N.I.O. COMPUTER PROGRAMS 12

edited
by
B. J. Hinde

N.I.O. Internal Report No. N.12

National Institute of Oceanography
<table>
<thead>
<tr>
<th>No.</th>
<th>Program Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>Paper tape duplication</td>
</tr>
<tr>
<td>111</td>
<td>Current Meter Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SUB-PROGRAMS</td>
</tr>
<tr>
<td>6</td>
<td>Tape Security System: Find a file</td>
</tr>
<tr>
<td>7</td>
<td>Bit setting routine</td>
</tr>
<tr>
<td>8</td>
<td>Three-point lineprinter plotting</td>
</tr>
<tr>
<td>9</td>
<td>Variable position integer printing</td>
</tr>
<tr>
<td>10</td>
<td>EBCDIC to ECDIC conversion table</td>
</tr>
<tr>
<td>11</td>
<td>Current Meter Conversion</td>
</tr>
<tr>
<td>12</td>
<td>1442 Stacker select</td>
</tr>
<tr>
<td>13</td>
<td>EBCDIC to EBCDI conversion table</td>
</tr>
<tr>
<td>14</td>
<td>Arctangent</td>
</tr>
</tbody>
</table>
Title

Paper tape duplication

Name

PADUP

Machine

1830

Language

1830 Assembler

Purpose

To duplicate 5, 6, 7 or 8 track paper tape.

Input

// JOB
// * Project No./Name/Job Title
// XE PADUP
//CED

Operation

Place the paper tape to be duplicated in the reader
with either runout or the first required character
over the heading head. Set the punch track selector
and tape guides as required. Load in the job cards
(or tape) in the usual way and after the program is
brought into core from disk, duplication will begin
and continue to the physical end of the tape. At any
time during execution, duplication may be stopped by
pressing console STOP and restarted using console
START without loss of characters. However, if it is
required to edit two tapes together (for example) this
method should not be used since, on restarting, the
characters (up to 4) remaining in the buffer areas
will be punched first. Instead, stop duplication by
setting data switch 14 ON. Execution will cease with
all characters that have been read correctly punched.
To restart, set data switch 14 OFF and press console
START.

Method

The program uses two buffer areas in order to achieve
maximum punching speed. The data switches are sensed
after every four characters.

Execution Speed

Slightly less than 150 characters/second.

Programmer

Brian Hinde
Title: Current Meter Analysis

Name: CRANO

Language: 16° Fortran IV

Purpose: To read, calibrate and plot data from Bergen or Plessey current meters. Data is read from cards, two complete sets of readings (e.g. reading, number, current speed, direction and temperature) being punched on each card. There may be up to six readings per set and any one of the readings may be designated as rotor count, the difference between consecutive readings then being used by the program as the basis of current speed. The first reading in each set is normally the reading number, and calibrations of the form \( y = ax + b \) are applied to all the other readings. Provision is made for a dead space on the speed rotor and up to three graphs may be plotted as output (e.g. speed, direction and temperature). All results are also output to magnetic tape.

Inputs: Job and data

Job Description:

// JOB
// *Job No./Name/Title
// XEO CRANO

*PLCES(24,MK,C) where K is the magnetic tape number to be used.

*LOCAL*(NEXT*,STOR*,PIND*), INEG

Data:
The data is in two sections, immediately following *CCEHD:

Part a), seven cards, is for program control.
Part b) is raw data.

a) CARD 1 Normally a blank card. (For exceptions see operators instructions).

CARD 2   0Cm 22  30  36  46  54
A(1)bTObA(6)bX.XXXXYbY.YYYYYbV.WVVVVbW.WWWWbQ.QQPPPbF.PPPP
X.XXXXX a number representing calibration constant A(1)
(See method)
Y.YYYYY similarly for A(2)

P.PPPP A(6)

CARD 3   0Cm 22  30  36  46  54
B(1)bTObB(6)bX.XXXXYbY.YYYYYbV.WVVVVbW.WWWWbQ.QQPPPbF.PPPP
identical to card 2, except that the numbers represent calibration constants B(1 to 6)
Program 111 p.2.

**CARD 4**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>XXXX.XX a positive number, representing the minimum expected value of the first calibrated result to be plotted.</td>
</tr>
<tr>
<td>7-12</td>
<td>YYYY.YY a positive number, of two decimal digits, representing the maximum expected value of the first calibrated result to be plotted.</td>
</tr>
<tr>
<td>13-18</td>
<td>VVVV.VW similar to XXXX.XX but for the second value to be plotted</td>
</tr>
<tr>
<td>19-24</td>
<td>VVVV.VW identical to YYYY.YY but for the second value to be plotted</td>
</tr>
<tr>
<td>25-30</td>
<td>min. for the third value to be plotted</td>
</tr>
<tr>
<td>31-36</td>
<td>max. for the third value to be plotted</td>
</tr>
</tbody>
</table>

**N.B.** When no graph is required, dummy values of 0 min. and 100 max. should be used.

**CARD 5**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-44</td>
<td>Similar to XXXX.XX but for the second value to be plotted</td>
</tr>
<tr>
<td>45-84</td>
<td>Similar to YYYY.YY but for the second value to be plotted</td>
</tr>
</tbody>
</table>

**CARD 6**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>XXXX.XX raw data card columns 1 to 6 or 41 to 46</td>
</tr>
<tr>
<td>7-12</td>
<td>YYYY.YY signifies the direction of meter rotation. If clockwise the EBCDIC character code for + (24640) should be used. If anti-clockwise, - (24632).</td>
</tr>
<tr>
<td>13-18</td>
<td>ZZZZZZ This is the last meter reading of the set of data. It will be found, on the last raw data card, in columns 1 to 6 or 41 to 46, depending on whether the last data card contains one or two sets of data (see section b)</td>
</tr>
<tr>
<td>19-24</td>
<td>max. for the third value to be plotted</td>
</tr>
</tbody>
</table>

**Series Number**

<table>
<thead>
<tr>
<th>XXXXX</th>
<th>raw data card columns 1 to 6 or 41 to 46</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYYY</td>
<td>1 to 12 or 47 to 52</td>
</tr>
<tr>
<td>VVVV</td>
<td>13 to 18 or 53 to 58</td>
</tr>
<tr>
<td>MMMM</td>
<td>19 to 24 or 59 to 64</td>
</tr>
<tr>
<td>PPPP</td>
<td>25 to 30 or 65 to 70</td>
</tr>
<tr>
<td>5</td>
<td>31 to 36 or 71 to 76</td>
</tr>
</tbody>
</table>

**CARD 7**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-15</td>
<td>XX a two digit number signifying the meters dead space.</td>
</tr>
<tr>
<td>16-33</td>
<td>Y the number of graphs required from 0 to 3.</td>
</tr>
<tr>
<td>34-41</td>
<td>Z may take values 1, 2, ..., 5. These are the series numbers as described in b). Any such series specified will have the similarly referenced value from the previous set of raw data subtracted from it before its value is analysed.</td>
</tr>
</tbody>
</table>
Program 111 p.3.

$W, 0, P$ may take values 3, 1, 2, ..., 6. They may only be identical for 0's. Any series number used will give a plotted position for that column of analysed data. The lowest of $W, 0$ or $P$ will be given the first two ranges as denoted on CARD 4. The second lowest, the next two etc.... The 0 is used when a graph is not required.

1) The number of columns of raw data per set (from 0 to 6). See data section b).

$R, S, T,$ the series numbers of columns without data on the $U, V, W$ data on the raw data card. (Range 1 to 6)

b) The raw data cards follow CARD 7 immediately:- Each card may contain two sets of data the first in cols. 1 to 36 and the second in 41 to 76. Each set may contain a maximum of 6 digit numbers in the serial number positions 1 to 6 as described under CARD 5. Any of the cols. can be blank.

EXAMPLE The following is a typical raw data card

```
CC 1 67 123 149 325 321 36 40
bb2k73bb32b6bb1b5bb555555bbbbb5bb - SET 1
CC 41 57 55 57 625 721 76 82
bb2k73bb21b555b1b55bb555555bbbbb5bb - SET 2
1 2 3 4 5 6
```

Serial Direct-Speed Temperature
No.

The first seven cards would then be as follows, assuming beginning of a run and that

a) serial nos. 2 and 5 were wanted for plotting,

b) that serial no. 3 was current speed,

c) the meter number was 999,

d) the meter rotation was clockwise,

e) the number in cc41 to 45 on the last data card was 8888,

f) the meter deadspace was 77,

g) the expected range of serial no. 2 when calibrated was between 0 and 50 and serial 5 was 11 and 22,

h) calibration constants all of $A(2$ to $6) = 2.0$

```
(see output to lineprinter)
```

CARD 1. blank

CARD 2. $A(1)bTOhA(6)b1, bbb0b2, bbb0b2, bbb0b2, bbb0b2, bbb0b2, bbb0b2

CARD 3. $B(1)bTOhB(6)b1, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5

CARD 4. $A(2)bTOhA(6)b1, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5

CARD 5. $A(3)bTOhA(6)b1, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5

CARD 6. $A(4)bTOhA(6)b1, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5

CARD 7. $A(5)bTOhA(6)b1, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5, bbb0b5

OUTPUT

The page will be headed with the position where the results will be stored on magnetic tape. Then, the meter number will be printed. Following this, each set of calibrated results will be printed on the left of the continuous stationery, the first value being treated as the meter number and printed as an integer. (Hence, $A(1)$ and $B(1)$
(CARDS 2 AND 3) should be 1 and 0 respectively. The results will be printed in ascending order of serial number (L to R) - for serial numbers see CARD 5.

Immediately following each set of results will be a graph of up to three of the variables utilising the maximum width of paper left. This graph will be headed with symbols and their ranges. At the foot of the graph the max. and min. of each range will be recorded together with their means and standard deviations. These results are based on actual values while points exceeding the ranges will be plotted on the upper or lower limits of the graph.

The output is terminated with the next available storage location on magnetic tape.

Magnetic Tape

The values as written on the lineprinter will be recorded on magnetic tape, their locations being noted on the lineprinter. The format is as follows:

Each file is headed with the meter number in I7 format. Every set of readings after this is in I7, 1X, 5(F7.2, 1%) format until the run is terminated by an end-of-file marker being written.

Operator Instructions

Due to the fact that large amounts of information are handled, a facility is included to abort a run in an orderly fashion, by switching on data switch 14. The lineprinter graph will be finished and the magnetic file will remain open (i.e. no end-of-file marker).

To re-start the run, the first data card must be punched with

X3I(7)bX\ldots

XXXX being found at the foot of the graph on the terminated run. The message on the lineprinter being,

RGXXXXXX.

Errors

Should any errors occur they will be flagged and explained.

Execution Time

Approx. 4 minutes per 10 cards (200 sets).

Method

The calibrations are performed with the equation

\[ Y = Ax + B \]

where A and B are calibration constants and \( x \) is the raw data value [see N.I.O. subroutine -11].

Programmer

W. T. J. Slade.
Title      Tape Security System: Find a file

Name       Subroutine FILEC

Machine    IBM 1800

Language   1800 Fortran IV

Purpose    An execution of CALL FILEC (N,L) will position the
            magnetic tape in use so that a READ or WRITE
            statement following will access the first record of
            file N.  L is the record length in words and should
            be set to 74 for formatted Fortran I/O tapes and to
            146 for unformatted tapes written on the IBM 1800.
            The *IOCS record should contain MAGNETIC TAPE,
            1443 PRINTER.

Modifications The version of FILEC stored on disk is set up for
            reading odd parity tapes, 3 bytes/word, 800 b.p.i.
            density.  By changing card FILEC35 as follows
            other combinations may be used:

            DATA MODE/35/  
            Always O          Always O
            O for odd parity
            1 for even parity
            O for 5 bytes/word, 800 bpi
            1 for 3 bytes/word, 800 bpi
            2 for 3 bytes/word, 556 bpi
            4 for 2 bytes/word, 800 bpi
            5 for 2 bytes/word, 200 bpi
            6 for 2 bytes/word, 556 bpi

Restrictions  L < 146
              N < 99

Errors      Error Messages are printed as follows:
            'WRONG LENGTH RECORD IN FILE NUMBER n'
            ' UNCORRECTABLE TAPE ERROR IN FILE NUMBER n'
            'READ CHECKS HAVE OCCURRED IN FILE NUMBER n'

Programmer B. J. Hinde
Title: Bit setting routine

Name: Subroutine MSKDO (LIST (1), N, MASK)

Machine: 1800

Language: 1800 Assembler

Purpose: To set a desired bit pattern in an 1800 word.

The result of a call to the subroutine is to place in MASK a word containing bits in positions listed in the first N elements of LIST. Bits already set in MASK are cleared.

E.g. if LIST (1) = 14, LIST (2) = 0, LIST (3) = 3, LIST (4) = 10 then

DIMENSION LIST (16)
CALL MSKDO (LIST, 4, MASK)

results in MASK having 1000 0000 0010 0010
(i.e. bits 14, 0, 3, 10 have been turned on)

while CALL MSKDO (LIST, 2, MASK)

results in 1000 0000 0000 0000
(i.e. only the first two elements of LIST have been used.)

Inputs: LIST (1), N

Output: MASK

Restrictions: N < 16, LIST (1) < 15

Programmer: J. Crease
Title: Three-point lineprinter plotting

Name: Subroutine POTD\^ (X, Y, Z MORE)

Language: Fortran

Machine: 187C

Purpose: To compute and plot the positions, on the same x-axis, of three or less points using the lineprinter. Additional facilities are the recording of maximum and minimum and the calculation of mean and standard deviation for each set of points.

Data: Data is entered by the parameters X, Y, Z and MORE and also with the use of a "COMMON" area.

a) Arguments

X, Y and Z are the values to be plotted. They must be positive, real and accurate only to six sig. figs. The substitution of negative values in X, Y and (or) Z will inhibit the printing of the respective points until finally, with three negative values, only the axis marker (I) will be present.

MORE may take four values only 1, 2, 3 or 4.

Plain graph (no headings or means etc.)

"MORE" should be given the value 4 for every "call".

Headings and graph only

For the first "call" "MORE" must be 1 and all other arguments should take dummy values. No positions will be plotted. Subsequently the value 4 should be used.

Graph, headings and means etc.

For the first "call" "MORE" should be 1 and all other arguments should take dummy values. No positions will be plotted and there will be no effect on the means etc. For every other "call" except the last it must be 2.

For the final "call" "MORE" should be 3.

b) Common

Rng (6), Orgin

Through these 7 variables are entered the following:-

Rng (1) and rng (2) represent respectively, the lower and upper ranges of x. Similarly rng (3) and rng (4) for y and rng (5) and rng (6) for z. All x, y and z values outside these ranges (excluding negatives) will be placed on the maximum or minimum for plotting (only). The variable ORGIN gives the character position at which the origin is to be. It can vary from 1 to 119.

Output

The positions are plotted by the characters +, *, 0 for X, Y and Z respectively. Should any points be unipositional then the + will be positioned correctly while the others will be stepped to the right. The reverse will occur only on the lineprinter 12\^th position. The axis is denoted by the character I. A new line must be called by the user after POTD\^ has been used unless over-printing is required.
b) Headings
In every case, when headings are requested, an origin marker followed by a line of minus signs will be printed. All ranges, for headings only, are truncated to the nearest integer, therefore the heading facility should not be used if rng (1) to rng (6) are non-integer or exceed 32767. They will be shown in the headings together with the representative symbols for X, Y and Z. This is true only if orgin has a value less than 100.

c) Means etc.
These results will be printed the full width of the page. The values used to compute them, omitting negatives, will be the actual values presented and not the corrected values used for plotting.

Error Messages
There are many different error messages - all self explanatory. For example if ORGIN was -3 the warning message ORGIN TOO SMALL -3 would be printed. It should be noted that as this a subroutine, control will then be transferred to the mainline program and the execution not halted. To allow for this, at every error MORE is put equal to 100 and hence a test in the mainline program can stop the output of many unnecessary error messages.

Execution Time
The time for each line of plotted points to be processed and printed is approximately (0.8 second when full facilities are in use).

Programmer
W. T. J. Slade
Title
Variable position integer printing

Name
Subroutine DNRCB (INT, IELEM, LIST)

Machine
IBM 1600

Language
1600 Fortran IV

Purpose
To convert the integer INT to 6A1 format suitable for printing and to position it in the integer array LIST starting at position IELEM.

Data - Inputs
Data is entered through the arguments "INT", "IELEM" and "LIST". "INT" is the integer requiring variable position output, limited only by 187 restrictions, i.e. $-32767 \leq INT \leq 32767$. It must be of the form x000xx, leading blanks being suppressed. "IELEM" specifies the position at which the first character of "INT" is to be placed. N.B. it may be a blank. Thus, the limitations are $-4 \leq IELEM \leq 115$. "LIST" is a 120 element array which will only be overwritten in the non-blank positions of "INT".

Output
Both "INT" and "IELEM" will retain their original values. The results will be given in array "LIST". Following is an example. If the complementary "LIST" array "ISET" contained the following,

\[
\begin{align*}
\text{ISET}(22) &= N, & \text{ISET}(23) &= O, & \text{ISET}(24) &= \cdot \\
\text{ISET}(25) &= A, & \text{ISET}(26) &= \text{"\text{=}"}, & \text{ISET}(27) &= *, & \text{ISET}(28) &= Z \\
\text{ISET}(29) &= \), & \text{ISET}(30) &= \text{blank} & \text{ISET}(31) &= * \\
\end{align*}
\]

a "CALL DNRCB" (-30, ISET) would result in

\[
\begin{align*}
\text{ISET}(22) &= N, & \text{ISET}(23) &= O, & \text{ISET}(24) &= \cdot \\
\text{ISET}(25) &= A, & \text{ISET}(26) &= \text{"\text{=}"}, & \text{ISET}(27) &= *, & \text{ISET}(28) &= Z \\
\text{ISET}(29) &= \), & \text{ISET}(30) &= \text{blank}, & \text{ISET}(31) &= * \\
\end{align*}
\]

N.B. When printing the results, all positions affected by "DNRCB" should be in A1 format.

Restrictions
This subroutine deals only with integers. If real numbers are supplied to the "INT" argument the results will not be a truncated integer, erroneous numbers will result.

Programmer
W. T. J. Slade
**Title**  
EBCDIC to BCDIC conversion table.

**Name**  
EBCDI

**Machine**  
IBM 1800

**Language**  
1800 Assembler

**Use**  
The table consists of 256 characters - 128 words with two 8 bit characters per word. The seven low-order bits of the character to be converted (input character) are used as an address. The address designates the position in the table of the corresponding conversion character. The high-order bit (bit 0) of the input character designates which half of the table word is to be used. When bit 0 is 1, the left half of the word is used. When bit 0 is 0, the right half of the word is used. All dummy entries (invalid characters) contain the code for space.

To obtain the address of the table entry point

```
CALL  EBCDI
```

The table may be used with 1800, ZIPO.

The table entries are defined in DPG/P/10.

**Programmer**  
B. J. Hinde.
Title
Current Meter Conversion

Name
Subroutine CURNT (XN, X, Y, Z, MORE)

Machine
IBM 1830

Language
1830 FORTRAN IV

Purpose
To convert raw current meter data into calibrated output for recording on both line-printer and magnetic tape. This sub-program is compatible with subroutine POTLO, three-point graph plotting (No. -8).

Data
Common
RNG(6), ORGIN,K,A(6),B(6),N,L(3),D,ISENS

Arguments
XN(a seven element integer array), X,Y,Z,MORE.

Data Entered

a) (For analysis of meter readings). Data is entered through the "COMMON" variables A(6), B(6), D, ISENS, K and by the arguments XN(1) to XN(7).

Arrays A and B contain calibration constants for the corresponding speed, direction etc., given by XN, an integer array (see method). Any XN elements not required should be set to -1000. XN(1) is normally regarded as the reading number and printed as an integer, hence, A(1) and B(1) should take the values 1 and 0 respectively, so as to maintain the original value of XN(1). XN(7) is normally controlled by the program itself and contains the previous XN(K) value. For the first "CALL CURNT" however, XN(7) should be loaded with a value, ≥, unless continuing a previous set of data. Variable D, an integer, should correspond to the meter's dead space. ISENS denotes clockwise and anti-clockwise meter rotation given respectively by 24-640 and 27-32 (the EBDIC character codes for + and -). Finally, K pertains to the element of XN from which successive adjacent readings are subtracted (see method).

b) (For plotting a graph - else all variables in this section may be set to 0). This data is entered through the "COMMON" area, RNG(6), N, L(3) and argument MORE. L(1) to L(3) will set the calibrated values derived from XN(L(1)), XN(L(2)) and XN(L(3)) in X, Y and Z. If one of L(1) is ≤ then Z is set to -100, if two are zero then Z = Y = -100, etc. (for further details see POTLO). Variable N signifies the number of graphs required, to a maximum of 3. RNG(1) and RNG(2) represent the lower and upper ranges of X. Similarly RNG(3) and RNG(4) for Y and RNG(5) and RNG(6) for Z. If MORE is set to value 1 then conditions are produced for POTLO but no results are written to either lineprinter or magnetic tape (for use with POTLO see range of MORE's values in program description -8).
Data is output through the "COMMON" variable ORGIN, an integer, and arguments X, Y and Z. (As instructed by b.) ORGIN is automatically set by CURAT for use in POTIO. X, Y and Z take calibrated meter data given by L(1) to L(3). These results are compatible with POTIO.

The following will be printed on the lineprinter and identically on magnetic tape:

\[ y_N \text{ (see method) will be printed on a new line in I7 format, for the reading number, a space and for all readings, not -100, the results will be printed in (F7.2,1X) format.} \]

There are several different error messages, all self-explanatory. Should any fault occur then the argument NORE will be given the value 100. A test for this should be made after every "CALL" as only the mainline program can terminate the run in good order. This is especially true if POTIO is being used, since such a value entered as an argument would cause a disordered abort.

Warning A "COMMON" declaration must be made in any mainline program calling this subroutine from disk. The layout of this is given under "DATA".

Calibration is applied in the form,

\[ Y_N = A(N) \times X(N) + B(N) \]

One set of raw data, determined by variable K, can have the previous term subtracted as in

\[ Y_K = A(K) \times X(N(K)) - X(N(K-1)) + B(K) \]

or, with the alternate value of ISETS, (27032)

\[ Y_K = A(K) \times X(N(K-1)) - X(N(K))) + B(K) \]

Programmer W. T. J. Slade
<table>
<thead>
<tr>
<th>Title</th>
<th>1442 Stacker Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Subroutine STACK</td>
</tr>
<tr>
<td>Machine</td>
<td>IBM 1800</td>
</tr>
<tr>
<td>Language</td>
<td>1800 Assembler</td>
</tr>
<tr>
<td>Purpose</td>
<td>To stack cards in the second hopper of the 1442 card reader-punch.</td>
</tr>
<tr>
<td>To Use</td>
<td>Use of the instruction CALL STACK.</td>
</tr>
<tr>
<td></td>
<td>after a reading or punching operation will cause the card to be stacked in the second hopper of the 1442. Successive cards will be stacked in the first hopper.</td>
</tr>
<tr>
<td>Programmer</td>
<td>W. Slade from a subroutine written by R. K. Louden.</td>
</tr>
</tbody>
</table>
Title: BCDIC to EBCDI conversion table.

Name: DICEB

Machine: IBM 1800

Language: 1800 Assembler

Use: The table consists of 256 characters - 128 words with two 8 bit characters per word. The seven low-order bits of the character to be converted (input character) are used as an address. The address designates the position in the table of the corresponding conversion character. The high-order bit (bit 0) of the input character designates which half of the table word is to be used. When bit 0 is 1, the left half of the word is used. When bit 0 is 0, the right half of the word is used. All dummy entries (invalid characters) contain the code for space.

To obtain the address of the table entry point

CALL DICEB

The table may be used 1800 ZIPE.

The table entries are defined in DPC/F/10.

Programmer: B. J. Hinde
Title: Arctangent

Name: Function ATAN2 (x, y)

Language: 1800 Fortran IV

Machine: IBM 1800

Purpose: To calculate the arctangent of x/y in the range $-\pi$ to $+\pi$; the quadrant is determined as if x and y are proportional to the sine and cosine of the angle respectively. The function is equivalent to the ATAN2 provided in the Fortran V language.

Inputs: The two real arguments x and y.

Output: The function returns a real result accurate to 9 decimal digits.

e.g. To compute $\tan^{-1} \frac{A}{B}$ and put the result in C:

    
    C = ATAN2 (A, B)

Restrictions and failures: None; whenever both arguments are zero, an answer of zero is returned by the function.

Programmer: Catherine Clayson