N. SMITH

NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

N.I.O. Computer Programs 15

N.I.O. Internal Report No. N.15

APRIL 1970

N.I.O. COMPUTER PROGRAMS 15

GENERAL PURPOSE UTILITY PROGRAMS

N.I.O. Internal Report No. N.15

National Institute of Oceanography

N.I.O. PROGRAMS 15

1) PROGRAMS

138	High Speed Paper Tape Listing	PALIS
139	High Speed Card to Paper Tape Conversion	CAPER
140	Duplicate Paper Tape	PAPER
142	High Speed Card Listing	CALIS
169	High Speed Paper Tape to Cards	PACAR
175	Prepare PTTC/8 Paper Tapes	PREP
176	Convert Binary Tape to Hexadecimal PTTC/8 Tape	BIHEX
177	List Paper Tape	PAPRA
182	Stack Blank Cards	BLAST

2) SUBPROGRAMS

-20	Reverse Data Array	RVERS
-21	Convert BCDIC code to EBCDIC	READM
-22	Paper Tape Free Format Read Routine	UNFMT
-23	Read and Check Paper Tape from Data Loggers	DLOG
-25	Convert EBCDIC to BCDIC	WRITM
-47	Punch ICT Atlas/Orion 7 Track Paper Tape	WRITA
-48	EBCDIC to ICT ATLAS/ORION 7 Track Paper Tape Code Conversion Table	EBCAT
-53	ICL Mercury/Atlas 5 Track Paper Tape Code to EBCDIC Conversion Table	MADIC
-55	Busy Test on Typewriter/Keyboard	KEYBD
-56	Atlas 7 Track Paper Tape Code to EBCDIC Conversion Table	ATLEB
-57	Punching Blank and Delete Codes for P/T Leaders	PLEAD,PLAGG

-58	Read ICL Atlas/Orion/Mercury 5 Track Paper Tape	READ5 (Version 1)
-61	Decoding of BCD number	BCDA
-64	I.S.O. 8 Track Paper Tape Code to EBCDIC conversion table	ISOEB
-65	Read I.SO. Code 8 Track Paper Tape	READ8
-67	Skip to next file	NEXTF

The use of sub-programs -20, -21, -25 and -67 in connection with ½" "compatible" magnetic tape recorded in BCD code is described in the N.I.O.publication "Reading and Writing ½" 7 Track Magnetic Tapes in various BCD Codes on the IBM 1800" in Ref. DPG/P/10.

Queries regarding the use or availability of any of the programs in this volume may be made to:-

The Program Librarian,
Data Processing Group,
National Institute of Oceanography,
Wormley, Godalming, Surrey.

from whom a comprehensive list of all current N.I.O.Programs is available.

ERRATA

Subprogram -23

Title should read "Check Paper Tape from Data Loggers"

Execution Time   The time given includes the associated read statements. These are necessary, but are not part of this sub-program.

Data Processing Group

New Format for Assembler Tapes under TSXPT

From the start of Cruise 29 (1 August 1969) all 1800 Assembler programs punched on paper tape must have the start of the label field (normally column 21) in column 1. This saves having to punch 20 spaces at the beginning of each paper tape record.

The high speed conversion programs (CAPER,N.I.O.139 cards to tape and PACAR, N.I.O. 169 Tape to cards) have been modified so that if Data Switch 0 on the operator's console is ON during execution, all records other than those beginning / or * will be re-formatted. Thus, data switch 0 should be OFF for Fortran programs and data, ON for Assembler Programs and in either position for systems tapes (e.g. // DUP, *DUMPLET, etc). All exisitng Assembler Programs on paper tape will need converting to the new format before they can be used again.

B.J.Hinde
R. Bromley
14th. July 1969

All the programs in this volume have been compiled and executed on an I.B.M.1800 Computer having the following configuration:-

1802 Processor-Controller with 16,384 words of core storage

2 2310 Disk Drives Model A

1 2401 Magnetic Tape Drive (30 kc/s)(7 Track)

1442 Model 6 Card Read - Punch

1443 Printer, 240 lines/minute

1816 Keyboard-Printer

Facit Paper Tape Reader, 1000 Characters/second

Facit Paper Tape Punch, 150 Characters/second

The operating system was TSX Version 3

N.I.O. PROGRAM 138

Title	High Speed paper tape listing
Name	PALIS
Computer	IBM 1802
Language	1800 Assembler
Purpose	To list paper tape punched in PTTC/8 code.

Input	// JOB
	// *Project No./Name/Job Title
	// XEQ PALIS
	*CCEND

Operation	Place the paper tape to be read in the reader with either runout or the first required character over the reading head. Load in the job cards in the usual way. Listing will continue to the physical end of the tape but may be halted at any point sooner than this by turning on data switch 14. Reading will cease at the end of the current record. To terminate the job, press console interrupt with program switch 7 on. Console STOP and START may be used in the usual way to give a pause without loss of data.
Method	The program uses two buffer areas in order to achieve peripheral overlap.
Execution Speed	About 4 records per second.
Programmer	Brian Hinde.

N.I.O. PROGRAM 139

Title — High Speed Card to Paper Tape Conversion

Name — CAPER

Machine — IBM 1802

Language — 1800 Assembler

Purpose — To convert complete program decks, including monitor control records, into paper tape in PTTC/8 code suitable for operating under TSXPT. Identification fields are not converted and following blanks are removed.

Input

```
// JOB
// Job No./Name/Job Title
// XEQ CAPER
*CCEND
```

followed by the deck to be copied including all control cards.

Follow the deck with a terminator card with all 12 punches in column 80.

Output — A paper tape version of the input deck suitable for operating under TSXPT.

The tape begins with the first character from column 1 of the first card. The first graphic character output on each line is preceded by a case shift character. Characters not in the PTTC/8 or card code set are replaced by a space. Columns 1 - 72 only are converted and blanks following the last graphic character of each card are removed. Cards with cols. 1 - 72 blank are output as SP NL.

Method — The program begins by disabling the //b check in CARDN. Card columns 1 - 72 are then read, converted to PTTC/8 code and following blanks are removed. The output is punched while the next card is being read. Each card is checked for all 12 punches in column 80. On reading such a card the //b check is enabled and the abort indicator is turned off to suppress the N03 error message.

The program is intended for converting programs for the ship system that have been developed on cards. It may be used for converting data if only columns 1 - 72 are used and if it is acceptable to have following blanks removed. Otherwise, an alternative utility program should be used (see DPG/P/9).

The program was designed to operate under TASK/TSK Version 2 or 3. The TASK EQUATE cards may need changing for future versions.

Note — It is most important that the terminator card is not omitted. If it is, the operator will not be able to stop the computer reading cards and converting them to paper tape!

Execution Speed — 100 - 140 characters per second, depending on record length.

Programmer — B. J. Hinde

N.I.O. PROGRAM 140

Title	Duplicate paper tape
Name	PAPER
Machine	IBM 1800
Language	1800 Assembler
Purpose	To copy 8 track paper tape (either binary or PTTC/8).

Input	// JOB
	// XEQ PAPER
	*CCEND

Method	Data switch 15 on to execute Data switch 14 on to copy deletes, off to pause at deletes Data switch 13 on for PTTC/8, off for binary tapes Data switch 0 on for continuous copy, off to do 2 frames and pause. These messages appear on the typewriter preceeding execution. Program then waits until tape to be copied has been loaddand START pressed. Return to supervisor by data switch 15 off. Options may be changed during execution.
Output	Paper tape copy on 'process' punch on ship.
Subroutines used	PAPIN, TYPEN.
Programmer	D. Brown.

N.I.O. PROGRAM 142

Title: High speed card listing

Name: CALIS

Machine: IBM 1802

Language: 1800 Assembler

Purpose: To list cards punched in IBM EBCDIC card code on the 1443 printer.

Input:

// JOB

// Job No./Name/Job Title

// XEQ CALIS

*CCEND

followed by the deck to be listed. The deck may contain control cards (including //b cards).

Follow the deck with a terminator card with all 12 punches in column 80.

Output: A listing on the 1443 of the deck, card column 1 being printed in character position 1, etc. A paper throw will occur when a channel 12 punch is detected in the printer carriage control tape. Characters not in the IBM card code are replaced with a space.

Method: The program begins by disabling the //b check in CARDN. The card is then read and converted to printer code. The output is printed while the next card is being read. Each card is checked for all 12 punches in column 80. On reading such a card the //b check is enabled and the abort indicator is turned off to suppress the NO3 error message.

The program was designed to operate under TASK/TSX Version 2 or 3. Certain EQUATE cards may need changing for future versions.

Note: It is most important that the // XEQ card, the *CCEND card and the terminator card be correctly punched and correctly positioned in the deck. If they are not, either control cards in the deck to be copied will be acted upon, or else the computer will continue reading and listing cards until the operator intervenes.

Execution Speed: About 3 cards per second.

Programmer: B. J. Hinde

N.I.O. PROGRAM 169

Title: High Speed Paper Tape to Cards

Name: PACAR

Computer: IBM 1800

Language: 1800 Assembler

Purpose: To convert paper tape to cards.

Input:

```
// JOB
// *Project No./Name/Job Title
            CC.16
// XEQ PACAR   FX
```

Operation: Reading will stop if data switch 14 is turned on and can be restarted by turning 14 off and pressing START. Program finishes by sensing a reader stop character on paper tape.

Method: The program uses two buffer areas in order to achieve peripheral overlap.

Programmer: Roy Bromley.

N.I.O. PROGRAM 175

Title	Prepare PTTC/8 paper tapes
Name	PREP
Machine	IBM 1801
Language	Fortran
Purpose	To prepare paper tapes in PTTC/8 using the typewriter - keyboard for input and nonprocess P/T punch for output.

Input	// JOB X
	// XEQ PREP FX
	// JOB
	// END

Method	Start typing in column 1, press EOF after each record; that record will then be punched. Return to supervisor by typing /* followed by EOF.
Programmer	B. J. Hinde

N.I.O. PROGRAM 176

Title	Convert Binary tape to Hexadecimal PTTC/8 Tape
Name	BIHEX
Machine	IBM 1800
Language	Assembler
Job Description	To convert binary tapes to hexadecimal to be listed offline. Written for use on ship system for easier reading of machine code tapes. Initial deletes are ignored and program will stop if 5 or more deletes are detected. Instructions for operation are typed when program is entered.

Input	// JOB
	// XEQ BIHEX FX
	*CCEND
	// JOB
	// END

Ouput	Paper tape
Programmer	J. Sherwood

N.I.O. PROGRAM 177

Title	List paper tape
Name	PAPRA
Machine	IBM 1800
Language	1800 Assembler
Purpose	To list paper tape on the 1816 typewriter

Input	// JOB
	// XEQ PAPRA
	*CCEND

Method	LOAD TAPE AND PRESS START 14 ON TO PAUSE 15 ON TO STOP PAUSES IF // END DETECTED The above messages are printed on the typewriter and program waits until tape to be listed has been loaded and 'START' pressed. The program will go into a 'WAIT' condition if data switch 14 is turned on or if a '// END' is detected. Program will end if data switch 15 is turned on.
Output	Listing on 1816 typewriter.
Subroutines used	WRTYN, PAPPR, PAPTN.
Programmer	R. Bromley

N.I.O. PROGRAM 182

Title	Stack blank cards
Name	BLAST
Machine	IBM 1800
Language	1800 Assembler
Purpose	To remove blank cards from a mixed deck. Can be used to check blank cards.
Use	The CARDN search for // cards is disabled. The input cards must end with a normal terminator card (all holes col. 80) to re-enable this.
Subroutines Used	CARDN

Control Cards	// JOB
	// XEQ BLAST FX
	*CCEND
	Followed by cards to be sorted and terminator card
	// JOB
	// END

Programmer	J. Sherwood

N.I.O. SUBPROGRAM -20

Title	Reverse Data Array
Name	Subroutine RVERS (KRAY,I)
Language	1800 Fortran IV
Machine	IBM 1800
Purpose	A call to the subroutine will reverse the order of the elements 1 to I in the integer array KRAY.
Programmer	B. J. Hinde

N.I.O. SUBPROGRAM -21

Title — Convert BCDIC code to EBCDIC

Name — SUBROUTINE READM(LIST(K))

Machine — IBM 1800

Language — 1800 Assembler

Use — Execution of

```
     CALL READM(LIST(K)) in a Fortran program
 or  BSI  L    READM
     DC        Address of Word Count preceding
               data array LIST
```

in an Assembler program

causes the characters packed two per word in LIST(1) to LIST(K-1) to be converted from BCD interchange code to EBCDIC. LIST(K) should contain the number of words to be converted (half the number of characters).

Method — The subroutine uses an 1800 version of 1130 ZIPCO to carry out the code conversion. The conversion table used in DICEB (N.I.O. subprogram -13).

Programmer — B. J. Hinde.

N.I.O. SUBPROGRAM -22

Title Paper Tape free format read routine

Name Subroutine UNFMT (N, ELMNT, IPOS, ITEST)

Computer IBM 1800

Language 1800 Fortran IV

Purpose The subroutine enables Fortran programmers to read paper tape data punched in free format. Numbers, in any of the usual forms, need only be separated by one or more spaces, tabs on newlines.

Use Each entry to the subroutine by means of

```
CALL UNFMT(N,ELMNT,IPOS,ITEST)
```

will read N elements from the paper tape reader (LUN = 4 for lab. system, = 2 for ship system). An element is defined as an integer, fixed point number or floating point number as defined below, being separated from the next element by at least one space, tab, or newline character. Elements must not, themselves, contain spaces. The values of the N elements are placed in the one dimensional real array ELMNT. If a spurious character (see below) is detected at any stage during reading, ITEST is set equal to the element number of ELMNT containing that character (i.e. all elements up to and including ELMNT (ITEST - 1) will be correctly evaluated) and control will be returned to the calling program. The programmer should follow each call, therefore, by a test of ITEST. If ITEST = 0 then the read has been successful. If ITEST ≠ 0 only ITEST - 1 elements are available for use. A subsequent call to UNFMT will begin by reading the element after the one containing the spurious character.

IPOS should be set by the programmer to 0 before the first CALL UNFMT relating to a particular data tape. On return from UNFMT (a normal return or a return caused by reading a spurious character) IPOS will be equal to the position in its record of the last character read. This should not normally be altered by the programmer as it will be used by the next call to UNFMT to determine where to start analysing the input buffer. However, increasing IPOS by 1 will cause a character to be skipped before the next read, decreasing IPOS by 1 will cause a character to be read twice. Care should be taken to see that IPOS always lies in the range 0≤IPOS≤80.

e.g. If a data tape contains

```
123   5·72   52E4   57·15E15
5   -9·3   /   41·73   0·157E-1   9·794   22
```

a program such as

```
      IPOS = 0
      CALL UNFMT(3,ARRAY,IPOS,ITEST)
      IF(ITEST) 1,2,1
2     CALL UNFMT(5,ALIST,IPOS,ITEST)
      IF(ITEST) 3,4,3
4     CALL UNFMT(3,BLIST,IPOS,ITEST)
      ..........
3     N = ITEST - 1
      GOTO 4
      ..........
```

will result in

```
      ARRAY(1) = 0·123E03
      ARRAY(2) = 0·572E01
      ARRAY(3) = 0·520E06
      ALIST(1) = 0·5715E17
      ALIST(2) = 0·5000E01
      ALIST(3) =-0·9300E01
      BLIST(1) = 0·4173E02
      BLIST(2) = 0·1570E-01
      BLIST(3) = 0·9794E01
```

ALIST(4) and ALIST(5) will be unaltered.

Restrictions

a) ELMNT must be dimensioned to a size equal to or greater than the value of N.

b) The first character of each element must be a digit (0-9), a minus sign, or a decimal point.

c) Each element must be separated by at least one space, tab, or newline character.

d) Spaces may not appear within any elements.

e) The letter E can appear anywhere in an element (except as the first character of an element) but it can be followed only by a minus sign (optional) and a maximum of two digits (0-9)

f) Erases and superfluous case shift characters may be used freely without restriction. Characters other than 0 to 9 - . E NL TB SP ER LC UC are considered superfluous and will be faulted.

Notes

The subroutine is a modification of the E.P.L. subprogram NOFMT (1130-7.0.001) - Unformatted Card Read Subroutine - written by F. L. Friedman, Goucher College, Baltimore. It uses subroutine QQQ for exponent evaluation and subroutine GET (part of 1130-SE-25X) for element evaluation.

Programmer

B. J. Hinde

N.I.O. SUBPROGRAM -23

Title

Read and check paper tape from data loggers

Name

Subroutine DLOG (INRAY,N,M,NERR,ICHAR)

Machine

IBM 1800

Language

1800 Fortran IV

Purpose

To check data consisting of up to 81 characters in ICHAR in A1 format produced by automatic data loggers in a fixed format. The subroutine checks the number of integers and the number of digits per integer and looks for spurious characters. Numbers are assumed to be separated by at least one blank. Correct records are output in an integer array. Errors are indicated and control is returned to the user.

To Use

A CALL DLOG(INRAY,N,M,NERR,ICHAR,L) will check up to L characters in ICHAR in A1 format and place the N integers into INRAY. The last character in ICHAR should be an EBCDIC newline.

Data Input

N - the expected number of integers per record

M - the expected number of digits (0 to 9) per integer

L - no characters in ICHAR including newline.

These two integers are input through the argument lists.

Output

a) INRAY(1) to INRAY(N) contains the N integers if the record read contains no errors.

b) NERR this is set within the subroutine according to the error condition found.

NERR=0 - no errors

NERR=1 - more or less than M digits in any integer

NERR=2 - more or less than N integers

NERR=3 - a character other than 0-9, - or blank has been found

NERR=4 - minus sign (-) in the wrong place, i.e. not at beginning of number

NERR=5 - no end of record marker has been found. Characters beyond the 80th on this line will be lost.

c) If NERR ≠ 0 then ICHAR(1) to ICHAR(L) contain the faulty input record in A1 format. If more than one error occur, the first error is listed.

Method

Up to 81 characters in A1 format are scanned for an EBCDIC NL character and if fewer than 80 characters/record are present, the remainder of the record is filled with blanks. The record is then checked for spurious characters. If none are present the positions of the signs are checked, and the number of blanks, and the characters are read. If the characters are correct, they are put into INRAY with the appropriate sign. Otherwise NERR is set to the appropriate error number, and there is an immediate exit from the subroutine.

Execution Time

Approx. 4 mins/100 records.

Programmer

R. A. Howarth

N.I.O. SUBPROGRAM -25

Title Convert EBCDIC code to BCDIC

Name SUBROUTINE WRITM(LIST(K))

Machine IBM 1800

Language 1800 Assembler

Use Execution of

```
     CALL WRITM(LIST(K)) in a Fortran program
 or  BSI  L    READM
     DC        Address of Word Count preceding
               data array LIST
```

in an Assembler program.

causes the characters packed two per word in LIST(1) to LIST(K-1) to be converted from EBCDIC to BCD interchange code. LIST(K) should contain the number of words to be converted (half the number of characters).

Method The subroutine uses an 1800 version of 1130 ZIPCO to carry out the code conversion. The conversion table used in EBCDI (N.I.O. subprogram -10).

Programmer B. J. Hinde

N.I.O. SUBPROGRAM -47

Title — Punch ICT Atlas/Orion 7 track paper tape

Name — WRITA

Machine — IBM 1800

Language — 1800 Assembler

Use — Execution of CALL WRITA(LIST(N+1)) in a Fortran program

```
          or CALL WRITA
             DC   Address of word count preceding data
                  array LIST in an Assembler program
```

where LIST(N+1) = N and LIST(N) to LIST(1) contains the string of EBCDIC(A1) characters to be punched, will cause one record to be punched on paper tape in ICT Atlas/Orion 7 track code. N is equal to the number of characters to be punched.

e.g. To punch an 80 character line:

```
      DIMENSION LIST(81)
      LIST(81) = 80
```

Place characters in order in LIST starting at LIST(80) and ending at LIST(1)

```
      CALL WRITA(LIST(81))
```

If a shorter record is required, set the word count accordingly:

e.g. 12 characters:

```
      LIST(13) = 12
      CALL WRITA(LIST(13))
```

A newline character will be punched at the end of each record.

If LIST(N+1) = 0 then 6 inches of upper case runout followed by a newline will be punched. e.g.

```
      CALL WRITA(0)
```

Method — The subroutine uses an 1800 version of 1130 ZIPCO to carry out the code conversion. The conversion table used in EBCAT (N.I.O. subprogram -48). The case bit of each character is examined and an appropriate case character is placed in the left-half of each word with the Atlas code character in the right-half. Superfluous case shift characters are then removed and the remaining characters packed two per word prior to punching.

Subroutines called — ZIPCO, EBCAT, PAPTN.

Notes

Techniques for placing real numbers or integers into A1 format are described in DPG/P/10 and in the Commercial subroutine package 1130-SE-25X.

The subroutines PUT, WHOLE and NSIGN in this package will prove useful. The NIO subroutine RVERS may be used to reverse arrays prior to a CALL WRITA.

Restrictions

The maximum record length is 80 characters. Up to 79 case shift characters can be generated in any one line.

Programmer

B. J. Hinde.

N.I.O. SUBPROGRAM -48

Title

EBCDIC to ICT ATLAS/ORION 7 track paper tape code conversion table.

Name

EBCAT

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 lists of the EBCDIC character to be converted (input character) are used as an address. This 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.
The high-code bit of the Atlas code character designates the case, 1 being upper case, 0 being lower case.
All EBCDIC codes not having an Atlas equivalent are converted to ERASE.

To obtain the address of the table entry point
CALL EBCAT

Programmer

Eileen Page

N.I.O. SUBPROGRAM -53

Title	ICL Mercury / Atlas 5 track paper tape code to EBCDIC conversion table.
Name	MADIC
Machine	IBM 1800
Language	1800 Assembler
Use	The table consits of 256 characters - 128 words with two 8 bit characters per word. The seven low-order bits of the Atlas character to be converted are used as an address. This address designates the position in the table of the corresponding conversion character. The high order bit (bit 0) of the input character is set by the user. When bit 0 is 1, denoting an upper case character to be converted, the left half of the word is used. When bit 0 is 0, denoting a lower case character, the right half of the word is used. All Atlas codes not having an EBCDIC equivalent are converted to ERASE. To obtain the address of the table entry point CALL MADIC.
Programmer	Eileen Page

N.I.O. SUBPROGRAM -55

Title	Busy test on typewriter/keyboard
Name	KEYBD
Machine	IBM 1800
Language	1800 Assembler
Purpose	To test the TYPEN routine for busy status. TYPEN is busy when either buffered typewriter output is waiting to be written, or when the proceed light is on.
Method	Called by CALL KEYBD, and this would immediately preceed a Fortran READ from the keyboard (LUN 6), especially when a message has just been written on the typewriter. KEYBD is not reentrant at present.
Called subroutines	TYPEN
Programmer	D. Brown.

N.I.O. SUBPROGRAM -56

Title — Atlas 7 track paper tape code to EBCDIC conversion table.

Name — ATLEB

Machine — IBM 1800

Language — 1800 Assembler

Use — The table consits of 256 characters - 128 words with two 8 bit characters per word. The seven low order bits of the 7 track character to be converted are used as an address. This address designates the position in the table of the corresponding conversion character. The high-order bit (bit 0) of the input character is set to 0 or 1 by the user. When bit 0 is 1 the left half of the word is used, and when bit 0 is 0 the right half of the word is used. Bit 0 designates the case of the Atlas character, 1 being upper case, 0 being lower case. All Atlas codes not having an EBCDIC equivalent are set to ERASE.

To obtain the address of the tables entry point call ATLEB.

Programmer — Eileen Page.

N.I.O. SUBPROGRAM -57

Title	Punching blank and delete codes for P/T leaders.
Name	PLEAD, PLAGG
Machine	IBM 1800
Language	1800 Assembler
Purpose	PLEAD will punch 12" of blank tape followed by 6" of 7-track delete tape on the '1055 punch' (LUN 4 on lab. system or LUN 2 on ship system). PLAGG will punch 6" of delete tape followed by 12" of blank tape.
Method	CALL PLEAD before any data is output to paper tape. CALL PLAGG after all data has been outputted to paper tape.
Called subroutines	PAPTN, QZSAV, QZEXT
Restrictions	The routines cannot be used for LUN 3 paper tape punch on ship system.
Core requirements	44 words
Programmer	D. Brown

N.I.O. SUBPROGRAM -58

Title	Read ICL Atlas/Orion/Mercury 5 track paper tape
Name	READ5 (Version 1)
Machine	IBM 1800
Language	1800 Assembler

Purpose

To read five track paper tape punched in ICL Atlas/Orion/Mercury teleprinter code. Version 1 will only read tapes punched throughout in figure shift.

To use

In a Fortran Program:

CALL READ5 (LIST(K + 1))
will read one record of paper tape placing N characters into LIST(1) to (LIST(N) in EBCDIC A1 format. N is the number of characters found in the record including the newline character (EBCDIC / 1540 translated from a line feed /16), excluding case shifts and erases. The maximum expected value of N,K, should be placed in LIST(K +1) by the user prior to the call.

If the number of characters found in the record is greater than K, LIST(K) will contain the Kth character and all succeeding ones including the newline will be lost. A subsequent CALL READ5 will always cause the next record to be read.

A record on five-track tape is defined as a string of characters ending with a line feed (LF). The carriage return character (CR) is converted to EBCDIC erase (/0707) as is any character not having an EBCDIC equivalent. For special character equivalents see the listing of MADIC, N.I.O. Subprogram -53

In an assembler program:
CALL READ5
DC   Address of word count K preceding the integer array where the data is to be stored.

e.g. To read a record such as :

```
123b456b78
```

consisting of 10 characters plus CRLF with a maximum expected record length of 11 characters plus CRLF

```
DIMENSION LIST (14)
LIST (14)  =  13
CALL READ5(LIST(14))
```

This will result in

```
LIST(1)  = /F140 (EBCDIC 1)
LIST(2)  = /F240 (EBCDIC 2)
  .            .
  .            .
  .            .
LIST(10) = /F840 (EBCDIC 8)
LIST(11) = /0707(EBCDIC erase)
LIST(12) = /1540 (EBCDIC NL)
LIST(13) = /4040 (EBCDIC space)
LIST(14) = /000D (Count of 13)
```

A *IOCS(PAPERTAPE) is NOT required when using READ5

Restrictions — The maximum character count, K, is 81

Subprograms called — ZIPCO(1800 version), PAPTN(modified as described in DPG/P/17, MADIC

Core requirements — 234

Execution time — About 250ms for a record of 50 characters.

Programmer — Brian Hinde

N.I.O. SUBPROGRAM -61

Title Decoding of BCD number

Name Subroutine BCDA (JWORD,IAR)

Machine IBM 1800

Operating System TSX

Language 1800 Assembler

Parameters JWORD is a word containing four BCD digits (8421 system) as shown below,

Bit No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BCD Code	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
	Digit 1				Digit 2				Digit 3				Digit 4			

IAR is a four word integer array.

Use After executing the instruction CALL BCDA (JWORD, IAR), digits one to four of the number will be contained in elements IAR(1) to IAR(4) respectively.

Programmer M. Fasham

N.I.O. SUBPROGRAM -64

Title
I.S.O. 8 track paper tape code to EBCDIC conversion table

Name
ISOEB

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 I.S.O. character to be converted are used as an address. This address designates the position in the table of the corresponding converted character. The high order bit (bit 0) of the input character is set by the user. 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 ISO codes not having an EBCDIC equivalent are converted to ERASE (/07). CR is converted to erase and LF to newline. For other conversion equivalents - see the program listing.

To obtain the address of the table entry point in a core load insert

```
     CALL ISOEB
```

in the calling program, but do not execute this instruction.

N.B. I.S.O. code is virtually identical
     to USASCII code.

Programmer
B. J. Hinde

N.I.O.SUBPROGRAM -65

Title   Read I.S.O. code eight track paper tape

Name   READ8

Machine   I.B.M. 1800

Language   1800 Assembler

Purpose   To read a record of eight-track paper tape punched in I.S.O. Code and convert it to A1 EBCDIC format.

To use.   In a Fortran Program

CALL READ8 (LIST(K+1))

will read one record of I.S.O. coded paper tape into the integer array LIST beginning at LIST(1). The information will be in LIST in A1 EBCDIC format.(LIST(K+1) should be set equal to K, the maximum number of characters to be read (excluding erases, carriage returns and blanks but including the newline character). If less than K characters are found, the remainder of LIST will be filled with spaces. If more than K Characters are found,LIST(1) to LIST(K) will be filled but the remaining characters in the record will be lost.

In an Assembler Program

CALL READ8

DC Address of word count,K, preceding the integer array where the data is to be stored.

A record on 8 track I.S.O. coded tape is defined as a strip of characters ending with a(newline) NL character (Hex.code /0A) or with CRLF(carriage return, line feed,codes /8D, /0A)

Special characters are treated as follows:

/00	Blank Tape	Ignored
/0C	Paper throw	Translated to EBCDIC erase /07
/8D	Carriage return(by itself)	Ignored
/FF	Erase	Ignored

For other character conversions see the conversion table listing ISOEB, N.I.0.subprogram -64

A *IOCS(PAPER TAPE) is NOT required when using READ8

Method The user's array LIST is first cleared to EBCDIC spaces. The tape is read, one character at a time into a buffer, testing for erases, blanks and carriage returns which are ignored and newlines which terminate reading. If LIST is filled before a newline is found then the rest of the record is skipped. The subroutine ZIPCO is called to carry out the code conversion using the table ISOEB. Finally the contents of the buffer are reversed into the user's array LIST.

Restrictions The maximum character count, K, is 81
I.S.O. code is an EVEN parity code. Violation of parity will cause the wrong character to be converted to EBCDIC erase (/07)

Subprograms called ZIPCO (1800 version), PAPTN (modified as described in [illegible] DPG/P/17), ISOEB.

Core requirements 224 words.

Execution time 250 milliseconds to read a record of about 50 characters.

Programmer Brian Hinde.

N.I.O.Subprogram -67

Title	Skip to Next File
Name	NEXTF
Machine	IB M 1800
Type of Program	Non-process
Language	1800 Fortran IV
Subprograms called	MAGOP

Purpose

To position a magnetic tape to the beginning of the next file (i.e. just after the next end-of-file mark).

Use

The calling sequence for NEXTF is:

```
CALL NEXTF(MODE)
```

MODE is a hexadecimal constant specifying the mode of the magnetic tape. The calling program should contain a data statement of the form.

```
DATA MODE/ZOABC/
```

Where A = 0 for odd parity
= 1 for even parity

B = 0 for 800 b.p.i.
= 1 for 200 b.p.i.
= 2 for 556 b.p.i.
} 3 bytes per word. 7 track tape only

= 4 for 800 b.p.i.
= 5 for 200 b.p.i.
= 6 for 556 b.p.i.
} 2 bytes per word

C = 0 for first tape drive
= 1 for second tape drive

For tapes written with a FORTRAN WRITE statement, set MODE = 0 for first tape drive, and MODE = 1 for second tape drive.

Method

NEXTF does not rewind the magnetic tape. It starts reading tape wherever the tape is positioned, and continues reading until an end-of-file record has been read.

ERROR MESSAGES

If, in reading the tape, an uncorrectable tape error is found (error 4 from MAGOP), the program prints 'UNCORRECTABLE TAPE ERROR' and continues. If a correctable tape error is found (error 3 from MAGOP), the program prints 'READ CHECKS HAVE OCCURRED' and continues reading the next record.

Programmer

J.Webster