indicator is negated by the presence of the binary place part. The decimal point in the principal part may also be omitted even when the exponent part is not present. The assumed location of the decimal point will again be at the right-hand end of the principal part.

The restrictions and observations noted for floating point numbers also apply to fixed point numbers. This is due to the use of floating point processes to convert the fixed point number. An

additional restriction arises from the processes that produce the fixed point number from the floating point form. This requires the shifting of the number relative to the understood binary point. The binary place part specification must not cause significant bits to be shifted past the left-hand end of the word. No restriction applies to shifting bits past the right-hand end of the word. However, if the right shifting is carried to extremes, complete loss of significance may result.

Zeroes need not be explicitly indicated on a decimal data card. That is, a zero may be specified by a zero length data subfield. Hence, a decimal data card containing no apparent data item actually specifies a zero and a card punched with N successive commas only in its data field specifies N+1 zeroes. The decimal integer associated with the exponent part and the binary place part of a data item may also be omitted if it is a zero. Finally, a data subfield containing only a minus sign specifies a minus zero whereas a subfield containing a 'D' or an 'EE' specifies two zero words to be loaded into two successive storage locations.

From the preceding discussion, it is apparent that the use of two or more component parts to formulate a given data item makes it possible to express the item in several different ways. It may be that these various ways are not equally efficient. Whatever the case may be, a quantity such as π can be written in single-precision form as follows:

```
3.14159265
31415.9265E-4
314159265E-8
.314159265E1
```

Similarly, the following formulations represent the same fixed point number, i.e., they will generate words with the same bit configuration and the same understood binary point:

```
.01E5B17
1.E3B17
1E3B17
1000B17
1B17E3
```

Words with the same bit configurations as those generated by the fixed point formulas above, but with various other understood binary points, are generated by the following representations:

```
2.5E2B15
1.25E2B14
31.25B12
3.90625B9
```

OCTAL DATA

Octal data recognized by the routine consist solely of octal integers. Each integer consists of a string of octal digits. Up to 12 octal digits may be included within each string. A plus sign is neither needed nor used to indicate a positive integer. A negative octal integer is indicated with a

minus sign. This minus sign becomes meaningless if the integer contains 12 octal digits and the leading digit is a 4 or larger. This is illustrated by the following representations which can be used to generate a word containing 36 binary ones:

```
-377777777777
777777777777
-777777777777
```

Zeroes need not be explicitly indicated on an octal data card. A zero length octal data subfield specifies a word of zeroes in each bit position.

DIAGNOSTICS

The loading routine will record a diagnostic message on the system output before calling for the job termination whenever it detects an improper calling sequence. It will record a symbolic diagnostic on the system output along with each faulted card image it detects. Appendix C contains a page of sample diagnostics recorded along with the offending card image. The meaning of each of these symbolic diagnostics is as follows:

1 - improper card column 1 character
2 - improper H-card column 2 character
D - improper character in a decimal data item field
B - binary place part error
E - exponent part error
P - decimal primitive error
EB - plus and minus signs in the exponent or the binary place part
8 - octal conversion error
SNO - conversion error in the loading point relocator

Most of the symbolic diagnostic should not require further explanation. The decimal primitive referred to above is created from the conversion of an integer or the principal part of a decimal data item. The primitive cannot occupy more than 35 bit positions for an integer nor more than 62 positions for the other decimal data items. A loading point relocator error is created whenever an improper character is used within the field of this number or whenever the number exceeds 9999.

MODIFICATIONS

The operation of the routine assumes designation of logical tapes 5 and 6 as the system input and the system output, respectively. If other logical tapes have been given these designations, the routine may be modified at only three places to accommodate the different unit specification. This accommodation may be achieved without the alteration of any source or object program cards by using the alternatives provided by the $FILE and the $NAME cards (References 1 and 2).

TESTING OF THE ROUTINE

The appendices to this report contain the listings of a test program, the loading routine and the table of powers of ten used by the routine, the output generated by the program and the routine during a test run, and the input data deck processed by the routine.

The test program was designed to provide a simple demonstration of most of the capabilities of the routine. It contained a cycle in which six successive calls on the routine were executed before the recording of loaded data on the system output. This arbitrary arrangement permitted the subdivision of the processing and loading of data, the demonstration of the *eof* capability, and the exercise of the loading point relocation feature of the routine.

The test program was locked in the cycle of loading and recording of data described above unless freed by the routine. This release was given only after the routine had detected at least one card image error. It resulted in a transfer of control to a program section where an improper call on the routine was executed. As noted in a preceding section, this faulty call results in a diagnostic and job termination.

Excluding the page headings, the outputs recorded by the program and the routine were printed on separate pages. The program output for data words generated and loaded by the routine was in octal to facilitate checking the bit configurations of the words. Except for the last line, the printed output of the routine consisted entirely of symbolic diagnostics with accompanying card images. The reader is referred to a preceding section of this report for the meanings of these diagnostics. The last line recorded by the routine resulted from the improper call on the routine.

The test data were subdivided and punched into seven card groupings within the data deck. All except the last of these card groupings contained valid data items. The ends of each of these card groupings were marked by an *eof* or an R-card. Since the test program was not set up to act on the detection of an *eof* by the routine, it made no difference which one of the two cards was used to terminate card image processing. The user of this routine may choose to incorporate an *eof* procedure of his own choosing within his program according to the dictates of his job.

Blank and comment bearing cards were included in the data deck to improve the appearance of the data deck listing. These cards had no significant effect on the execution of the routine.

Octal, integer, single-precision floating point, single-precision fixed point, double-precision floating point, and zeroes and Hollerith data were punched in an arbitrary, segregated arrangement into the first six card groupings in the order listed. This data arrangement was utilized to assist checking the input data against the test program output. For the same reason, the number of data items punched into all except two cards was set to produce one line of test program output per data bearing card. These arrangements were not dictated by any test program or loading routine requirement or restriction.

The loading point relocation feature of the routine was exercised before the loading of all except the first and the last groupings of data. This relocation was necessitated by the grouping of

the data and by the use of a calling sequence wherein the routine was provided with the same initial loading point for each group of valid data. The relocation directed in each instance was sufficient to result in the loading of a continuous area in memory. Data gaps of varying lengths and arbitrary placement within the storage area designated for the data could have been generated if desired. These gaps can be placed between any two data words except those generated by a double-precision data item. These relocations could have been avoided by processing all of the valid data items during any one of the first six calls on the routine. This avoidance, however, would not have demonstrated the relocation feature of the routine.

REMARKS

This routine has been executed with the test program listed in Appendix A on a stand alone IBM 7094 and on the IBM 7094-7040 direct coupled system at GSFC. The existence of these systems with differing system input and output designations also provided an opportunity to verify the use of the $NAME card.

It is suggested that the programmer utilizing this routine group all input data locations in a single storage area such as a common block. This arrangement makes all input data locations accessible to the routine during any given call on the routine even if the initial loading point is different each time. As noted in a preceding section, the object time user must be made aware of the displacements of the various locations reserved for the input data from the initial loading point provided in each call on the routine. This information is most easily transmitted if all the input data locations are contained within a single storage area. The movement of the current loading point of the routine is more readily understood and followed by the user if he can concern himself with only one arrangement of input data locations.

The routine is not restricted to the loading of storage areas defined within the body of the calling program. It has been used to load an independently defined area of arbitrary dimensions. This capability when combined with the octal data loading capability of the routine, opens the way to some interesting programming possibilities.

The table of powers of ten utilized by the routine may be replaced by a table contained within the IBJOB library. This table was located in subroutines FIOH and FCNV of the IBJOB Library, Version 2 and Version 4, respectively. This replacement may be carried out by the inclusion of an additional entry point to the applicable library routine. The external name of the entry point must be 'D1E0'.

ACKNOWLEDGMENT

The author is especially grateful to the systems programmers at GSFC for their interest and assistance in this effort and wishes to acknowledge that this work must necessarily include revisions and adoptions of program segments and concepts established by others. The author also wishes to thank the reviewers of an earlier version of this work for their timely and helpful comments.

(Manuscript received November 5, 1964)

REFERENCES

1. "IBM 7090/7094 IBSYS Operating System: Input/Output Control System," IBM Corporation Form C28-6345-1, April 1964.

2. "IBM 7090/7094 Operating System: IBJOB Processor," IBM Corporation Form C28-6275-3, June 1964. (Superseded by form C28-6389).

Appendix A

Listing of a Test Program for Subroutine LOAD

EXTERNAL FORMULA NUMBER - SOURCE STATEMENT - INTERNAL FORMULA NUMBER(S)

```
CTEST              PROG. TO TEST SUBR. LOAD.
C
    3 FORMAT( 1H1 / 1HO,54X,10HLOAD TEST.,48X,5HPAGE ,I2 / 1HO )
    5 FORMAT( 1HO,9X,8015 )
C
      DIMENSION A(12), B(14), C(800)
      COMMON N2, A, B, C
C
C        OUTPUT THE PAGE HEADING.
C
      K = 0                                                          ,1
   10 K = K + 1                                                      ,2
      WRITE( 6,3 ) K
C
C        1ST TIME THROUGH-LOAD THE VALID DATA IN 6 SEPARATE GROUPS.
C        OUTPUT THE VALID DATA ON ONE PAGE.
C        NOTE THAT ARRAYS NOT REF'D. IN THE CALL SEQ. ARE LOADED.
C        2ND TIME THROUGH-TRY READING CARDS CONTAINING INVALID DATA.
C        THE INVALID DATA WILL RESULT IN TRANSFER TO '50 CALL LOAD'.
C                                                                    ,3    ,4    ,5
      DO 30  J = 1,6                                                 ,6
      CALL LOAD( N1,N2 )                                             ,7
      IF( N1 .GT. 0 ) GO TO 50                                       ,8    ,9    ,10
   30 CONTINUE
C                                                                    ,11   ,12
      M = N2 - 26                                                    ,13
      IF( M .LT. 1 ) GO TO 10                                        ,14   ,15   ,16
      WRITE( 6,5 ) A, B, ( C(J), J = 1,M )                           ,17   ,18   ,19   ,20   ,21   ,22

      GO TO 10
C
C        THE FOLLOWING CALL ON LOAD IS PURPOSELY INVALID.
C        THE INVALID CALL ON LOAD WILL RESULT IN JOB TERMINATION.
C                                                                    ,23
   50 CALL LOAD                                                      ,24
      IF( N1 .GT. 0 ) GO TO 90                                       ,25   ,26   ,27
      GO TO 10
C                                                                    ,28
   90 STOP                                                           ,29
      END                                                            ,30
```

Appendix B

Assembly Listing of Subroutine LOAD and the Table of Powers of Ten Used by the Subroutine

IBMAP SUBR. LOAD - 6/21/65 - L.F.H.

```
                                   SUBROUTINE LOAD.
                  -CONVERTS AND STORES BCI, OCT., AND DEC. DATA.  THE DEC.
                       DATA MAY BE INTEGER, S.P. FLOATING PT., S.P. FIXED
                       PT., AND D.P. FLOATING PT.
                  -CAN SKIP OVER STORAGE LOCATIONS AS REQUIRED TO LOAD ANY
                       GIVEN LOCATION WHICH CAN BE REFERENCED TO THE DATA
                       STORAGE LOCATION GIVEN IN THE CALLING SEQUENCE.
                  -FILLS THE CALLING PROG. INDICATOR LOC. WITH A ZERO ON
                       ENTRY, A 1 WHEN CONV. ERRORS ARE DETECTED, AND A -1
                       WHEN AN EOF IS ENCOUNTERED.  A 1 OVERRIDES A -1.
                  -STOPS THE JOB IF THE CALLING SEQUENCE IS BAD OR IF A
                       READING ERROR OCCURS.


                        ENTRY SECT.
                      ...ALLOWED C.C. 1 CHARACTERS...
                  S-MEANS SKIP TO A LOWER STORAGE LOCATION.
                  BLANK-CHARACTERIZES A COMMENTARY CARD.
                  R-MEANS RETURN TO THE CALLING ROUTINE.
                  P-MEANS SKIP TO A HIGHER STORAGE LOCATION.
                  H-DENOTES BCI DATA.
                  E-DENOTES OCTAL DATA.
                  D-DENOTES DECIMAL DATA.
                  C-CHARACTERIZES A COMMENTS CARD.
                  A 7-8 PUNCH-DENOTING AN EOF.

                                   ENTRY  LOAD

BINARY CARD ID. LOAD0002
      00000  1 00000 0 00007  10001  LOAD   SAVE   (1,2)I          SAVE THE XRS, ETC.
      00001  0774 00 2 00000  10000
      00002  0774 00 1 00000  10000
      00003  0774 00 4 00000  10000
      00004  0441 00 0 00006  10001
      00005  0020 00 4 00001  10000
      00006  0 00000 0 00000  10000
      00007  0604 00 0 00006  10001
      00010  0634 00 4 07000  10011
      00011  0634 00 4 01136  10001
      00012  0634 00 4 00003  10001
      00013  0634 00 1 00002  10001
      00014  0634 00 2 00001  10001
      00015  4500 00 4 00003  10000        CAL    3,4              GET THE CALLER ERROR INDIC. LOC...
      00016  0621 00 0 01122  10001        STA    ELOC             ...AND SAVE IT.
      00017  4320 00 0 00121  10001        ANA    MASK             ...AND MASK OUT ITS ADDRESS.
      00020  4100 00 0 01022  10001        TNZ    LGOOF            ...NOT GOOD IF NON-ZERO
      00021  0600 60 0 01122  10001        STZ*   ELOC             ZERO CALLER ERROR INDIC. LOC.
      00022  4500 00 4 00004  10000        CAL    4,4              GET THE INIT. OR REF. STORE LOC...

BINARY CARD ID. LOAD0003
      00023  0621 00 0 01016  10001        STA    SLOC             ...AND SAVE IT.
      00024  4320 00 0 00121  10001        ANA    MASK             ...AND MASK OUT ITS ADDRESS.
      00025  4100 00 0 01022  10001        TNZ    LGOOF            ...BAD IF NON-ZERO.
      00026  0600 00 0 00117  10001        STZ    TEST             INITIALIZE TEST...
      00027  4634 00 0 00065  10001        ZSD    XRA              ...AND XRA.
```

```
        IBMAP SUBR. LOAD - 6/21/65 - L.F.H.

                                                    * * *
00030  0500 00 0 10000  10011          CLA    .UN05.            SET UP THE INPUT FILE...
00031  0621 00 0 01003  10011          STA    *+3               ...
00032  0621 00 0 01004  10011          STA    *+4               ...
00033  0074 00 4 11000  10011          TSX    .OPEN,4           OPEN THE FILE AS REQUIRED...
00034  5 00000 0 00000  10000          MON    **                ...WITHOUT REWIND.
00035  0074 00 4 12000  10011  RDD     TSX    .READ,4           INPUT ONE CARD...
00036  0 01003 0 00000  11100          PZE    **,,*+3           ...
00037  0 01054 0 01047  10101          PZE    PASS,,PDMP        ...
00040  3 00016 0 00101  10001          IORT   CARD,,14          ...
                                                    * * *
00041  4500 00 0 00122  10001          CAL    MASK1             BLANK OUT ...
00042  4602 00 0 00116  10001          ORS    CARD+13           ...THE EQUIV. OF C.C. 81 - 84.
00043  0560 00 0 00101  10001          LDQ    CARD              CHECK C.C. 1...
00044  4754 00 0 00000  10000          ZAC                      ...
00045  4763 00 0 00006  10000          LGL    6                 ...

BINARY CARD ID. LOAD0004
00046  0774 00 1 00010  10000          AXT    8,1               ...FOR 1 OF 8 CHARACTERS.
00047  4600 00 0 00120  10001          STQ    MQ                ...SAVE THE MQ.
00050  0340 00 1 00065  10001  CAS1    CAS    C+1,1             ...
00051  7 00000 0 01063  10001          TXL    CGOOF,,           ...OUCH.
00052  0020 00 1 00075  10001          TRA    COP+1,1           ...A-OK.
00053  2 00001 1 00050  10001          TIX    CAS1,1,1          ...TRY AGAIN.
00054  7 00000 0 01063  10001          TXL    CGOOF,,           ...HEY.
00055  000000000062     10000          BCI    1,00000S          CHARACTER TABLE ( C.C. 1 ).
00056  000000000060     10000          BCI    1,00000                ...
00057  000000000051     10000          BCI    1,00000R               ...
00060  000000000047     10000          BCI    1,00000P               ...
00061  000000000030     10000          BCI    1,00000H               ...
00062  000000000025     10000          BCI    1,00000E               ...
00063  000000000024     10000          BCI    1,00000D               ...
00064  000000000023     10000  C       BCI    1,00000C               ...
00065  7 00000 0 00674  10001  XRA     TXL    SOP,,**           GO INCR. XRA.
00066  7 00000 0 00035  10001          TXL    RDD,,             BLANK, NEXT CARD.
00067  7 00000 0 00075  10001          TXL    RET,,             RETURN.
00070  7 00000 0 00672  10001          TXL    POP,,             GO DECR. XRA.

BINARY CARD ID. LOAD0005
00071  7 00000 0 00650  10001          TXL    HOP,,             BCI DATA.
00072  7 00000 0 00564  10001          TXL    EOP,,             OCT. DATA.
00073  7 00000 0 00124  10001          TXL    DOP,,             DEC. DATA.
00074  7 00000 0 00035  10001  COP     TXL    RDD,,             COMMENTARY, NEXT CARD.
                                                    * * *
00075  4500 00 0 00123  10001  RET     CAL    LW                RESET L(8)...
00076  0602 00 0 00010  10000          SLW    8                 ...
                00077                   RETURN LOAD
                                                    * * *
00100  200000000017     00001  DATA    BSS    15                STORAGE...
                00101          CARD    EQU    DATA+1                 ...
00117  200000000001     00001  TEST    BSS    1                      ...
00120  200000000001     00001  MQ      BSS    1                      ...
00121  777777700000     10000  MASK    OCT    777777700000
00122  000060606060     10000  MASK1   OCT    000060606060
00123  0021 00 0 13000  10011  LW      TTR    .FFPT.
```

```
                                        SECT. DOP.
                                   ...ALLOWED DEC. DATA FIELD CHARACTERS...
                              COMMA-TERMINATES DATA ITEM FIELD.
                              BLANK-TERMINATES DATA FIELD, AS DOES THE END OF CARD.
                              MINUS-REQD. FOR NEG. NOS. AND/OR TO INIT. NEG. EXP. COMP.
                              POINT-REQD. FOR CERTAIN S.P. AND D.P. FLOATING PT. DATA.
                              B-REQD. FOR 8 PLACE PART COMP. AND FIXED PT. CONV.
                              D-REQD. TO CHARACTERIZE CERTAIN D.P. FLOATING PT. DATA.
                              E-REQD. TO INIT. COMP. FOR S.P. FLOATING PT. EXP.
                              PLUS-OPTIONAL, HAS NO EFFECT.
                              DIGITS-ALL DECIMAL DIGITS.

                                        ...NOTES...
                              AN 'EE' IS EQUIVALENT TO A 'D'.
                              A 'B' NEGATES THE D.P. MEANING OF A 'D' OR AN 'EE'.
                              THE FIELD OF A ZERO NEED NOT BE EXPLICITLY STATED, BUT
                                   SHOULD NOT BE LEFT BLANK.  THAT IS, THE IMPLICIT
                                   FIELD OF A ZERO IS OF LENGTH ZERO.



      00124  4500 00 0 00201  10001  DOP    CAL    M3                RESET DSW...
      00125  0630 00 0 00155  10001         STP    DSW               ...
      00126  4500 00 0 00201  10001  DZZ    CAL    M3                RESET THE REQUIRED SWITCHES / REGS...
      00127  0630 00 0 00461  10001         STP    DCMA              ...
      00130  0630 00 0 00417  10001         STP    DE                ...
      00131  0630 00 0 00215  10001         STP    SW1               ...

BINARY CARD ID. LOAD0006
      00132  0630 00 0 00222  10001         STP    SW2               ...
      00133  0630 00 0 00235  10001         STP    SW3               ...
      00134  0630 00 0 00256  10001         STP    SW6               ...
      00135  0630 00 0 00310  10001         STP    SW7               ...
      00136  0630 00 0 00321  10001         STP    SW8               ...
      00137  0630 00 0 00467  10001         STP    SW50              ...
      00140  0630 00 0 00516  10001         STP    SW51              ...
      00141  0630 00 0 00473  10001         STP    SW55              ...
      00142  0630 00 0 00432  10001         STP    SW100             ...
      00143  0630 00 0 00433  10001         STP    SW101             ...
      00144  0630 00 0 00444  10001         STP    SW130             ...
      00145  0630 00 0 00445  10001         STP    SW131             ...
      00146  0443 00 0 01132  10001         DLD    DZERO             ...
      00147  4603 00 0 01124  10001         DST    UNO               ...
      00150  4603 00 0 01126  10001         DST    DENO              ...
      00151  4603 00 0 01130  10001         DST    DBNO              ...
      00152  0600 00 0 00535  10001         STZ    FCNT              ...
      00153  4634 00 0 00241  10001         ZSD    DCNT              ...
      00154  0140 00 0 01001  10011         TOV    *+1               TURN OFF THE OVERFLOW INDICATOR.

BINARY CARD ID. LOAD0007
      00155  7 00000 0 00157  10001  DSW    TXL    DW,,              *
      00156  7 00000 0 00551  10001         TXL    DZ,,
                                          * * *
      00157  0500 00 0 00201  10001  DW     CLA    M3                INV. DSW...
```

IBMAP SUBR. LOAD - 6/21/65 - L.F.H.

```
00160  0630 00 0 00155  10001        STP   DSW                 ...
00161  0074 00 4 00536  10001        TSX   SET,4
00162  0774 00 1 00011  10000        AXT   9,1
00163  0340 00 1 00201  10001  CAS2  CAS   NINE+1,1            CHECK THIS CHARACTER...
00164  7 00000 0 01065  10001        TXL   DGOOF1,,            ...ILLEGAL.
00165  0020 00 1 00212  10001        TRA   D9+1,1              ...VALID.
00166  2 00001 1 00163  10001        TIX   CAS2,1,1            ...TRY AGAIN.
00167  7 00000 0 00211  10001        TXL   D9,,                GOT TO BE BETWEEN 0 - 8 INCL.
00170  000000000073     10000        BCI   1,00000,            CHARACTER TABLE ( DEC. DATA FIELD ).
00171  000000000060     10000  BLK   BCI   1,00000                 ...
00172  000000000040     10000        BCI   1,00000-                ...
00173  000000000033     10000        BCI   1,00000.                ...
00174  000000000025     10000        BCI   1,00000E                ...
00175  000000000024     10000        BCI   1,00000D                ...
00176  000000000022     10000        BCI   1,00000B                ...
00177  000000000020     10000        BCI   1,00000+                ...

BINARY CARD ID. LOAD0008
00200  000000000011     10000  NINE  BCI   1,000009                ...
00201  7 00000 0 00461  10001  M3    TXL   DCMA,,              TRANSFER TABLE.
00202  7 00000 0 00457  10001  XRB   TXL   DBLK,,**                ...
00203  7 00000 0 00443  10001  XR2   TXL   DMIN,,**                ...
00204  7 00000 0 00437  10001  XRC   TXL   DPT,,**                 ...
00205  7 00000 0 00417  10001        TXL   DE,,                    ...
00206  7 00000 0 00414  10001        TXL   DD,,                    ...
00207  7 00000 0 00410  10001        TXL   DB,,                    ...
00210  7 00000 0 00431  10001        TXL   DPL,,                   ...
00211  0601 00 0 00534  10001  D9    STO   TEMP
00212  4500 00 0 00201  10001        CAL   M3                RESET SW101 AND SW131...
00213  0630 00 0 00433  10001        STP   SW101             ...
00214  0630 00 0 00445  10001        STP   SW131             ...

                                          WORD CONSTRUCTION SUBSECTION.

00215  7 00000 0 00222  10001  SW1   TXL   SW2,,             *
00216  0074 00 4 00733  10001        TSX   CMPT,4             GO COMP. BNO...
00217  0 00000 0 01130  10001        PZE   DBNO               ...
00220  0 00000 0 01066  10001        PZE   DGOOF2             ...
00221  7 00000 0 00551  10001        TXL   DZ,,
00222  7 00000 0 00227  10001  SW2   TXL   PPART,,           *

BINARY CARD ID. LOAD0009
00223  0074 00 4 00733  10001        TSX   CMPT,4             GO COMP. ENO...
00224  0 00000 0 01126  10001        PZE   DENO               ...
00225  0 00000 0 01067  10001        PZE   DGOOF3
00226  7 00000 0 00551  10001        TXL   DZ,,
00227  0074 00 4 00733  10001  PPART TSX   CMPT,4             GO COMP. THE D-PRIMITIVE...
00230  0 00000 0 01124  10001        PZE   UNO                ...
00231  0 00000 0 01070  10001        PZE   DGOOF4             ...
00232  4534 00 1 00241  10001        LXD   DCNT,1             INCREMENT THE DIGIT COUNT...
00233  1 00001 1 01001  10011        TXI   *+1,1,1            ...
00234  4634 00 1 00241  10001        SXD   DCNT,1             ...
00235  7 00000 0 00551  10001  SW3   TXL   DZ,,              *
00236  0500 00 0 00535  10001        CLA   FCNT               INCREMENT THE FRACT. COUNT...
00237  0400 00 0 01140  10001        ADD   =1                 ...
00240  0601 00 0 00535  10001        STO   FCNT               ...
```

```
00241  7 00000 0 00551  10001  DCNT   TXL    DZ,,**
                                                    * * *
00242  0443 00 0 01124  10001  INO    DLD    UNO          GET THE D-INTEGER PRIMITIVE...
00243  0100 00 0 00247  10001         TZE    SMQ          ...
00244  4773 00 0 00010  10000         RQL    8            ...PACK IT IN THE MQ.
00245  0765 00 0 00010  10000         LRS    8            ...

BINARY CARD ID. LOAD0010
00246  4100 00 0 01070  10001         TNZ    DGOOF4       ...OR TRANSFER OUT IF IMPOSSIBLE.
00247  4600 00 0 01124  10001  SMQ    STQ    UNO          STASH THE PACKED INTEGER.
00250  7 00000 0 00463  10001         TXL    PFIX,,
                                                    * * *
00251  0074 00 4 00766  10001  FLOAT  TSX    FLT,4        GO FLOAT THE D-PRIMITIVE.
00252  4500 00 0 00533  10001         CAL    LW1          SET UP L(8)...
00253  0602 00 0 00010  10000         SLW    8            ...
00254  0774 00 1 00114  10000         AXT    76,1         SET UP XR1 IN CASE 10**38 IS NEEDED.
00255  0502 00 0 00535  10001         CLS    FCNT         SET UP TO SCALE THE FLOATED NO...
00256  7 00000 0 00272  10001  SW6    TXL    QQ4,,       *
00257  0402 00 0 01127  10001         SUB    ENO          ...
00260  0100 00 0 00306  10001         TZE    FINV         ...
00261  0767 00 0 00001  10000  QQ1    ALS    1            ...
00262  0734 00 4 00000  10000         PAX    ,4           ...
00263  0443 00 0 01124  10001         DLD    UNO          GET THE FLOATED NO...
00264  3 00114 4 00270  10001  QQ2    TXH    QQ3,4,76     ...AND SCALE DOWNWARD AS REQUIRED.
00265  7 00000 4 00305  10001         TXL    QQ7,4,0      TRANSFER OUT IF SCALING IS DONE.
00266  4241 00 4 14000  10011         DFDP   D1E0,4       ...DIVIDE BY 10**N ( N .LE. 38 ).
00267  7 00000 0 00305  10001         TXL    QQ7,,        ...
00270  4241 00 1 14000  10011  QQ3    DFDP   D1E0,1       ...DIVIDE BY 10**38.

BINARY CARD ID. LOAD0011
00271  2 00114 4 00264  10001         TIX    QQ2,4,76     ...
00272  0400 00 0 01127  10001  QQ4    ADD    ENO          ...
00273  0100 00 0 00306  10001         TZE    FINV         ...
00274  4120 00 0 00261  10001         TMI    QQ1          ...
00275  0767 00 0 00001  10000         ALS    1            ...
00276  0734 00 4 00000  10000         PAX    ,4           ...
00277  0443 00 0 01124  10001         DLD    UNO          GET THE FLOATED NO...
00300  3 00114 4 00303  10001  QQ5    TXH    QQ6,4,76     ...AND SCALE UPWARD AS REQUIRED.
00301  0261 00 4 14000  10011         DFMP   D1E0,4       ...MULTIPLY BY 10**N ( N .LE. 38 ).
00302  7 00000 0 00305  10001         TXL    QQ7,,        ...
00303  0261 00 1 14000  10011  QQ6    DFMP   D1E0,1       ...MULTIPLY BY 10**38.
00304  2 00114 4 00300  10001         TIX    QQ5,4,76     ...
00305  4603 00 0 01124  10001  QQ7    DST    UNO          SAVE THE SCALED D-WORD.
00306  0500 00 0 00201  10001  FINV   CLA    M3           INV. SW55...
00307  0630 00 0 00473  10001         STP    SW55         ...
00310  7 00000 0 00466  10001  SW7    TXL    FIXS,,      *
00311  4500 00 0 00201  10001         CAL    M3           RESET SW200 AND ZERO DCNT...
00312  0630 00 0 01010  10001         STP    SW200        ...
00313  4634 00 0 00241  10001         ZSD    DCNT         ...

BINARY CARD ID. LOAD0012
00314  0140 00 0 01001  10011         TOV    *+1          TURN OFF THE OVERFLOW INDIC.
00315  0560 00 0 01124  10001         LDQ    UNO          GET THE H-ORDER FLOATED PART...
00316  4754 00 0 00000  10000         ZAC                 ...
00317  0763 00 0 00010  10000         LLS    8            ...AND SHIFT ITS EXP. INTO THE AC.
00320  0402 00 0 01141  10001         SUB    =93          COMPUTE THE SHIFTING REQD...
```

```
00321  7 00000 0 01003  10011  SW8     TXL    *+3,,          *...TRANSFER FOR A POS. BNO.
00322  0400 00 0 01131  10001          ADD    BNO             ...
00323  7 00000 0 01002  10011          TXL    *+2,,           ...
00324  0402 00 0 01131  10001          SUB    BNO             ...
00325  0100 00 0 01002  10011          TZE    *+2
00326  4120 00 0 00366  10001          TMI    RSHFT           ...TRANSFER IF R-SHIFTING IS REQD.
00327  0340 00 0 01142  10001          CAS    =27             CHECK THE L-SHIFT REQUIRED...
00330  0761 00 0 00000  10000          NOP                    ...GT. 27.
00331  7 00000 0 00352  10001          TXL    SUB,,           ...EQ. 27.
00332  0621 00 0 00335  10001          STA    SHFT1           SET UP A LONG L-SHIFT...
00333  0621 00 0 00340  10001          STA    SHFT2           ...AND AN AC R-SHIFT.
00334  4754 00 0 00000  10000          ZAC                    ...
00335  0763 00 0 00000  10000  SHFT1   LLS    **             LONG L-SHIFT 26 OR LESS.
00336  0602 00 0 01124  10001          SLW    UNO             SAVE THE H-ORDER SHIFTED PART.

BINARY CARD ID. LOAD0013
00337  0131 00 0 00000  10000          XCA
00340  0771 00 0 00000  10000  SHFT2   ARS    **             AC R-SHIFT 26 OR LESS.
00341  0560 00 0 01125  10001          LDQ    UNO+1           GET THE L-ORDER PART.
00342  4773 00 0 00011  10000          RQL    9               ERASE THE L-ORDER EXP...
00343  4765 00 0 00011  10000          LGR    9               ...
00344  4763 00 0 00001  10000          LGL    1               ...
00345  4773 00 0 00007  10000          RQL    7               ...
00346  0763 00 0 00010  10000          LLS    8              REINITIALIZE.
00347  0522 00 0 00335  10001          XEC    SHFT1           EXECUTE THE 1ST L-SHIFT AGAIN.
00350  0602 00 0 01125  10001          SLW    UNO+1           SAVE THE SHIFTED L-ORDER PART.
00351  7 00000 0 00466  10001          TXL    FIXS,,
00352  0402 00 0 01142  10001  SUB     SUB    =27             DECR. THE L-SHIFT COUNT BY 27.
00353  0621 00 0 00362  10001          STA    SHFT3           SET UP THE LONG L-SHIFT.
00354  0131 00 0 00000  10000          XCA
00355  0560 00 0 01125  10001          LDQ    UNO+1           GET THE L-ORDER PART.
00356  4773 00 0 00011  10000          RQL    9               ERASE THE L-ORDER EXP...
00357  4765 00 0 00011  10000          LGR    9               ...
00360  4763 00 0 00001  10000          LGL    1               ...
00361  4773 00 0 00007  10000          RQL    7               ...

BINARY CARD ID. LOAD0014
00362  0763 00 0 00000  10000  SHFT3   LLS    **             COMPLETE THE L-SHIFTING.
00363  0140 00 0 01066  10001          TOV    DGOOF2          TRANSFER ON OVERFLOW.
00364  4603 00 0 01124  10001          DST    UNO             SAVE BOTH SHIFTED PARTS.
00365  7 00000 0 00466  10001          TXL    FIXS,,
00366  0340 00 0 01143  10001  RSHFT   CAS    =-8             CHECK THE R-SHIFT REQD...
00367  7 00000 0 00376  10001          TXL    SHFT5,,         ...TRANSFER IF .LT. 8.
00370  0761 00 0 00000  10000          NOP                    ...
00371  0621 00 0 00373  10001          STA    SHFT4           SET UP THE R-SHIFT FOR 8 OR MORE...
00372  4754 00 0 00000  10000          ZAC                    ...
00373  0765 00 0 00000  10000  SHFT4   LRS    **             R-SHIFT THE WORKS...
00374  4603 00 0 01124  10001          DST    UNO             ...AND STORE THE RESULT.
00375  7 00000 0 00466  10001          TXL    FIXS,,
00376  0400 00 0 01144  10001  SHFT5   ADD    =8              RESET AC FOR L-SHIFT...
00377  0621 00 0 00404  10001          STA    SHFT6           ...AND STORE THE RESULT.
00400  0131 00 0 00000  10000          XCA                    SET UP THE REGISTERS...
00401  0771 00 0 00010  10000          ARS    8               ...
00402  0560 00 0 01125  10001          LDQ    UNO+1           ...
00403  4773 00 0 00011  10000          RQL    9               ...
00404  4763 00 0 00000  10000  SHFT6   LGL    **             L-SHIFT THE WORKS...
```

```
BINARY CARD ID. LOAD0015
      00405  0601 00 0 01125  10001          STO    UNO+1              ...AND SAVE THE AC ONLY.
      00406  0600 00 0 01124  10001          STZ    UNO                ZERO THE H-ORDER PART.
      00407  7 00000 0 00466  10001          TXL    FIXS,,

                                                    SWITCH SETTING SUBSECTION.

      00410  0500 00 0 00201  10001  DB      CLA    M3
      00411  0630 00 0 00215  10001          STP    SW1
      00412  0630 00 0 00310  10001          STP    SW7
      00413  7 00000 0 00426  10001          TXL    DE2,,
      00414  0500 00 0 00201  10001  DD      CLA    M3
      00415  0630 00 0 01010  10001          STP    SW200
      00416  7 00000 0 00421  10001          TXL    DE1,,
      00417  7 00000 0 00421  10001  DE      TXL    DE1,,              *
      00420  7 00000 0 00414  10001          TXL    DD,,
      00421  4500 00 0 00201  10001  DE1     CAL    M3
      00422  0630 00 0 00215  10001          STP    SW1
      00423  0500 00 0 00201  10001          CLA    M3
      00424  0630 00 0 00417  10001          STP    DE
      00425  0630 00 0 00222  10001          STP    SW2
      00426  0630 00 0 00432  10001  DE2     STP    SW100
      00427  0630 00 0 00444  10001          STP    SW130

BINAPY CARD ID. LOAD0016
      00430  7 00000 0 00441  10001          TXL    DPT1,,
      00431  0500 00 0 00201  10001  DPL     CLA    M3
      00432  7 00000 0 00551  10001  SW100   TXL    DZ,,               *
      00433  7 00000 0 00435  10001  SW101   TXL    DPL1,,             *
      00434  7 00000 0 01071  10001          TXL    DGOOF5,,           -,+ IN EXP. OR B-PLACE PART.
      00435  0630 00 0 00445  10001  DPL1    STP    SW131
      00436  7 00000 0 00551  10001          TXL    DZ,,
      00437  0500 00 0 00201  10001  DPT     CLA    M3
      00440  0630 00 0 00235  10001          STP    SW3
      00441  0630 00 0 00461  10001  DPT1    STP    DCMA
      00442  7 00000 0 00551  10001          TXL    DZ,,
      00443  0500 00 0 00215  10001  DMIN    CLA    SW1
      00444  7 00000 0 00447  10001  SW130   TXL    DMIN1,,            *
      00445  7 00000 0 00451  10001  SW131   TXL    DMIN2,,            *
      00446  7 00000 0 01071  10001          TXL    DGOOF5,,           +,- IN EXP. OR B-PLACE PART.
      00447  0630 00 0 00467  10001  DMIN1   STP    SW50
      00450  7 00000 0 00551  10001          TXL    DZ,,
      00451  0630 00 0 00433  10001  DMIN2   STP    SW101
      00452  4120 00 0 00455  10001          TMI    DMIN3

BINAPY CARD ID. LOAD0017
      00453  0630 00 0 00321  10001          STP    SW8
      00454  7 00000 0 00551  10001          TXL    DZ,,
      00455  0630 00 0 00256  10001  DMIN3   STP    SW6
      00456  7 00000 0 00551  10001          TXL    DZ,,
      00457  0500 00 0 00201  10001  DBLK    CLA    M3
      00460  0630 00 0 00516  10001          STP    SW51
      00461  7 00000 0 00242  10001  DCMA    TXL    INO,,              *
      00462  7 00000 0 00251  10001          TXL    FLOAT,,

                                                    SIGN AND ROUNDING SUBSECTION.
```

```
00463  4500 00 0 00201  10001  PFIX   CAL   M3              RESET SW55 AND SW200...
00464  0630 00 0 00473  10001         STP   SW55            ...
00465  0630 00 0 01010  10001         STP   SW200           ...
00466  0443 00 0 01124  10001  FIXS   DLD   UNO             GET THE DATA WORD.
00467  7 00000 0 01003  10011  SW50   TXL   *+3,,           *
00470  4760 00 0 00003  10000         SSM                   ...MAKE IT NEG.
00471  0765 00 0 00000  10000         LRS   0               ...
00472  4603 00 0 01124  10001         DST   ITEM            SET IT UP FOR STOR.
00473  7 00000 0 00515  10001  SW55   TXL   REL,,           *
00474  4534 00 1 00241  10001         LXD   DCNT,1          CHECK DCNT...
00475  3 00011 1 00513  10001         TXH   INV,1,9         ...TRANSFER IF .GT. 9.

BINARY CARD ID. LOAD0018
00476  0500 00 0 01010  10001         CLA   SW200           CHECK SW200...
00477  0120 00 0 00515  10001         TPL   REL             ...TRANSFER IF PLUS.
00500  0500 00 0 00310  10001         CLA   SW7             CHECK SW7...
00501  0120 00 0 00506  10001         TPL   RND             ...TRANSFER IF PLUS.
00502  0500 00 0 01124  10001         CLA   ITEM            RESTORE THE AC...
00503  0760 00 0 00011  10000         FRN                   ...AND F-ROUND IT.
00504  0601 00 0 01124  10001         STO   ITEM            STORE THE ROUNDED AC.
00505  7 00000 0 00515  10001         TXL   REL,,
00506  0500 00 0 01124  10001  RND    CLA   ITEM            RESTORE THE AC...
00507  0760 00 0 00010  10000         RND                   ...AND ROUND IT OUT.
00510  0140 00 0 01070  10001         TOV   DGOOF4          TRANSFER OUT ON OVERFLOW.
00511  0601 00 0 01124  10001         STO   ITEM              STORE THE ROUNDED ITEM.
00512  7 00000 0 00515  10001         TXL   REL,,
00513  0500 00 0 00201  10001  INV    CLA   M3
00514  0630 00 0 01010  10001         STP   SW200           ...
00515  0074 00 4 01004  10001  REL    TSX   STOR,4          GO RELOCATE ITEM AS REQUIRED.
00516  7 00000 0 00126  10001  SW51   TXL   DZZ,,           *GO RESET SWITCHES, REG., ETC...
00517  7 00000 0 00035  10001         TXL   RDD,,           ...OR GO READ ANOTHER CARD.

                                          MISC. SUBSECTION.

00520  4603 00 0 01134  10001  CHEK   DST   TMP1            SAVE THE AC AND MQ.

BINARY CARD ID. LOAD0019
00521  4500 00 0 00000  10000         CAL   0               GET THE CONTENTS OF L(0)...
00522  0044 00 0 00000  10000         PAI                   ...AND STASH IT IN THE INDICATORS.
00523  0621 00 0 00532  10001         STA   THERE           SET UP THE RETURN.
00524  4054 00  000004  10000         LFT   4               CHECK FOR AN AC OVERFLOW...
00525  0020 00 0 01067  10001         TRA   DGOOF3          ...POW.
00526  4054 00  000002  10000         LFT   2               CHECK FOR AN AC UNDERFLOW...
00527  0020 00 0 01067  10001         TRA   DGOOF3          ...BLAM.
00530  0600 00 0 01135  10001         STZ   TMP2            ...
00531  0443 00 0 01134  10001         DLD   TMP1            RESET THE AC AND MQ.
00532  0020 00 0 00000  10000  THERE  TRA   **
                                          * * *
00533  0021 00 0 00520  10001  LW1    TTR   CHEK
00534  200000000001     00001  TEMP   BSS   1
00535  200000000001     00001  FCNT   BSS   1
```

```
                                        SECT. SET.
                                SET-IS ENTERED THROUGH...TSX   SET,4
                                                    ...AXT   N,1
                                   -EXECUTES THE 'AXT   N,1' INSTRUCTION TO SET UP THE
                                        COMPARISON OF A BCD CHARACTER SHIFTED INTO THE AC
                                        AGAINST N ALLOWED BCD CHARACTERS.
                                   -IS USED BY BOTH SECT. DOP AND EOP FOR SAVING, RESTOR-
                                        ING, AND LOADING THE MQ WITH BCD DATA FROM 'CARD'
                                        TO 'CARD+13'.


        00536  0774 00 2 00000  10000  SET   AXT   ,2              INITIALIZE XRB AND XR2...
        00537  4634 00 2 00202  10001        SXD   XRB,2           ...
        00540  0774 00 2 00005  10000        AXT   5,2             ...
        00541  4634 00 2 00203  10001  DX    SXD   XR2,2           ...
        00542  0522 00 4 00001  10000  DY    XEC   1,4             SET UP XR1...
        00543  0560 00 0 00120  10001        LDQ   MQ              RESTORE THE MQ.

BINARY CARD ID. LOAD0020
        00544  4754 00 0 00000  10000        ZAC                 CLEAR THE AC.
        00545  4763 00 0 00006  10000        LGL   6               SHIFT IN A CHARACTER.
        00546  4600 00 0 00120  10001        STQ   MQ              SAVE THE MQ.
        00547  4634 00 4 00204  10001        SXD   XRC,4           SAVE XR4.
        00550  0020 00 4 00002  10000        TRA   2,4
        00551  4534 00 4 00204  10001  DZ    LXD   XRC,4           RESTORE XR4.
        00552  4534 00 2 00203  10001        LXD   XR2,2           RESTORE XR2...
        00553  2 00001 2 00541  10001        TIX   DX,2,1          ...DECR-LOOP OR...
        00554  0774 00 2 00006  10000        AXT   6,2             ...RESET XR2
        00555  4634 00 2 00203  10001        SXD   XR2,2           ...AND SAVE IT.
        00556  4534 00 2 00202  10001        LXD   XRB,2           RESTORE XRB.
        00557  0500 00 2 00102  10001        CLA   CARD+1,2        GET THE NEXT WORD...
        00560  0601 00 0 00120  10001        STO   MQ              ...AND STORE IT IN LOC( MQ ).
        00561  1 77777 2 01001  10011        TXI   *+1,2,-1        DECR. XRB...
        00562  4634 00 2 00202  10001        SXD   XRB,2           ...AND SAVE IT.
        00563  7 00000 0 00542  10001        TXL   DY,,            LOOP.
```

```
                                                                SECT. EOP.
                                                     ...ALLOWED OCT. DATA FIELD CHARACTERS...
                                                COMMA-TERMINATES OCTAL DATA ITEM FIELD.
                                                BLANK-TERMINATES THE OCTAL DATA FIELD.
                                                MINUS-REQD. FOR NEG. ITEMS OF LESS THAN 12 DIGITS.
                                                DIGITS-ALL OCTAL DIGITS.


      00564  4500 00 0 00201  10001  EOP    CAL    M3                 RESET ESW...
      00565  0630 00 0 00575  10001         STP    ESW                ...
      00566  0500 00 0 00201  10001  ESET   CLA    M3                 RESET SWITCHES / REGISTERS...

BINARY CARD ID. LOAD0021
      00567  0630 00 0 00620  10001         STP    ESW1               ...
      00570  4500 00 0 00201  10001         CAL    M3                 ...
      00571  0630 00 0 00637  10001         STP    ESW2               ...
      00572  0630 00 0 00643  10001         STP    ESW3               ...
      00573  0600 00 0 01135  10001         STZ    EWD                ...
      00574  0140 00 0 01001  10011         TOV    *+1                TURN OFF THE OVERFLOW INDICATOR.
      00575  7 00000 0 00577  10001  ESW    TXL    EW,,               *
      00576  7 00000 0 00551  10001         TXL    DZ,,
                                                    * * *
      00577  0500 00 0 00201  10001  EW     CLA    M3                 INV. ESW...
      00600  0630 00 0 00575  10001         STP    ESW                ...
      00601  0074 00 4 00536  10001         TSX    SET,4
      00602  0774 00 1 00004  10000         AXT    4,1
      00603  0340 00 1 00614  10001  CAS3   CAS    SVN+1,1            CHECK THE AC CHARACTER...
      00604  7 00000 0 01072  10001         TXL    EGOOF,,            ...ILLEGAL.
      00605  0020 00 1 00620  10001         TRA    E7+1,1             ...VALID.
      00606  2 00001 1 00603  10001         TIX    CAS3,1,1           ...TRY AGAIN.
      00607  7 00000 0 00617  10001         TXL    E7,,               ...GOT TO BE 0 - 6.
      00610  000000000073     10000         BCI    1,00000,           CHARACTER TABLE ( OCT. DATA FIELD ).
      00611  000000000060     10000         BCI    1,00000                 ...

BINARY CARD ID. LOAD0022
      00612  000000000040     10000         BCI    1,00000-                ...
      00613  000000000007     10000  SVN    BCI    1,000007                ...
      00614  7 00000 0 00636  10001         TXL    ECMA,,             TRANSFER TABLE
      00615  7 00000 0 00634  10001         TXL    EBLK,,                  ...
      00616  7 00000 0 00645  10001         TXL    EMIN,,                  ...
      00617  0601 00 0 00534  10001  E7     STO    TEMP
                                                    * * *
      00620  3 00000 0 01072  10001  ESW1   TXH    EGOOF,,            *TRANSFER TO TROUBLE.
      00621  4500 00 0 01135  10001         CAL    EWD                BUILD THE EWD...
      00622  0767 00 0 00002  10000         ALS    2                  ...
      00623  0140 00 0 01072  10001         TOV    EGOOF              ...WHAM.
      00624  0767 00 0 00001  10000         ALS    1                  ...
      00625  0400 00 0 00534  10001         ADD    TEMP               ...
      00626  0602 00 0 01135  10001         SLW    EWD                ...
      00627  0140 00 0 00631  10001         TOV    EINV               ...
      00630  7 00000 0 00551  10001         TXL    DZ,,               ...DONE.
      00631  4500 00 0 00201  10001  EINV   CAL    M3                 INV. ESW1...
      00632  0630 00 0 00620  10001         STP    ESW1               ...
      00633  7 00000 0 00551  10001         TXL    DZ,,
```

```
                                                    * * *
00634  0500 00 0 00201  10001  EBLK   CLA    M3              INV. ESW3...

BINARY CARD ID. LOAD0023
00635  0630 00 0 00643  10001         STP    ESW3            ...
00636  0500 00 0 01135  10001  ECMA   CLA    EWD             GET THE EWD...
00637  7 00000 0 01002  10011  ESW2   TXL    *+2,,           *...SKIP OR...
00640  4760 00 0 00003  10000         SSM                    ...SET SIGN MINUS...
00641  0601 00 0 01124  10001         STO    ITEM            ...AND STORE IT.
00642  0074 00 4 01004  10001         TSX    STOR,4          GO RELOCATE THE ITEM.
00643  7 00000 0 00566  10001  ESW3   TXL    ESET,,          *RESET SWITCHES / REG. OR...
00644  7 00000 0 00035  10001         TXL    RDD,,           ...GO READ ANOTHER CARD.
00645  0500 00 0 00201  10001  EMIN   CLA    M3              INV. ESW2...
00646  0630 00 0 00637  10001         STP    ESW2            ...
00647  7 00000 0 00551  10001         TXL    DZ,,
                01135           EWD    EQU    TMP2
```

SECT. HOP.
HOP-CHECKS C.C. 2 FOR BCI WORD COUNT.
-TAKES A ZERO IN C.C. 2 TO BE .EQ. 10.
-EXPECTS THE BCI WORD COUNT TO BE .LE. 10/CARD.
-EXPECTS BCI FIELDS TO BEGIN FROM C.C. 7.
-PLACES BCI WORDS IN ITEM FOR RELOCATION BY STOR.

```
00650  4754 00 0 00000  10000  HOP    ZAC                    CLEAR THE AC.
00651  4763 00 0 00006  10000         LGL    6               SHIFT IN C.C. 2.
00652  0340 00 0 00200  10001         CAS    NINE            COMPARE WITH 9.
00653  7 00000 0 01064  10001         TXL    HGOOF,,         ...TOO LARGE.
00654  7 00000 0 00656  10001         TXL    NINER,,         ...TIS NINE.
00655  0100 00 0 00667  10001         TZE    TENNER          ...TIS ZERO, DENOTING TEN.
00656  0734 00 1 00000  10000  NINER  PAX    ,1              SET UP XR1...
00657  0734 00 2 00000  10000         PAX    ,2              ...AND XR2.

BINARY CARD ID. LOAD0024
00660  1 00102 1 01001  10111         TXI    *+1,1,CARD+1    COMPUTE ADDR...
00661  0634 00 1 01001  10011         SXA    *+1,1           ...FOR STORAGE...
00662  4500 00 2 00000  10000         CAL    **,2            ...HERE.
00663  0602 00 0 01124  10001         SLW    ITEM            PLACE BCI WORD IN ITEM.
00664  0074 00 4 01004  10001         TSX    STOR,4          LET STOR RELOCATE IT.
00665  2 00001 2 41003  10011         TIX    *-3,2,1         NEXT WORD.
00666  7 00000 0 00035  10001         TXL    RDD,,           NEXT CARD.
00667  4534 00 1 00671  10001  TENNER LXD    H10,1           SET UP XR1...
00670  4534 00 2 00671  10001         LXD    H10,2           ...AND XR2.
00671  7 00012 0 00660  10001  H10    TXL    NINER+2,,10
```

```
                                        SECT. POP / SOP.
                                ENTRY THROUGH POP RESULTS IN DECR. XRA ( AND SKIPPING TO
                                        A HIGHER STORING LOC. ).
                                ENTRY THROUGH SOP RESULTS IN INCR. XRA ( AND SKIPPING TO
                                        A LOWER STORING LOC. ).
                                SKIPPING NO., SNO, MUST BEGIN IN C.C. 2 AND NOT EXTEND
                                        BEYOND C.C. 5 ( I.E., IT CANNOT BE .GT. 9999 ).


      00672  0500 00 0 00201  10001  POP    CLA    M3              SET UP SWA...
      00673  7 00000 0 01002  10011         TXL    *+2,,           ...
      00674  4500 00 0 00201  10001  SOP    CAL    M3              ...
      00675  0630 00 0 00724  10001         STP    SWA             ...
      00676  0600 00 0 01134  10001         STZ    TMP1            ZERO TMP1...
      00677  0600 00 0 01135  10001         STZ    SNO             ...AND SNO.
      00700  4754 00 0 00000  10000  S1     ZAC                    CLEAR THE AC.
      00701  4763 00 0 00006  10000         LGL    6               SHIFT IN A CHARACTER.
      00702  0340 00 0 00200  10001         CAS    NINE            IS IT A NINE...

BINARY CARD ID. LOAD0025
      00703  7 00000 0 00717  10001         TXL    SBLK,,          ...NO, PERHAPS TIS A BLANK.
      00704  0761 00 0 00000  10000         NOP                    ...YES.
      00705  0601 00 0 00534  10001         STO    TEMP            ...NO, BUT ITS OK.
      00706  4600 00 0 00120  10001         STQ    MQ              SAVE THE MQ.
      00707  0074 00 4 00733  10001         TSX    CMPT,4          GO COMP. SNO.
      00710  0 00000 0 01134  10001         PZE    TMP1            ...
      00711  0 00000 0 01073  10001         PZE    SGOOF           ...
      00712  0500 00 0 01135  10001         CLA    SNO             CHECK SNO.
      00713  0402 00 0 01145  10001         SUB    =10000          ...
      00714  0120 00 0 01073  10001         TPL    SGOOF           ...BAD IF PLUS.
      00715  0560 00 0 00120  10001         LDQ    MQ              RESTORE THE MQ.
      00716  7 00000 0 00700  10001         TXL    S1,,            LOOP.
      00717  0340 00 0 00171  10001  SBLK   CAS    BLK             IS IT A BLANK...
      00720  7 00000 0 01073  10001         TXL    SGOOF,,         ...NO, TOO BAD.
      00721  7 00000 0 01002  10011         TXL    *+2,,           ...YES.
      00722  7 00000 0 01073  10001         TXL    SGOOF,,         ...NO, HOW SAD.
      00723  0534 00 1 01135  10001         LXA    SNO,1           INCR. ( OR DECR. ) XRA AS REQD...
      00724  7 00000 0 01002  10011  SWA    TXL    *+2,,           *
      00725  0535 00 1 01135  10001         LAC    SNO,1           ...

BINARY CARD ID. LOAD0026
      00726  4634 00 1 01002  10011         SXD    *+2,1           ...
      00727  4534 00 1 00065  10001         LXD    XRA,1           ...
      00730  1 00000 1 01001  10011         TXI    *+1,1,**        ...
      00731  4634 00 1 00065  10001         SXD    XRA,1           ...
      00732  7 00000 0 00035  10001         TXL    RDD,,           NEXT CARD.
                    01135          SNO     EQU    TMP2
```

```
                                SECT. CMPT.
CMPT-IS ENTERED THROUGH...TSX   CMPT,4
                         ...PZE   LOC1
                         ...PZE   LOC2
    -MULTS. LOC1 D-PRIMITIVE BY 10 AND ADDS TEMP TO IT.
    -MAINTAINS THE FORMAT OF THE PRIMITIVE.
    -TRANSFERS ON OVERFLOW TO LOC2.


THE D-PRIMITIVE FORMAT IS AS FOLLOWS - A 27 BIT L-ORDER
     PART AND A 35 BIT H-ORDER PART.  CMPT BUILDS THE
     PRIMITIVE IN THE L-ORDER PART AND TRANSFERS ALL OVER-
     FLOW INTO THE H-ORDER PART.  AN OVERFLOW IN THE
     H-ORDER PART RESULTS IN AN APPROPRIATE DIAGNOSTIC.

 00733  0140 00 0 01001  10011  CMPT    TOV    *+1              TURN OFF THE OVERFLOW INDICATOR.
 00734  0443 60 4 00001  10000          DLD*   1,4              GET THE PRIMITIVE...
 00735  4603 00 0 01134  10001          DST    TMP1             ...
 00736  C774 00 1 00002  10000          AXT    2,1              SET UP XR1...
 00737  4100 00 0 01002  10011          TNZ    *+2              ...
 00740  1 77777 1 01001  10011          TXI    *+1,1,-1         ...RESET XR1 IF DESIRABLE...
 00741  4500 00 1 01136  10001  CMPT1   CAL    TMP1+2,1         ...TO MULT. THE D-PRIMITIVE BY 10.
 00742  0767 00 0 00003  10000          ALS    3                ...
 00743  0400 00 1 01136  10001          ADD    TMP1+2,1         ...
 00744  0400 00 1 01136  10001          ADD    TMP1+2,1         ...
 00745  0602 00 1 01136  10001          SLW    TMP1+2,1         ...
 00746  2 00001 1 00741  10001          TIX    CMPT1,1,1
 00747  0400 00 0 00534  10001          ADD    TEMP             ADD TEMP TO THE L-ORDER PART...
 00750  0602 00 0 01135  10001          SLW    TMP2             ...AND SAVE THE RESULT.

BINARY CARD ID. LOAD0027
 00751  4320 00 0 01146  10001          ANA    =25588           SAVE THE BITS IN POS. 1 - 8...
 00752  0100 00 0 00763  10001          TZE    CMPT2            TRANSFER IF ZERO.
 00753  4130 00 0 00000  10000          XCL                     ...
 00754  4754 00 0 00000  10000          ZAC                     ...
 00755  0763 00 0 00010  10000          LLS    8                ...
 00756  0400 00 0 01134  10001          ADD    TMP1             ...AND ADD THE H-ORDER PART.
 00757  0140 60 4 00002  10000          TOV*   2,4              TRANSFER OUT IF TOO MUCH.
 00760  0602 00 0 01134  10001          SLW    TMP1             SAVE THE H-ORDER PART.
 00761  4500 00 0 01147  10001          CAL    =0777777777      GET A MASK...
 00762  0320 00 0 01135  10001          ANS    TMP2             ...AND BLANK OUT EXCESS BITS IN TMP2.
 00763  0443 00 0 01134  10001  CMPT2   DLD    TMP1             RETURN THE RESULTS...
 00764  4603 60 4 00001  10000          DST*   1,4              ...
 00765  0020 00 4 00003  10000          TRA    3,4
```

```
                                                    SECT. FLT.
                                          FLT-IS ENTERED THROUGH...TSX   FLT,4
                                             -FLOATS THE DOUBLE PRIMITIVE IN L(UNO) AND L(UNO+1).


                                          SEE SECT. CMPT ABOVE FOR A DESCRIPTION OF THE D-PRIMITIVE
                                               FORMAT.

     00766  4500 00 0 01125  10001  FLT    CAL    UNO+1              AFFIX AN EXP. TO THE L-ORDER PART...
     00767  4501 00 0 01150  10001         ORA    =15588             ...
     00770  0602 00 0 01125  10001         SLW    UNO+1              ...
     00771  4500 00 0 01124  10001         CAL    UNO                DO THE SAME FOR THE H-ORDER PART...
     00772  0100 00 0 01001  10001         TZE    X                  ...
     00773  0765 00 0 00010  10000         LRS    8                  ...

BINARY CARD ID. LOAD0028
     00774  4501 00 0 01151  10001         ORA    =19088             ...
     00775  0602 00 0 01124  10001         SLW    UNO                ...
     00776  4500 00 0 01152  10001         CAL    =182816            ...
     00777  0763 00 0 00010  10000         LLS    8                  ...
     01000  0300 00 0 01124  10001         FAD    UNO                ...
     01001  0300 00 0 01125  10001  X      FAD    UNO+1              NORMALIZE THE H-ORDER PART...
     01002  4603 00 0 01124  10001         DST    UNO                ...AND SAVE THE D-RESULT.
     01003  0020 00 4 00001  10000         TRA    1,4



                                                    SECT. STOR.
                                          STOR-IS ENTERED THROUGH...TSX STOR,4
                                              -STORES THE WORD IN ITEM AS DIRECTED.
                                              -RETURNS WITHOUT STORING IF TEST .NE. ZERO.


     01004  0500 00 0 00117  10001  STOR   CLA    TEST               CHECK TEST...
     01005  4100 00 4 00001  10000         TNZ    1,4                ...AND RETURN IF NON-ZERO.
     01006  4534 00 1 00065  10001         LXD    XRA,1              SET UP XR1.
     01007  0443 00 0 01124  10001         DLD    ITEM               GET THE DOUBLE WORD...
     01010  7 00000 0 01016  10001  SW200  TXL    SLOC,,             *...AND TRANSFER FOR S-WORD.
     01011  4603 60 0 01016  10001         DST*   SLOC               RELOCATE THE D-WORD.
     01012  1 77776 1 01001  10011         TXI    *+1,1,-2           D-DECR. XR1.
     01013  4500 00 0 00201  10001         CAL    M3                 RESTORE SW200...
     01014  0630 00 0 01010  10001         STP    SW200              ...
     01015  0020 00 0 01020  10001         TRA    SX1
     01016  0601 00 1 00000  10000  SLOC   STO    **,1               RELOCATE THE S-WORD.

BINARY CARD ID. LOAD0029
     01017  1 77777 1 01001  10011         TXI    *+1,1,-1           DECREMENT XR1...
     01020  4634 00 1 00065  10001  SX1    SXD    XRA,1              ...AND SAVE IT.
     01021  0020 00 4 00001  10000         TRA    1,4
```

```
                                        GOOF OUTPUT SECT.
                                    LGOOF - ENTERED FROM ENTRY SECT. FOR CALLING ERROR.
                                    PASS - ENTERED FROM ENTRY SECT. FOR EOF.
                                    PDMP - ENTERED FROM ENTRY SECT. FOR READING ERROR.
                                    CGOOF - ENTERED FROM ENTRY SECT. FOR C.C. 1 ERROR.
                                    HGOOF - ENTERED FROM SECT. HOP FOR C.C. 2 ERROR.
                                    DGOOFX - ENTERED FROM SECT. DOP FOR VAR. DEC. ERRORS.
                                    EGOOF - ENTERED FROM SECT. EOP FOR OCT. CONV. ERRORS.
                                    SGOOF - ENTERED FROM SECT. POP / SOP FOR SKIP ERROR.


                                    -A CALLING ERROR RESULTS IN JOB TERMINATION.
                                    -AN EOF RESULTS IN THE LOADING OF A -1 IN THE CALL. PROG.
                                         INDICATOR LOC. FOLLOWED BY A RETURN TO THE CALLER.
                                         THIS LOADING IS NOT EXECUTED IF A 1 HAS BEEN LOADED
                                         AS DESCRIBED BELOW.
                                    -A READING ERROR RESULTS IN A DUMP OF LOCATIONS 'CARD' TO
                                         'CARD+13' FOLLOWED BY JOB TERMINATION.
                                    -CONVERSION ERRORS RESULT IN APPROPRIATE DIAGNOSTICS
                                         ( LIMITED ONE TO A CARD ) AND THE LOADING OF A 1 IN
                                         THE CALL. PROG. INDICATOR LOC.



 01022  0500 00 0 15000  10011  LGOOF  CLA    .UN06.            SET UP THE OUTPUT FILE...
 01023  0621 00 0 01003  10011         STA    *+3               ...
 01024  0621 00 0 01004  10011         STA    *+4               ...
 01025  0074 00 4 11000  10011         TSX    .OPEN,4           OPEN AS REQUIRED...
 01026  5 00000 0 00000  10000         MON    **                ...
 01027  0074 00 4 16000  10011         TSX    .WRITE,4          OUTPUT THE SAD STORY...
 01030  0 01002 0 00000  11100         PZE    **,,*+2           ...
 01031  0 00012 0 01035  10001         IOCD   TALE,,10          ...
 01032  000000000000     00010  XIT    CALL   .EXIT.            STOP THE JOB.
 01032  0074 00 4 05000  10011
 01033  1 00000 0 01002  10011
 01034  0 01136 0 01272  10100
 01035  006060606060     10000  TALE   BCI    5,0
 01036  606060606060     10000
 01037  606060606060     10000
 01040  606060606060     10000

BINARY CARD ID. LOAD0030
 01041  606060606060     10000
 01042  545454604346     10000         BCI    5,*** LOAD CALLING ERROR ***
 01043  212460232143     10000
 01044  433145276025     10000
 01045  515146516054     10000
 01046  545460606060     10000
                                              * * *
 01047  0520 00 0 00117  10001  PASS   ZET    TEST              CHECK TEST...
 01050  0020 00 0 00075  10001         TRA    RET               ...TRANSFER IF TEST IS NOT ZERO.
 01051  0500 00 0 01153  10001         CLA    =-1               SET UP THE EOF INDICATION...
 01052  0601 60 0 01122  10001         STO*   ELOC              ...
 01053  0020 00 0 00075  10001         TRA    RET
```

IBMAP SUBR. LOAD - 6/21/65 - L.F.H.

```
                                                    * * *
                                  PDMP   CALL   PDUMP              CHECK LOC. CARD TO CARD+13...
                                         ETC    (CARD              ...
                                         ETC    ,CARD+13           ...
01054  000000000000      00010           ETC    ,=0)               ...
01054  0074 00 4 06000   10011
01055  1 00003 0 01005   10011
01056  0 01136 0 01304   10100
01057  0 00000 0 00101   10001
01060  0 00000 0 00116   10001
01061  0 00000 0 01154   10001
01062  0020 00 0 01032   10001           TRA    XIT
                                                    * * *
BINARY CARD ID. LOAD0031
01063  0074 00 4 01105   10001  CGOOF    TSX    WGOOF,4            SET UP THE DIAGNOSTICS...
01064  0074 00 4 01105   10001  HGOOF    TSX    WGOOF,4            ...
01065  0074 00 4 01105   10001  DGOOF1   TSX    WGOOF,4            ...
01066  0074 00 4 01105   10001  DGOOF2   TSX    WGOOF,4            ...
01067  0074 00 4 01105   10001  DGOOF3   TSX    WGOOF,4            ...
01070  0074 00 4 01105   10001  DGOOF4   TSX    WGOOF,4            ...
01071  0074 00 4 01105   10001  DGOOF5   TSX    WGOOF,4            ...
01072  0074 00 4 01105   10001  EGOOF    TSX    WGOOF,4            ...
01073  0074 00 4 01105   10001  SGOOF    TSX    WGOOF,4            ...
01074  016060606060      10000           BCI    1,1                ILLEGAL C.C. 1 OP. CHARACTER.
01075  026060606060      10000           BCI    1,2                ILLEGAL C.C. 2 CHAR. IN H-CARD.
01076  246060606060      10000           BCI    1,D                ILLEGAL CHAR. IN DEC. DATA FIELD.
01077  226060606060      10000           BCI    1,B                B PLACE PART TROUBLE.
01100  256060606060      10000           BCI    1,E                DEC. EXP. TROUBLE.
01101  476060606060      10000           BCI    1,P                PRINCIPAL PART TROUBLE.
01102  252260606060      10000           BCI    1,EB               +,- IN EXP. OR B PLACE PART.
01103  106060606060      10000           BCI    1,8                OCTAL CONV. TROUBLE.
01104  624546606060      10000           BCI    1,SNC              SKIP NO. TROUBLE.
01105  0500 00 4 00011   10000  WGOOF    CLA    9,4                ...
BINARY CARD ID. LOAD0032
01106  0601 00 0 00100   10001           STO    DATA               ...
01107  0500 00 0 15000   10011           CLA    .UN06.             SET UP THE OUTPUT FILE...
01110  0621 00 0 01003   10011           STA    *+3                ...
01111  0621 00 0 01004   10011           STA    *+4                ...
01112  0074 00 4 11000   10011           TSX    .OPEN,4            OPEN AS REQUIRED...
01113  5 00000 0 00000   10000           MON    **                 ...
01114  0074 00 4 16000   10011           TSX    .WRITE,4           OUTPUT DIAGNOSTIC AND CARD...
01115  0 01003 0 00000   11100           PZE    **,,*+3            ...
01116  4 00005 0 01035   10001           IOCP   TALE,,5            ...
01117  0 00017 0 00100   10001           IOCD   DATA,,15           ...
01120  0500 00 0 01140   10001           CLA    =1                 SET ERROR INDIC.
01121  0601 00 0 00117   10001           STO    TEST               ...
01122  0601 00 0 00000   10000  ELOC     STO    **                 ...
01123  7 00000 0 00035   10001           TXL    RDD,,              NEXT CARD.
```

STORAGE SECT.
STORAGE FOR D-PRECISION WORDS.

```
        01124  300000000004     00001          EVEN
        01124  200000000002     00001  UNO     BSS    2
        01126  200000000002     00001  DENO    BSS    2
        01130  200000000002     00001  DBNO    BSS    2
        01132  000000000000     10000  DZERO   DEC    0.0EE0

BINARY CARD ID. LOAD0033
        01133  000000000000     10000
        01134  200000000001     00001  TMP1    BSS    1
        01135  200000000001     00001  TMP2    BSS    1
                        01131          BNO     EQU    DBNO+1
                        01127          ENO     EQU    DENO+1
                        01124          ITEM    EQU    UNO
        01136  000000000000     10000          *LDIR
        01137  434621246060     10000
        01140  000000000001     10000          *LORG
        01141  000000000135     10000
        01142  000000000033     10000
        01143  400000000010     10000
        01144  000000000010     10000
        01145  000000023420     10000
        01146  377000000000     10000
        01147  000777777777     10000
        01150  233000000000     10000
        01151  276000000000     10000
        01152  000554000000     10000
        01153  400000000001     10000
        01154  000000000000     10000
                        90000  01111          END
```

CONTROL DICTIONARY

```
          $CDICT LOAD                                                    LOAD0034

BINARY CARD ID. LOAD0035
          001155000000          PREFACE   START=0,LENGTH=621,TYPE=7094,CMPLX=5
          000004000005
          434621246060  LOAD    DECK      LOC=0,LENGTH=621
          001155000000
          434621246060  LOAD    REAL      LOC=0,LENGTH=0
          000000000000
          434621246060  LOAD    REAL      LOC=0,LENGTH=0
          000000000000
          000000000000          EVEN      LOC=1124
          100000001124
          332567316333  .EXIT.  VIRTUAL   SECT. 5,CALL
          200000100000
          472464444760  PDUMP   VIRTUAL   SECT. 6,CALL
          200000100000
          627062434623  SYSLOC  VIRTUAL   SECT. 7
          200000000000
          336445000533  .UNO5.  VIRTUAL   SECT. 8
          200000000000
          334647254560  .OPEN   VIRTUAL   SECT. 9
          200000000000
          335125212460  .READ   VIRTUAL   SECT. 10
          200000000000

BINARY CARD ID. LOAD0036
          332626476333  .FFPT.  VIRTUAL   SECT. 11
          200000000000
          240125006060  D1E0    VIRTUAL   SECT. 12
          200000000000
          336445000633  .UNO6.  VIRTUAL   SECT. 13
          200000000000
          336651316325  .WRITE  VIRTUAL   SECT. 14
          200000000000

          $DKEND LOAD                                                    LOAD0037

NO MESSAGES FOR THIS ASSEMBLY
```

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	BLK	00171	717
	BNO	01131	322,324
	CARD	00101	40,42,43,557,660,1057,1060
	CAS1	00050	53
	CAS2	00163	166
	CAS3	00603	606
	CGCOF	01063	51,54
	CHEK	00520	533
	CMPT1	00741	746
	CMPT2	00763	752
	CMPT	00733	216,223,227,707
	COP	00074	52
	C	00064	50
	D9	00211	165,167
	DATA	00100	117,1106,1117
	DBLK	00457	202
	DBNO	01130	151,217,1136
	DB	00410	207
	DCMA	00461	127,201,441
	DCNT	00241	153,232,234,313,474
	DD	00414	206,420
	DE1	00421	416,417
	DE2	00426	413
	DENO	01126	150,224,1136
	DE	00417	130,205,424
	DGCOF1	01065	164
	DGCOF2	01066	220,363
	DGCOF3	01067	225,525,527
	DGCOF4	01070	231,246,510
	DGCOF5	01071	434,446
	DMIN1	00447	444
	DMIN2	00451	445
	DMIN3	00455	452
	DMIN	00443	203
	DOP	00124	73
	DPL1	00435	433
	DPL	00431	210
	DPT1	00441	430
	DPT	00437	204
	DSW	00155	125,160
	DW	00157	155
	DX	00541	553
	DY	00542	563
	DZERO	01132	146
	DZ	00551	156,221,226,235,241,432,436,442,450,454,456,576,630,633,647
	DZZ	00126	516
	E7	00617	605,607
	EBLK	00634	615
	ECMA	00636	614
	EGCOF	01072	604,620,623
	EINV	00631	627

SYMBOL REFERENCE DATA

ELOC	01122	16,21,1052
EMIN	00645	616
END	01127	257,272
EOP	00564	72
ESET	00566	643
ESW1	00620	567,632
ESW2	00637	571,646
ESW3	00643	572,635
ESW	00575	565,600
EWD	01135	573,621,626,636
EW	00577	575
FCNT	00535	152,236,240,255
FINV	00306	260,273
FIXS	00466	310,351,365,375,407
FLOAT	00251	462
FLT	00766	251
HI0	00671	667,670
HGOOF	01064	653
HOP	00650	71
INO	00242	461
INV	00513	475
ITEM	01124	472,502,504,506,511,641,663,1007
..0001	00003	12,13,14
..0002	00005	4,7
..0003	00007	0
LGOOF	01022	20,25
LOAD	00000	77
LW1	00533	252
LW	00123	75
M3	00201	124,126,157,212,306,311,410,414,421,423,431,437,457,463,513,564,566,570,577,631,634,645,672,674,1013
MASK1	00122	41
MASK	00121	17,24
MQ	00120	47,543,546,560,706,715
NINER	00656	654,671
NINE	00200	163,652,702
PASS	01047	37
PDMP	01054	37
PFIX	00463	250
POP	00672	70
PPART	00227	222
QQ1	00261	274
QQ2	00264	271
QQ3	00270	264
QQ4	00272	256
QQ5	00300	304
QQ6	00303	300
QQ7	00305	265,267,302
ROD	00035	66,74,517,644,666,732,1123
REL	00515	473,477,505,512
RET	00075	67,1050,1053
RND	00506	501
RSHFT	00366	326

LCTR BLCTR
QUAL UNCS
LCTR //

SYMBOL REFERENCE DATA

S1	00700	716
SBLK	00717	703
SET	00536	161,601
SGOOF	01073	711,714,720,722
SHFT1	00335	332,347
SHFT2	00340	333
SHFT3	00362	353
SHFT4	00373	371
SHFT5	00376	367
SHFT6	00404	377
SLOC	01016	23,1010,1011
SMQ	00247	243
SNO	01135	677,712,723,725
SOP	00674	65
STOR	01004	515,642,664
SUB	00352	331
SVN	00613	603
SW100	00432	142,426
SW101	00433	143,213,451
SW130	00444	144,427
SW131	00445	145,214,435
SW1	00215	131,411,422,443
SW200	01010	312,415,465,476,514,1014
SW2	00222	132,215,425
SW3	00235	133,440
SW50	00467	137,447
SW51	00516	140,460
SW55	00473	141,307,464
SW6	00256	134,455
SW7	00310	135,412,500
SW8	00321	136,453
SWA	00724	675
SX1	01020	1015
TALE	01035	1031,1116
TEMP	00534	211,617,625,705,747
TENNER	00667	655
TEST	00117	26,1004,1047,1121
THERE	00532	523
TMP1	01134	520,531,676,710,735,741,743,744,745,756,760,763
TMP2	01135	530,650,733,750,762
UNO	01124	147,230,242,247,263,277,305,315,336,341,350,355,364,374,402,405,406,466,766,770,771,775, 1000,1001,1002,1136
WGOOF	01105	1063,1064,1065,1066,1067,1070,1071,1072,1073
XIT	01032	1062
XR2	00203	541,552,555
XRA	00065	27,727,731,1006,1020
XRB	00202	537,556,562
XRC	00204	547,551
X	01001	772

REFERENCES TO VIRTUAL SYMBOLS.

DIEO	12	266,270,301,303
.EXIT.	5	1032
.FFPT.	11	123

SYMBOL REFERENCE DATA

.OPEN	9	33,1025,1112
.READ	10	35
.UNO5.	8	30
.UNO6.	13	1022,1107
.WRITE	14	1027,1114
PDUMP	6	1054
SYSLOC	7	10

```
                                        SIMULATION ROUTINE.
                                    -SIMULATES A PART OF DECK FCNV.
                                    -TO BE USED TO SCALE D-PRECISION DECIMAL DATA.

                                                ENTRY  DIEO

BINARY CARD (NOT PUNCHED)
      00000  300000000002     00001            EVEN
      00000  377454732312     10000  DE38      DEC    1.0EE+38,1.0EE+37,1.0EE+36,1.0EE+35,1.0EE+34,1.0EE+33
      00001  344413241542     10000
      00002  373741367020     10000
      00003  340653551067     10000
      00004  370601137163     10000
      00005  335674440705     10000
      00006  365464114134     10000
      00007  332543515404     10000
      00010  361755023372     10000
      00011  326554174007     10000
      00012  356612334310     10000
      00013  323443311471     10000
      00014  353473426555     10000            DEC    1.0EE+32,1.0EE+31,1.0EE+30,1.0EE+29,1.0EE+28,1.0EE+27
      00015  320202556055     10000
      00016  347770675742     10000
      00017  314004260111     10000
      00020  344623713116     10000
      00021  311320214724     10000

BINARY CARD (NOT PUNCHED)
      00022  341503074076     10000
      00023  306563327103     10000
      00024  336402374713     10000
      00025  303617422402     10000
      00026  332635456171     10000
      00027  277177204004     10000
      00030  327512676455     10000            DEC    1.0EE+26,1.0EE+25,1.0EE+24,1.0EE+23,1.0EE+22,1.0EE+21
      00031  274631003151     10000
      00032  324410545213     10000
      00033  271024002441     10000
      00034  320647410336     10000
      00035  265354635550     10000
      00036  315522640261     10000
      00037  262760512755     10000
      00040  312417031701     10000
      00041  257446725444     10000
      00042  306661534465     10000
      00043  253561357240     10000
      00044  303532743536     10000            DEC    1.0EE+20,1.0EE+19,1.0EE+18,1.0EE+17,1.0EE+16,1.0EE+15

BINARY CARD (NOT PUNCHED)
      00045  250132614200     10000
      00046  300425434430     10000
      00047  245110475000     10000
      00050  274674055531     10000
```

ASSEMBLED TEXT.

```
00051  241647310000     10000
00052  271543212741     10000
00053  236354240000     10000
00054  266434157115     10000
00055  233760200000     10000
00056  262706576511     10000
00057  227432000000     10000
00060  257553630407     10000          DEC     1.0EE+14,1.0EE+13,1.0EE+12,1.0EE+11,1.0EE+10,1.0EE+9
00061  224510000000     10000
00062  254443023471     10000
00063  221240000000     10000
00064  250721522450     10000
00065  215400000000     10000
00066  245564416672     10000
00067  212000000000     10000

BINARY CARD (NOT PUNCHED)
00070  242452013710     10000
00071  207000000000     10000
00072  236734654500     10000
00073  203000000000     10000
00074  233575360400     10000          DEC     1.0EE+8,1.0EE+7,1.0EE+6,1.0EE+5,1.0EE+4,1.0EE+3,1.0EE+2
00075  200000000000     10000
00076  230461132000     10000
00077  175000000000     10000
00100  224750220000     10000
00101  171000000000     10000
00102  221606500000     10000
00103  166000000000     10000
00104  216470400000     10000
00105  163000000000     10000
00106  212764000000     10000
00107  157000000000     10000
00110  207620000000     10000
00111  154000000000     10000
00112  204500000000     10000          DEC     1.0EE+1

BINARY CARD (NOT PUNCHED)
00113  151000000000     10000
00114  201400000000     10000  01E0    DEC     1.0EE+0
00115  146000000000     10000
                 00000  01111          END
```

CONTROL DICTIONARY

```
          $CDICT TAB10                                                    TAB10002


BINARY CARD (NOT PUNCHED)
          000116000000              PREFACE   START=0,LENGTH=78,TYPE=7094,CMPLX=5
          000004000005
          632122010060    TAB10     DECK      LOC=0,LENGTH=78
          000116000000
          C00000000000              EVEN      LOC=0
          100000000000
          240125006060    D1E0      REAL      LOC=114,LENGTH=0
          C00000000114

          $DKEND TAB10                                                    TAB10003


NO MESSAGES FOR THIS ASSEMBLY
```

SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

CLASS	SYMBOL	VALUE	REFERENCES
	D1E0	00114	
	DE38	00000	
LCTR	BLCTR		
QUAL	UNQS		
LCTR	//		

Appendix C

Output from the Test Program and Subroutine LOAD

000000000000	000000000001	000000000012	000000000123	000000001234	000000012345	000000123456	000001234567
000000000000	400000000000	400000000001	400000000012	377777777777	777777777777	777777777777	777777777777
000000000010	400000000010	000000010000	000000200000	000004000000	000100000000	002000000000	040000000000
000000000177	000000003777	000000077777	000001777777	000037777777	000777777777	017777777777	377777777777
176400000000	604400000000	207400000000	215400000000	221400000000	225400000000	231400000000	235400000000
209700000000	203700000000	207774000000	213777600000	217777770000	223777777400	227777777760	233777777777
002663437347	172400000000	175631463146	211500000000	214500000000	224750220000	341503074077	377454732313
100000000000	020000000000	003000000000	000400000000	000050000000	000006000000	000000700000	000000100000
100000000000	020000000000	003000000000	000400000000	000050000000	000006000000	000000700000	000000100000
100000000000	020000000000	003000000000	000400000000	000050000000	000006000000	000000700000	000000100000
100000000000	020000000000	003000000000	000400000000	000050000000	000006000000	000000700000	000000100000
000001400000	000001700000	000001770000	000000001750	000000000002	000000000000	031463146315	200000000000
176400000000	143000000000	604400000000	551000000000	207400000000	154000000000	235400000000	202000000000
172400000000	137000000000	203700000000	150000000000	217777770000	164000000000	233777777777	200000000000
002663437347	000000000000	040625513677	005523263217	341503074076	306563327102	377454732312	344413241542
245400000000	212000000000	246400000000	213000000000	247400000000	214000000000	250400000000	215000000000
200777400000	145000000000	200777600000	145000000000	200777700000	145000000000	200777740000	145000000000
000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
000102030405	060710112122	232425262730	314142434445	464750516263	646566677071	137353331474	543420004060
606060606060	606063303162	603162606330	256025452460	462660652143	312460242163	213360606060	606060606060
500000000000	500000000000	500000000000	500000000000	500000000000	500000000000	500000000000	500000000000

```
1     *
D     D*
P     D34359738368
B     D1B0
B     D256B8
E     D1E39
E     D1E-39
E     D1D39
E     D1D-39
P     D9.99999999999999999999999999999
B     D6.4E2B6
EB    D1.E-+3
EB    D1.E+-3
EB    D1B+-17
EB    D1B-+17
8     E8
8     E1234567654321
2     H*
SNO   P*
SNO   P10000
*** LOAD CALLING ERROR ***
```

Appendix D

Input Data Deck Listing

```
$DATA

                    ...TEST DATA FOR SUBROUTINE LOAD...

 NOTE.  A BLANK IN C.C.1 IS EQUIV. TO A C IN C.C.1.
 NOTE.  IN THE D.P. DATA BELOW, A -D- MAY BE REPLACED BY A -EE-.
 NOTE.  THE GROUPS OF DATA ARE SEPARATED BY EOF-S.
 NOTE.  AN EOF IS NOT EXACTLY EQUIV. TO AN -R- CARD.
 NOTE.  THE REACTION OPTION TO AN EOF RESIDES IN THE CALLING ROUTINE.
 NOTE.  SKIPPING OVER PREVIOUSLY LOADED DATA IS PROGRAMMED.


D168

                    ...OCTAL DATA...

E0,1,12,123,1234,12345,123456,1234567
E,-,-1,-12,377777777777,-377777777777,777777777777,-777777777777
-EOF.
P17   SKIP.

                    ...INTEGER DATA...

D8,-8,4096,65536,1048576,16777216,268435456,4294967296
D127,2047,32767,524287,8388607,134217727,2147483647,34359738367
-EOF.
P33   SKIP.

                    ...S.P. FLOATING PT. DATA...

D0.125,-8.,64.,4096.,65536.,1048576.,16777216.,268435456.
D0.875,7.,127.,2047.,32767.,524287.,8388607.,134217727.
D1E-38,7.8125E-3,1E-1,3.2E2,2.56E3,1E6,1E29,1E38
-EOF.
P57   SKIP.

                    ...S.P. FIXED PT. DATA...

D1B2,2B5,3B8,4B11,5B14,6B17,7B20,8B23
D64B8,128B11,192B14,256B17,320B20,384B23,448B26,512B29
D6.4E1B8,1.28E2B11,1.92E2B14,2.56E2B17,3.2E2B20,3.84E2B23,4.48E2B26,5.12E2B29
D6.4B8E1,1.28B11E2,1.92B14E2,2.56B17E2,3.2B20E2,3.84B23E2,4.48B26E2,5.12B29F
D1.5B17,1.875B17,1.984375B17,1E3B35,1E3B44,1E3B52,.1B0,1.25E-1B-2
-EOF.
P97   SKIP.

                    ...D.P. FLOATING PT. DATA...

D1.25D-1,-8D0,6.4D1.268435456D0
D7.8125D-3,7D0,3.2767D4,134217727D0
D1D-38,1D-29,1D29,1D38
D68719476736.,137438953472.,274877906944.,549755813888.
D0.9990234375,0.99951171875,0.999755859375,0.9998779296875
-EOF.
P137  SKIP.

                    ...ZEROES...

E0,
D0,0.0,D.,
```

```
C
C              ...BCI WORD DATA...
C
H8     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ=,$.-(*)+0-
H8              THIS IS THE END OF VALID DATA.
R

                    ...GENERATE A SAMPLE LIST OF DIAGNOSTICS...
                         ...THE DATA BELOW IS INVALID...

*
D*
D34359738368
D1B0
D256B8
D1E39
D1E-39
D1D39
D1D-39
D9.99999999999999999999999999999
D6.4E2B6
D1.E-+3
D1.E+-3
D1B+-17
D1B-+17
E8
E1234567654321
H*
P*
P10000
-EOF.
```

92 CARDS

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546