J. Crease

NATIONAL INSTITUTE OF OCEANOGRAPHY
WORMLEY, GODALMING, SURREY

General Purpose Utility Programs

N.I.O. INTERNAL REPORT NO. N. 27

AUGUST 1972

NATIONAL INSTITUTE OF OCEANOGRAPHY
WORMLEY, GODALMING, SURREY

GENERAL PURPOSE
UTILITY PROGRAMS

N.I.O. Internal Report No. N27

AUGUST 1972

N.I.O. PROGRAMS N.27

(Subprograms have negative numbers)

Input/Output and Code Conversion

- 72	Integer version of the Commercial Subroutine GET to convert from A1 format to I format	IGET
- 82	Integer version of the Commercial Subroutine PUT to convert from I format to A1 format	IPUT
- 91	Truncated EBCDIC to EBCDIC	EXPND
-112	Decoding of a BCD number	BCDB
-113	BCDIC to card-code conversion table	DICHO
-119	BCDIC to EBCDIC conversion table	BCDA1
-120 to -123	To provide fast conversion from one code to another with or without reversal of the output array. The input and output arrays may be in packed or unpacked format	ZZIPU ZZIUP ZZIUU ZZIPP
-124	EBCDIC to BCDIC conversion table	A1BCD
-132	EBCDIC to ISO conversion table	A1ISO
-133	ISO to EBCDIC conversion table	ISOA1
-151	Magnetic tape operation	MAGOP
-161	Free format read	FREAD
-165	Completion code tests for Paper Tape	TPTAP
-166	1442 Completion code tests	TCARD
-167	1443 Completion code tests	TPRIN
-173	To double buffer the 1443 line printer	(PBUFA (PBUFX
-174	To double buffer the 1442 card reader	(CBUFA (CUBFX
-195	PTTC/8 to ASCII Conversion Routine	PTASC
-196	PTTC/8 to ASCII, PTTC/8 to KDF9 table	TPTAK
-197	EBCDIC to PTTC/8 to KDF9 table	A1PTC
-203	PTTC/8 to EBCDIC table	PTCA1
106	1130/1800 object card to PTS paper tape conversion	C DPTS
232	Conversion of CDS object card deck to an IPL mode	PUNPT
247	Fortran to BCD magnetic tape conversion	FORBD
248	BCD to Fortran magnetic tape conversion	BDFOR
276	OBJPN to PTS conversion	OBPTS
279	PTTC/8 to ASCII paper tape conversion	PADPA
280	PTS to OBJPN conversion	PTSOB

Number and Bit Manipulation

- 89	Variable length unpacking of bits in a word	NPACK
-146	Conversion of multiple precision integer to one word integer	ICONB
-158	Free format number conversion routine	NCONV
-163	Free format expression conversion	FCONV
-187	Maximum or Minimum integer value	MAX/MIN
-199	Convert a real number to A1 format	CONF
-201	Convert a real number to E format	CONE

Operational Utilities, Copying and Listing Programs

- 79	Pause if data switch 0 is on	DASWO
-159	Change plotter pen	XPEN
-162	Fast Intra-Core copy	COPYU
154	Tape copying utility program	TCUP
173	Edit paper tapes conversationally	PUNED
229	List BCD tapes	LIMP
230	List Fortran formatted magnetic tapes	TALIS
235	To compare two program card decks	FCOMP
253	To find size of a BCD magnetic tape block	BKSIZ
254	Dump program from magnetic tape to card and line printer	CDUMP
264	Correct a binary card	BREAD

Machine Functions, File Manipulation

-160	Search for literal characters	SKIPD
-164	Dynamic allocation of timers	LOTIM
-194	Find a record for a given time in a dynamic file	FINDT
-198	Test if Day1, Time1 is before Day2, Time2	ITMLM
-202	Load floating point accumulator	LOADF
268	Plot ship's data files	GRAPH

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.

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 32,768 words of core storage

3 1810 Disk Drives Model A

2 2401 Magnetic Tape Drives (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 Systems were T.S.X. Version 3 or M.P.X. Version 3.

N.I.C.SUBPROGRAM -72 CLASSIFICATION U

Title Integer version of the Commercial Subroutine GET to convert from A1 format to I format.

Name IGET

Machine IBM 1800

Language 1800 Assembler

Operating System TSX

Purpose A function subroutine to convert characters in A1 format to an integer number in I format.

To use IGET(IRAY,IFST,ILAST)

where IRAY is a one dimensional integer array which holds the characters to be retrieved.

IFST is an integer constant, variable or expression which is the position of the first character to be retrieved.

ILAST is an integer constant, variable or expression which is the position of the last character to be retrieved where ILAST≥IFST.

e.g.

	IRAY	NUM
Before:	12345678	0

NUM = IGET(IRAY,3,5)

	IRAY	NUM
After:	12345678	345

Note If any characters other than a blank, numerals or a minus sign are encountered the result will be set to zero. The minus sign, if present, should appear at the most significant end of the number.

N.B. The largest number which can be handled is 32767. Anything greater than this will produce unpredictable answers.

Programmer Paul Lindsay

N.I.C.SUBPROGRAM -82 **Classification U**

Title Integer version of the Commercial Subroutine PUT to convert from I format to A1 format.

Name IPUT

Machine IBM 1800

Language 1800 Assembler

Operating System TSX

Purpose A subroutine to convert numbers in I format to the equivalent characters in A1 format.

To use CALL IPUT(IRAY, IFST, ILAST, NUM)

where IRAY is a one dimensional integer array into which the resulting characters will be placed. (right-justified).

IFST is an integer constant, variable or expression which is the position of the first or least significant character to be placed in the field.

ILAST is an integer constant, variable or expression which is the position of the last or most significant character to be placed in the field where ILAST≥ IFST.

If output field too small, all extra characters will over-write each other in IFST.(i.e. most significant, left-justified).

NUM is the integer constant, variable or expression, which is to be put in A1 format, into the array IRAY.

e.g.

```
               IRAY          NUM
   before   ABCDEFG          -99
               IPUT(IRAY,3,6,NUM)
               IRAY          NUM
   after    ABb-99FG         -99
```

Programmer Paul Lindsay.

N.I.O. SUBPROGRAM -91 CLASSIFICATION U

Title Truncated EBCDIC to EPCDIC

Name EXPND

Language 1800 Fortran IV

Operating System T.S.X.

Machine I.B.M. 1800

Purpose To convert an integer array of two elements in truncated EBCDIC into an array of five elements in EBCDIC.

Use CALL EXPND (IN,IOUT)

Where IN is an array of two elements containing the name to be converted IOUT is an array of five elements containing the results in A1 format.

Programmer W. Strudwick.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	-112
Program Title: (For Lib. List)	Decoding of a BCD number
Classification Letter:	U
Program Name:	BCDB
Version Number:	20
Type of Program:	Subroutine
Operating System:	T.S.X.
Language:	Fortran
Programmer/Date:	M. Fasham
Program Passed By:	E. Page
Description:	Converts four BCD digits to four Integer values
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	BCDA
Programs Called:	
Files Called:	

Operation and Method:

Subroutine BCDB (JWORD,LAR,ISW)

JWORD is a word containing four BCD digits as shown below,

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

LAR is a four word integer array.

ISW is put equal to zero for 4221 BCD code or one for 8421 BCD code.

On execution of the instruction CALL BCDB(JWORD,LAR,ISW) digits one to four of JWORD will be decoded and put in elements LAR(1) to LAR(4) respectively.

The operation is carried out using Assembler subroutine BCDA which assumes the number is 8421 BCD code. If ISW had been put equal to zero the 8421 value is converted to a 4221 value. The 4221 code assumed is given below:-

Number	4	2	2*	1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	1	0
5	0	1	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

N.I.O. SUBPROGRAM -113 CLASSIFICATION U

Title	BCDIC to card-code conversion table
Name	DICHO
Machine	IBM 1800
Language	1800 Assembler
Operating System	T.S.X. or M.P.X.
Use	The table consists of 128 characters - 64 words with two eight-bit characters per word. The left hand characters are the 8-bit card code NUL character. The right-hand characters are 8-bit card code corresponding to the BCDIC character which is the relative address in the table. See the description of ZIPCO for explanation of 8-bit card code. When used with ZIPCO, the output characters are unpacked to 12-bi card code. To obtain the address of the first word of the table use CALL DICHO The table is mainly for use with ZIPCO, and operates in a complementary way to HOLDI.
Programmer	David Brown.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM: -119

Program Title: BCD to EBCDIC Conversion Table
(For Lib. List)

Classification Letter: U

Program Name: BCDA1

Version Number: 20

Type of Program: Subroutine

Operating System: T.S.M. or M.P.M.

Language: Assembler

Programmer/Date: J. Crease

Program Passed By: E. Page

Description: To provide a direct look up conversion table for use by program ZZIPU, ZZIPP, ZZIUP and ZZIUU, to convert array of packed input codes into an array of output codes. The conversion is from 7 track tape BCD to EBCDIC.

System Prerequisites:

Inskel Common Variables:

Subroutines Called: ZZIPU, ZZIPP, ZZIUP and ZZIUU

Programs Called:

Files Called:

Operation and Method:

The S/R consists entirely of a table of 64 B.C.'s. The input code is used as an address relative to the Head of the table to extract the output EBCDIC code named in the D.C. (e.g. tape BCD code / ØØ11 / extracts / 614Ø). The right half of the EBCDIC code is always 4Ø (blank) as the output is unpacked.

Note:

Codes in the input list greater than 63 will give an unprodictable output. These codes should not occur.of course.

N.I.O. SUBPROGRAM No. -120, -121, -122, -123 CLASSIFICATION U

Name ZZIPU, ZZIUP, ZZIUU, ZZIPP

Purpose To Provide fast conversion from one code to another with or without reversal of the output array. The input and output arrays may be in packed or unpacked format.

Machine IBM 1800

Operating System Assembler

S/R Selection The following table gives the appropriate S/R to use for a given conversion.

Output \ Input	Packed	Unpacked
Packed	ZZIPP	ZZIUP
Unpacked	ZZIPU	ZZIUU

Call The fortran CALL is similar for each pf the S/R's and is illustrated here by a call to ZZIPU

CALL ZZIPU(IN,LOUT,N,ABCD)

1. IN is the low core address of the input array. (In Fortran because the arrays are in reverse order IN would be equivalenced to the last word of the Fortran Input Array).
2. LOUT is equivalenced to the high core end of the output array if it is to be in reverse order to the input otherwise it should be at the low core end.
3. N is the word count of the input array. It should be negative if array is to be reversed.
4. ABCD is the name of an Assembler S/R containing a conversion table and should be declared in an EXTERNAL statement in the calling program (see Program BCDA1 as an example of its structure).

Restrictions One word integers are assumed.

Notes

1. ZZIPU output array has twice as many variables as input
ZZIUP " " " half " " " " "
2. Effort has been concentrated on ways of reducing the inner loop execution time as follows:-

a) Counting and addressing is all done with index registers as the MDX L instruction is slow.

b) By combining conversion, unpacking and reversal aspects, intermediate storage of data is avoided.

c) Although the ABCD table is twice as long (because it isn't packed) as the corresponding ZIPCO tables the programs here are considerably shorter than ZIPCO and there is little loss in storage space.

3. ABCD need only be as long as required for the particular conversion. 7-track tape BCD to EBCDIC for example requires only 64 DC statements. It works as a direct table look up. The binary value of the input data is used as relative address and added to the entry point to access a DC statement containing the output code in unpacked format (e.g. in EBCDIC we have /XY40 where XY is the decoded EBCDIC 8 bit character and 40 fills the right half with a space character).

Undefined conversions can be put either to space or invalid code depending on ones need to detect such codes.

Timing   90 - 100 μs/character

Programmer   J. Crease.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	-124
Program Title: (For Lib. List)	EBCDIC to BCDIC conversion table
Classification Letter:	U
Program Name:	A1BCD
Version Number:	20
Type of Program:	Subroutine
Operating System:	T.S.X. orM.P.X.
Language:	Assembler
Programmer/Date:	M. Olliff
Program Passed By:	E. Page
Description:	To provide a direct look up conversion table for use by programs ZZIPP, ZZIPU, ZZIUP and ZZIUU, to convert from 7-track tape EBCDIC to BCD.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	ZZIPP, ZZIPU, ZZIUP and ZZIUU
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is a table of 256 D.C.'s. The input code is used as an address relative to the head of the table to extract the output BCD code named in the D.C. (e.g. tape EBCDIC code /ØØC2 extracts /32ØØ). The right half of the BCD is blank as the output is unpacked.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	-132
Program Title:	EBCDIC to ISO conversion table.
(For Lib. List)	
Classification Letter:	U
Program Name:	A1ISO
Version Number:	20
Type of Program:	Subroutine
Operating System:	T.S.A. or M.P.X.
Language:	Assembler
Programmer/Date:	M. Olliff
Description:	To provide a direct look up conversion table for use by subprograms ZZIPP, ZZIPU, ZZIUP and ZZIUU, to convert from EBCDIC to 8 track ISO paper tape code with even parity.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	ZZIPP, ZZIPU, ZZIUP and ZZIUU
Programs Called:	
Files Called:	
Operation and Method:	

The subroutine is a table of 256 Data Constants. The input code is used as an address relative to the head of the table to extract the output ISO code value from the data constant, (e.g. EBCDIC code /ØØC2 extracts /42ØØ). The right half of the word is blank as the output is unpacked. Any packing required is done by the ZZIPP or ZZIUP routines.

Note:

As the parity bit is set in the Data Constants, this table is not suitable for use when converting to IBM ASCII paper tape code, as the latter has odd parity.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM:	-133
Program Title: (For Lib. List)	ISO to EBCDIC conversion table
Classification Letter:	U
Program Name:	ISOA1
Version Number:	20
Type of Program:	Subroutine
Operating System:	T.S.R. or M.P.R.
Language:	Assembler
Programmer/Date:	M. Olliff
Program Passed By:	E. Page
Description:	To provide a direct look up conversion table for use by subprograms ZZIPP, ZZIPU, ZZIUP, and ZZIUU, to convert from 8 track paper tape to ISO or IBM ASCII code to EBCDIC.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	ZZIPP, ZZIPU, ZZIUP, ZZIUU
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is a table of 128 Data Constants. The input code is used as an address relative to the head of the table to extract the output EBCDIC code named in the data constant (e.g. ISO code / 42 extracts / C240). It is assumed that the parity bit has been removed from the code value by the program which requires the conversion. The parity bit is the left most bit for these codes. This enables the same table to be used for ISO 8 track paper tape code with even parity and IBM ASCII with odd parity.

Note:

The parity bit is not removed by the subprograms ZZIPP, ZZIPU, ZZIUP or ZZIUU.

N.I.O. SUBPROGRAM -151 CLASSIFICATION U

Title	Magnetic Tape Operation
Name	MAGOP
Machine	I.B.M. 1800
Language	1800 FORTRAN IV
Operating System	M.P.X.

Purpose To perform all magnetic tape operations without the need to call the MAGT routine and without double buffering of I/O.

Use The subroutine is used through the normal call statement.

```
CALL MAGOP(MODE,JOP,LIST(K+1),NERR)
```

where MODE is set in a DATA statement of the form

```
DATA MODE/ZOABC/
```

where A=0 for odd parity tape
 =1 for even parity tape
B=0 for 3 bytes per word, 800 bpi
 =1 for 3 bytes per word, 200 bpi
 =2 for 3 bytes per word, 556 bpi
 =4 for 2 bytes per word, 800 bpi
 =5 for 2 bytes per word, 200 bpi
 =6 for 2 bytes per word, 556 bpi
C=0 for first tape drive
 =1 for second tape drive

JOP is the operation to be performed.

=1 Read one record with retries
=2 Read one record without retries
=3 Write one record with retries
=4 Write one record without retries
=5 Rewind
=6 Rewind and Unload
=7 Backspace one record
=8 Write end of file marker
=9 Erase

LIST(K+1) contains the record length in words, where LIST(1) to LIST(K) will be the integer array containing the data to be read or written.

NERR is an error parameter set up by MAGOP and is set as follows:-

=0 No errors

=1 End of file mark read

=2 Wrong length record specified

=4 Uncorrectable tape error

=5 Device logically offline

=6 Device not ready

=7 End of tape

=8 File Protect Error

=9 Request aborted

Called Subprograms MAGT

Programmer William Strudwick

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

NIO PROGRAM - 161

Program Title: (for Lib. List)	Free Format Read
Classification Letter:	U
Program Name:	FREAD
Version Number:	20
Type of Program:	Subroutine
Operating System:	M.P.X./T.S.X.
Language:	1800 Fortran IV
Programmer/date:	T. Voss / 3.11.71
Program Passed By:	C. Spackman
Description:	To read data in free format and return one value per call
System Prerequisites:	None
Inskel Common Variables:	None
Subroutines Called:	FCONV (which calls NCONV, SKIPD, GET, NZONE)
Programs Called:	None
Files Called:	None

Operation and Method:

A = FREAD(LIN,LOUT)

where LIN is the input LUN

LOUT is the output LUN for messages

A change in LIN only becomes effective when all values on the last record have been interpreted.

Inputs:

Input records may contain any number of free format numbers or expressions as laid down in DPG/P/32

Paper Tape When input is from paper tape, records must be of at least 80 characters, of which characters 1-80 will be interpreted. If less than 80 characters are present, FREAD will read into the next record. On the ship, where there is no // inhibit it could read over the // END

Ampersand & When an ampersand is encountered, the remainder of the record is ignored, thus allowing comments e.g.
29.3 14.8, 23E17 & DATA FOR GROUP 1

Output: Error messages from FCONV or NCONV may occur.

Size: 144 words

Free Format Input

Introduction

A large proportion of execution-time errors are a result of incorrect formatting in the input data. Fixed format data is very efficient for large quantities, produced either by punch operators or nachine, but is entirely unsuited to control data, where each record may have a different format. In the scheme described here input data is allowed without restriction on its format.

Free Format Numbers

With Free Format Numbers there is no distinction between fixed and floating point values, e.g.

Punched	Value Read
1Ø	1Ø
1Ø.	1Ø
23.2	23.2
232E-1	
.232E2	
E4	10^4
E-6	10^{-6}

The end of a number is indicated by any of the following:

ƀ	(blank)
,	(Comma)
	(New Line)
+	
–	
*	
/	
&	(Ampersand)

Free Format Expressions

With Free Format Expressions any number of Free Format Numbers may be joined by the five operators +,-,*,/,** to give an expression which will be evaluated on being read in:

Punched	Value Read
22/7	3.143
2**.5	1.4142

An expression is terminated by a blank, comma new line or ampersand.

Note (1): Expressions are evaluated strictly left to right - no brackets are allowed and the normal operator higherarchy does not exist; thus

2-1/2+3/2 is taken as 2 not 1

If the angle 28° 14' 23" were required in degrees, it could be written as follows

23/60+14/60+28

Note (2): As much help as possible is given in interpreting incomplete expressions e.g.

/3	is	read	as	1/3
**2		as		1**2
3--.5		as		3-(-.5)=3.5
29E--6		as		29-6=23

If no number follows an operator the remainder of the record will be searched for a continuation, thus

35+   4   5

would be read as 35+4   5

However, if no further numbers exist on the card the incomplete expression is ignored and a message printed.

Note (3): Punching errors can often be corrected without repunching. If -3 were punched with the sign omitted, it would be simple to correct the fault by subtracting 6 i.e. 3-6=-3

Method of Use

Free Format Input is implemented with the function subprogram FREAD(LIN,LOUT) where LIN is the input LUN (logical unit number) and LOUT the LUN for any messages.

Example:

```
A = FREAD(2,3)
```

causes the input stream to be scanned and the next value plaued in A.

Further Reading

For further details see the write-ups of

FREAD		NIO -161
NCONV	(Conversion of numbers)	NIO -158
FCONV	(Conversion of expressions)	NIO -163

Core and Timing

Free Format Input requires Fortran I/O in addition to the special conversion routines. Core requirements may be considerable, especially if the program is otherwise simple. Execution is slower than for formatted input. Where space is no problem or efficient use of LOCALS can be made, Free Format Input can be advantagous.

A. Voss
November 1971

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM - 165	
Program Title: (for Lib. List)	COMPLETION CODE TESTS FOR PAPER TAPE
Classification Letter:	U
Program Name:	TPTAP
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	Assembler
Programmer/date	M. Olliff / 10.12.71
Program Passed By:	E. Page
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called	PAPTN
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is passed the address of the link/busy parameter used in the previous call to PAPTN. A test for device busy is made. If the reader/punch is not busy, the completion code (LIST+6) is tested for device not ready. If the device is ready, the subroutine exits. If the device is not ready, the subroutine loops until it is ready and then repeats the previous call to PAPTN, waits until the call is complete and then exits. The subroutine can be used for punching or reading.

Inputs

```
In Fortran CALL TPTAP(LIST(9))
In Assembler CALL TPTAP
                 DC   LIST
```

after a call to PAPTN has been made. Can be used especially when double buffering is used

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -166

Program Title: (For Lib. List)	1442 Completion Code Tests
Classification Letter:	U
Program Name:	TCARD
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX Only
Language:	Assembler
Programmer/date:	M. Olliff / 10.12.71
Program Passed By:	E. Page
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	CARDN
Programs Called:	
Files Called:	

Operation and Method:
The subroutine is passed the address of the 1st word of the I/O list (link/busy parameter) used in the previous call to CARDN. A test for device busy is made. If the 1442 is not busy, the completion code (LIST + 6) is tested for device not ready. If the 1442 is ready the subroutine loops until it is ready and then repeats the previous call to CARDN, waits until the call is complete and then exits.

In Fortran

```
CALL TCARD (LIST(9))
```

In Assembler

```
CALL      TCARD
DC        LIST
```

after a call to CARDN has been made. Can be used especially when double buffering is used.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -167

Program Title: (For Lib. List)	1443 Completion Code Tests
Classification Letter:	U
Program Name:	TPRIN
Version Number:	20
Type of Program:	Subroutine
Operating System:	M.P.X. Only
Language:	1800 Assembler
Programmer/date:	M. Olliff / 10.12.71
Program Passed By:	E. Page
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	PRNTN
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is passed the address of the 1st word of the I/O list (link/busy parameter) used in the previous call to PRNTN. A test for device busy is made. If the 1443 is not busy, the completion code (LIST + 6) is tested for device not ready, if the 1443 is ready, the subroutine is exited. If the 1443 is not ready, the subroutine loops until it is ready and then repeats the previous call to PRNTN waits until the call is complete, and then exits.

Inputs:

In Fortran

```
CALL TPRIN(LIST(9))
```

In Assembler

```
CALL   TPRIN
DC     LIST
```

after a call to PRNTN has been made, can be used especially when double buffering is used.

This subroutine is recommended for <u>all</u> programs using the lineprinter for data output to increase run time.

NATIONAL INSTITUTE OF OCEANOGRAPHY
DATA PROCESSING GROUP
LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -173

Program Title: (For Lib. List)	To double Buffer the 1443 line printer.
Classification Letter:	U
Program Name:	PBUFA, PBUFX
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX Only
Language:	1800 Assembler
Programmer/date:	D. Brown/11.1.72 (from Westminster Hospital)
Program Passed By:	E. Page
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	PRNTN
Programs Called:	
Files Called:	
Operation and Method:	

The subroutine is called at the beginning of a Fortran program by CALL PBUFA. This causes the lineprinter to be double buffered during program execution i.e. a line is printed while the program continues to execute. This causes the program to run faster. The optimum is to have about 1/4 second C.P.U. time between Fortran WRITE statements. At the end of the program before the CALL EXIT statement, a statement CALL PBUFX must be executed to clear the print buffer.

This subroutine is recommended for <u>all</u> programs using card data input, to increase throughput and shorten running time.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -174

Program Title: (For Lib. List)	To double Buffer the 1442 card reader.
Classification Letter:	U
Program Name:	CBUFA, CBUFX
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX Only
Language:	1800 Assembler
Programmer/date:	D. Brown/11.1.72 (from Westminster Hospital)
Program Passed By:	E. Page
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	CARDN
Programs Called:	
Files Called:	
Operation and Method:	

The subroutine is called at the beginning of a Fortran program by CALL CBUFA. This causes the card reader to be double buffered during the program execution i.e. a card is read in, and while its data is being computed the next card is read in. This causes the program to run faster. The optimum is to have $\frac{1}{4}$ second C.P.U. time between Fortran READ statements. This will halve the run time. At the end of the program before the CALL EXIT statement, a statement CALL CBUFX must be executed to clear the card buffer. A blank card must be added to the back of the data cards.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -195	
Program Title: (For Lib. List)	PTTC/8 to ASCII conversion routine
Classification Letter:	U
Program Name:	PTASC
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	1800 Assembler
Programmer/date:	M. Olliff 11.5.72
Program Passed By:	E. Page
Description:	To convert from PTTC/8 code to ISO ASCII code
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	TPTAK
Programs Called:	
Files Called:	

Operation and Method:

The characters to be converted are packed two per word. The subroutine obtains the no. of chars. to be converted by multiplying by two the word count found at the beginning of the array containing the characters. Individual characters are isolated, and the value of each character is used to pick up the appropriate entry in the conversion table TPTAK. Upper case characters have /8Ø added to them to access the last half of the table. If a PTTC/8 case character itself is found, a delete is output in its place, and all the characters following it are set according to that case, until another case char. is found. The case is originally assumed to be lower, but the current case is retained on subsequent calls to the subroutine, unless the subroutine is reloaded between calls.

```
CALLING SEQUENCE                  CALL PTASC       AREA1  DC   WORD COUNT
                                  DC   AREA1              BSS   "     "
```

where AREA1 + 1 etc. contain the characters to be converted

or in Fortran

CALL PTASC(AREA(N))   AREA(N)=N-1

and AREA (N-1) to AREA(1) contain the characters.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -196

Program Title: (For Lib. List)	PTTC/8 to ASCII, PTTC/8 to KDF9 table
Classification Letter:	U
Program Name:	TPTAK
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	1800 Assembler
Programmer/date:	M. Olliff 11.5.72
Program Passed By:	E. Page
Description:	Direct look up conversion table
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is a table of 256 Data Constants. The input code (PTTC/8) is used as an address relative to the head of the table to obtain the ISO ASCII or KDF9 code value named in the data constants. The PTTC/8 code is assumed to have had /8Ø added to its value if it is an upper case character.

For example:

PTTC/8 'S' is an upper case char., value /32, so it becomes /B2. The value in the table at /B2 is /5363. The left hand value represents ISO ASCII code, and the right hand value KDF9 Code for 'S'. The KDF9 shift characters in the table have had /8Ø added to them.

For an example of calling the table see the coding for NIO Subprogram PTASC.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -197

Program Title: (For Lib. List)	EBCDIC to PTTC/8 table
Classification Letter:	U
Program Name:	A1PTC
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	Assembler
Programmer/date:	M. Olliff 3.7.72
Program Passed By:	W. Strudwick
Description:	Look-up table for conversion from EBCDIC to PTTC/8
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is a table of 256 data constants. The input code, EBCDIC is used as an address relative to the head of the table to obtain the PTTC/8 code value named in the data constants. If the PTTC/8 code is an upper case character, it will have had /8Ø added to its value.

For an example of calling a look-up table see the coding for PTASC (N.I.O. -195).

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -203

Program Title: (For Lib. List)	PTTC/8 to EBCDIC table
Classification Letter:	U
Program Name:	PTCA1
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	Assembler
Programmer/date:	M. Olliff 3.7.72
Program Passed By:	W. Strudwick
Description:	Lookup table for conversion to EBCDIC from PTTC/8
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

The subroutine is a table of 256 data constants. The input code (PTTC/8) is used as an address relative to the head of the table to obtain the EBCDIC equivalent.

The PTTC/8 code is assumed to have had /8∅ added to its value if it is an upper case character.

For an example of calling a look-up table see the coding for PTASC (N.I.O. -195).

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	106
Program Title: (For Lib. List)	1130/1800 object card to PTS tape conversion
Classification Letter:	U
Program Name:	CDPTS
Version Number:	20
Type of Program:	Mainline
Operating System:	T.S.X.
Language:	Assembler
Programmer/Date:	B. Spatz
Program Passed By:	E. Page
Description:	The program reads card object decks and punches them on paper tape in TSXPT PTS format.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	CARDN, PAPTN
Programs Called:	
Files Called:	

Operation and Method:

CDPTS reads 1130/1800 card object decks and punches them on paper tape in TSXPT PTS format. These tapes can be read by TSXPT, 1130 disk Monitor 1 and 2, 1130/1800 card/P.T. system and by C.E. diagnostic loaders.

Ensure paper tape punch is ready and run out blank tape. Then punch the following cards:-

```
//b̸JOB
//b̸XEQb̸CDPTS
*CCEND
```

followed immediately by the card object deck, followed by a terminator card with all rows punched in column 80.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	232
Program Title: (For Lib. List)	Conversion of CDS object card deck to an IPL mode tape
Classification Letter:	U
Program Name:	PUNPT
Version Number:	20
Type of Program:	Mainline
Operating System:	T.S.X.
Language:	Assembler
Programmer/Date:	B. Spatz
Program Passed By:	E. Page
Description:	The program reads object cards and punches out an IPL mode tape.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	CARDN, EBPRT, TYPEN, PAPTN
Programs Called:	
Files Called:	
Operation and Method:	

To execute the program punch the following cards:

```
//ƀJOB
//ƀXEQƀPUNPT
*CCEND
```

Place object deck of cards immediately after the *CCEND. The program will read the cards and punch out an IPL mode tape. When the end of program card is detected (word 3 equal tp ØFØØ) a column 5 punch is made on the IPL tape and the program exits.

Error Messages: The error message

OBJECT DECK ERROR N

can occur where N has the following meanings:-

1. Illegal type code.
2. Illegal type code.
3. CDS program contains an unresolved LIBF or CALL.
4. The CDS cards are not in correct sequence.
5. The first word of the program is not at core location zero.

Restrictions: The program to be converted must

1. Contain no unresolved LIBF's or CALL's.
2. Be relocatable or be origined to core location zero.
3. The entry point/starting address must be the first word of the program.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 247

Program Title: (For Lib. List)	Fortran tape to BCD tape conversion
Classification Letter:	S
Program Name:	FORBD
Version Number:	2Ø
Type of Program:	Utility non-process program
Operating System:	MPX
Language:	Fortran IV
Programmer/date:	M. Olliff / 7.12.71
Program Passed By:	
Description:	Conversion of a magnetic tape in formatted fortran to BCD
System Prerequisistes:	
Inskel Common Variables:	
Subroutines Called:	MAGOP(-151), NEXTF(-67), ZZIPP(-123), A1BCD(-124)
Programs Called:	
Files Called:	

Operation and Method:

The FORTRAN tape on drive 2 is positioned at the start of the first file to be converted, the BCD tape is positioned at the start of the first file to be written. The number of characters to be stored in a BCD block is calculated, and if this number exceeds 4ØØØ characters a message 'BLOCK SIZE TOO LARGE' is output on the 1443 printer and the program exits.

FORTRAN blocks are read off the tape, one block at a time, converted to BCD and stored one after another into a BCD block until the number of FORTRAN blocks stored is equal to the blocking factor. If the number of characters in the FORTRAN block is less than 146, the unused words at the end of the block are ignored and the blocks are packed together within the BCD block. If there is an odd number of characters in a FORTRAN block, then the right-hand half of the last word in the block is left blank. The BCD block is then written to the tape on drive 1.

This is repeated until a complete file has been converted. A tape mark is then written to the BCD tape and the following message is output on the 1443 printer.

FORTRAN FILE X COPIED TO BCD FILE S

The process is then repeated for the next file until the total number of files (as indicated on the header card) have been converted. If the file numbers on the cards are not in numerical order, the following message is output on the 1443 printer.

FORTRAN FILE NUMBERS NOT IN ORDER

the tapes are rewound and the program exits.

When all the files have been converted the following message is output on the 1443 printer.

FORTRAN TAPE TO BCD TAPE CONVERSION FINISHED

the tapes are rewound and the program exits.

If an uncorrectable tape error occurs or a wrong block length specified, an appropriate message is output, the tapes are rewound and the program exits. If an error is detected and corrected, a message is output on the 1443 printer and the program continues.

Input

The fortran tape to be converted must always be on magnetic tape drive 2, with odd parity and 8ØØ bpi density (standard 18ØØ form).

Cards required:-

```
//ƀJOB
//ƀ*PROGECT NO./NAME/TITLE
//ƀXEQ FORBD
*CCEND
```

followed by one header card of the format:-

Card column

1Ø	2Ø	3Ø	4Ø	5Ø
$M_1M_2M_3M_4$	BBBB	CCC	SSSS	NNNN

and then any number of cards of the format:

Card column

1Ø	2Ø	or	1Ø
XXXX	YYYY		XXXX

Where: $M_1M_2M_3M_4$ is the mode of the BCD tape.

M_1 = Ø

M_2 = Ø for odd parity, 1 for even

M_3 = 4 for 8ØØ bpi density

5 for 2ØØ bpi density

6 for 556 bpi density

M_4 = Ø for tape drive 1

All values except for the mode are right justified decimal integers.

BBBB is the blocking factor of the BCD tape, i.e. the number of fortran blocks in one BCD block.

CCC is the number of characters in one fortran block with a maximum number of 146 characters.

BBBB*CCC is the number of characters in 1 BCD block, this number must not exceed 4ØØØ characters (2ØØØ words).

SSSS is the number of the starting file on the BCD tape. All subsequent files are numbered consecutively.

NNNN is the total number of files to be converted from the Fortran tape.

XXXX is the number of the next file to be converted.

YYYY is the number of the last consecutive file to be converted. If YYYY is absent, then file XXXX is copied and the next input card is read, this card may contain XXXX and YYYY or just XXXX. Any number of files (up to 9999) may be copied from one tape.

e.g. If files 1 to 5, 7, 9 and 1Ø are to be converted, then after the header

card,	1Ø	2Ø
first card contains	1	5
2nd " "	7	
3rd " "	9	1Ø

Output

The required files are converted to BCD and written to the tape on drive 1, starting at the file number indicated by SSSS on the header card. The file numbers are printed out on the 1443 printer, after each one has been converted, in the form:-

FORTRAN FILE X COPIED TO BCD FILE S

FORTRAN FILE Y COPIED TO BCD FILE S+1 etc.

Notes

If an error is found after a call to MAGOP the appropriate error nessage is accompanied by the nessage ERROR AT MAGOP CALL XX, where XX=1 when error occurred while reading fortran tape. Where XX=2 while writing to BCD tape. XX=3 while writing tape nark to BCD tape. XX=4 while rewinding BCD tape, and XX=5 while rewinding fortran tape.

NATIONAL INSTITUTE OF OCEANOGRAPHY
DATA PROCESSING GROUP
LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 248

Program Title: BCD Tape To Fortran Tape Conversion
(For Lib. List)

Classification Letter: U

Program Name: BDFOR

Version Number: 1

Type of Program: Non-Process

Operating System: M.P.X.

Language: Fortran IV

Programmer/Date: M.Olliff

Program Passed By: M.Fasham

Description: To convert a tape in BCD to formatted Fortran.

System Prerequisites:

Inskel Common Variables:

Subroutines Called: MAGOP, NEXTF, ZZIPP, BCDA1

Programs Called:

Operation and Method: The BCD tape on drive 1 is positioned at the start of the first file to be converted , the Fortran tape is positioned at the start of the first file to be written. The number of characters to be stored in a Fortran block is calculated, and if this number exceeds 146 the message 'FORTRAN RECORD SIZE TOO LARGE' is output on the 1443 printer and the program exits. BCD blocks are read from tape one at a time. The block is divided into the correct sized blocks for writing to the Fortran tape, converted to Fortran and written to tape, one block at a time, until the number of Fortran blocks written is equal to the blocking factor. If the number of characters to be put into one Fortran block is less than 146, the unused words are left blank. The process is repeated until a complete file has been converted. A tape mark is written to the Fortran tape and the message 'BCD FILE X COPIED TO FORTRAN FILE S' is output on the 1443 printer. All the files are converted in the same way, until the total number of files (as indicated on the header card) have been converted. If the file numbers on the cards are not in numeric order, the following message is output on the 1443 printer 'BCD FILE NUMBERS NOT IN ORDER' the tapes are rewound and the program exits. When all the files have been converted, the following message is output on the 1443 printer 'BCD TAPE TO FORTRAN TAPE CONVERSION FINISHED' , the tapes are rewound and the program exits. If an uncorrectable tape error occurs or a wrong block length specified, the appropriate message is output, the tapes are rewound and the program exits. If an error is detected and corrected, a message is output on the 1443 printer and the program continues.

Inputs: The BCD tape to be converted must always be on magnetic tape drive 1. Cards required:-

```
//ƀJOB
//ƀ*P.No/NAME/TITLE
//ƀXEQƀBDFOR
*CCEND
```

followed by one header card of the format:-

Card Column	10	20	30	40	50
	$M_1M_2M_3M_4$	BBBB	CCCC	SSSS	NNNN

and then any number of oards of the format:-

Card Column	10	20	or	10
	XXXX	$X_1X_1X_1X_1$		XXXX

where

$M_1M_2M_3M_4$ is the mode of the BCD tape M_1 = Ø

M_2 = Ø for odd parity, 1 for even

M_3 = 4 for 8ØØ bpi density
5 for 2ØØ bpi density
6 for 556 bpi density

M_4 = Ø for tape drive 1

All values except for the mode are right justified decimal integers.
BBBB is the blocking factor of the BCD tape i.e. the number of Fortran blocks in one BCD block.
CCCC is the number of characters in one BCD block, maximum number is 4ØØØ characters.
CCCC/BBBB is the number of characters in one Fortran block, this value must not exceed 146 characters.
SSSS is the number of the starting file on the Fortran tape, all subsequent files are numbered consecutively.
NNNN is the total number of files to be converted to Fortran from BCD.
XXXX is the number of the next file to be converted.
$X_1X_1X_1X_1$ is the number of the last consecutive file to be converted. if $X_1X_1X_1X_1$ is absent, then file XXXX is converted and the next input card is read, this card may contain XXXX and $X_1X_1X_1X_1$ or just XXXX. Any number of files (up to 9999) may be copied from one tape.

Output: The required files are converted to Fortran in odd parity, 8ØØ bpi density and written to tape on drive 2, starting at the file number indicated by SSSS on the header card. A list of converted file numbers is printed out on the 1443 printer'BCD FILE X COPIED TO FORTRAN FILE S'.

Notes: If an error is found after a call to MAGOP, the appropriate error message is accompanied by the message 'ERROR AT MAGOP CALL XX'
where XX = 1 when error occurred while reading BCD tape.
XX = 2 while writing to Fortran tape.
XX = 3 while writing tape mark to Fortran tape.
XX = 4 while rewinding Fortran tape.
XX = 5 while rewinding BCD tape.

Execution Time: $(2+F)+(\frac{NOR}{8ØØ}+NOF)$ minutes

where F = number of starting file.
NOR = number of records to be converted.
NOF = number of files to be converted.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program 276

Program Title: (For Lib. List)	OBJPN to PTS conversion
Classification Letter:	U
Program Name:	OBPTS
Version Number:	20
Type of Program:	Nonprocess
Operating System:	MPX or MPXPT
Language:	18ØØ Fortran IV
Programmer/date:	W. Strudwick 16.5.72
Program Passed By:	E. Page
Description:	To convert papertapes in OBJPN format to PTS format
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	PAPTN, TPTAP
Programs Called:	
Files Called:	
Operation and Method:	

The program is invoked by the commands:-

```
//b̸JOB
//b̸XEQb̸OBPTSb̸b̸b̸FX
```

The papertape to be converted is placed in the reader on the first character of the first record (a delete character) and each record is read, converted and output to the punch. The program terminates when data switch zero is put on.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program 279

Program Title: (For Lib. List)	PTTC/8 to ASCII paper tape conversion
Classification Letter:	U
Program Name:	PADPA
Version Number:	20
Type of Program:	Nonprocess program
Operating System:	MPX
Language:	1800 Assembler
Programmer/date:	M. Olliff 11.5.72
Program Passed By:	E. Page
Description:	To convert PTTC/8 paper tape to ASCII code
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	TPTAK, PTASC
Programs Called:	
Files Called:	

Operation and Method:

The program reads in a PTTC/8 coded paper tape, two characters at a time, converts them to ASCII code and punches them out, until the whole tape has been converted. The job should then be aborted. It converts all characters except upper and lower case characters, which are punched out as ASCII deletes, as there are no case characters in ASCII.

Input

```
     // JOB
                  16
    // XEQ PADPA   FX
```

with the paper tape to be converted in the reader.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 280

Program Title: (For Lib. List)	PTS to OBJPN Conversion
Classification Letter:	U
Program Name:	PTSOB
Version Number:	20
Type of Program:	Non-Process
Operating System:	MPX/MPXPT
Language:	1800 FORTRAN IV
Programmer/Date:	W. Strudwick/16.5.72
Program Passed By:	E. Page
Description:	To convert papertapes in PTS format to OBJPN format
System Prerequisites:	None
Inskel Common Variables:	None
Subroutines Called:	PAPTN, TPTAP
Programs Called:	None
Files Called:	None

Operation and Method: The program is invoked by the commands:-

```
//ƀJOB
//ƀXEQƀPTSOBƀƀƀFX
```

The papertape to be converted is placed in the reader on the first character of the first record (a delete character) and each record is read, converted and output to the punch. The program terminates when data switch zero is put on.

N.I.O.SUBPROGRAM -89 CLASSIFICATION U

Title Variable length unpacking of bits in a word

Name NPACK

Language 1800 Fortran IV

Machine IBM 1800

Purpose To extract a group of bits from an integer word and to make a new word from them

Use To extract bits IA to IB inclusive from word IWD and place the result in K:-

K = NPACK(IWD,IA,IB)

For example, if IWD had the following bit pattern:-

$\overset{0}{0}000\overset{4}{1}00\overset{7}{0}0000110\overset{15}{0}$

and the following was executed:-

K = NPACK(IWD,4,7)

Then afterwards IWD would be unchanged and K would have the bit pattern

$000000000000100\overset{15}{0}$

(Bits 4 to 7 of IWD have been placed at the low order end of the word)

Programmer W. Strudwick

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -146

Program Title: (For Lib. List)	Conversion of multiple precision integer to one word integer
Classification Letter:	U
Program Name:	ICONB
Version Number:	1
Type of Program:	Subroutine Sub-program
Operating System:	Any 1800 System
Language:	Fortran
Programmer/date:	J. Crease
Program Passed By:	
Description:	For use with S/R's IADD etc. when doing multiple precision integer arithmetic.
System Prerequisistes:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

CALL ICONB(L,LB,IMAX,IBASE)

LB is an array holding multiple precision integer. It has dimension IMAX and the base of the multiple precision arithmetic is IBASE. See Prog. Description of IADD (-129) for description of the format of LB.

L is an integer variable which after the call holds the value of LB in ordinary integer form.

Note there are no checks on whether LB is too large to be held as a single integer variable.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

NIO PROGRAM - 158

Program Title: (For Lib.List)	Free Format number conversion routine
Classification Letter:	U
Program Name:	NCONV
Version Number:	2∅
Type of Program:	Subroutine
Operating System:	MPX/TSX
Language:	1800 FORTRAN IV
Programmer/date:	Tony Voss/3.11.71
Program Passed By.	C. Spackman
Description:	To extract a number written in free format from a literal field packed in A1 format
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	SKIPD, GET, N2ONE
Programs Called	
Files Called:	

Operation and Method:

CALL NCONV(IMAGE,NS,NL,LOUT,VAL,KODE)

```
On call IMAGE is name of literal array
        NS    is starting point
        NL    is finishing point in array
        LOUT  is output LUN for messages
on return VAL is the value obtained
```

KODE is a return code giving the delimiter found. If negative no number was obtained because an operator was found before a numeral as follows

```
   -5     /
   -4     *
   -3     -     )when used as
   -2     +     ) operator rather than sign
   -1           end of record, &
```

Otherwise a value was obtained, being terminated as given by following value of KODE

```
   ∅      blank or comma
   1      &, RETURN or NEWLINE
```

2	±	not following E
3	*	
4	/	

NS points to the delimiter, except when KODE = ± 1 when NS = Ø

Error Messages

On finding an illegal character i.e. not a numeral, decimal point, E or delimiter NCONV outputs the message

ILLEGAL CHARACTER 'X'

where X is the character found

On finding an illegal format, NCONV outputs

ILLEGAL FORMAT CODE = n

where n has the following meaning

1 Pre-exponent contains more than one decimal point
2 E exponent contains decimal point
3 more than one E

In all cases where an error message is printed, VAL is set to Ø and NS to the next delimiter, zero if NS would be > NL

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -163

Program Title: (For Lib. List)	Free Format Expression Conversion
Classification Letter:	U
Program Name:	FCONV
Version Number:	20
Type of Program:	Subprogram
Operating System:	MPX/TSX
Language:	1800 Fortran IV
Programmer/date:	T. Voss / 3.11.71
Program Passed By:	E. Page
Descritpion:	To evaluate an expression in a literal field, packed in A1 format comprising a number of free format numbers connected by the five fortran operators +,-,*,/,**
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	NCONV, SKIPD and indirectly GET, NZONE
Programs Called:	
Files Called:	
Operation and Method:	

CALL FCONV(IMAGE,NS,NL,LOUT,VAL,KODE)

On call	IMAGE	is name of literal array
	NS	is starting point in array
	NL	is ending point in array
	LOUT	is output LUN for messages
On return	VAL	is value obtained
	KODE	is a return code with the following meaning
		< Ø no number found before NL, RETURN, NEW LINE &
		Ø expression terminated by blank or comma
		> Ø terminated by & , return, or new line
	NS	points to position after delimiter, except when

KODE>∅, when NS=∅.

Nature of Expressions

Numbers may be connected by operators in any reasonable combination. The expression will be evaluated left to right.

Incomplete expression will be evaluated where possible

a) If an expression starts with *,/, or ** a 1 will be assumed missing.

b) If an expression finishes with an operator, blanks or commas will be ignored and the expression connected logically to any other data on the record. If no other data exists on this record, a fresh start will be made on the next record. The exception is that if an E exponent is incomplete it will be ignored.

e.g.

*.5+4	Value	4.5
3*7+ 4*2		5∅
23E+-4		19
23E+ ,-4		23 and -4

Error Messages

On finding an illegal combination of operators i.e. */, FCONV outputs the message

ILLEGAL OPERATOR COMBINATION '*/' etc.

and returns in VAL the value obtained prior to the illegal combination

LEGAL COMBINATIONS	ILLEGAL COMBINATIONS	
*+	*/	***
*-	/*	/**
/*	+*	**/
*-	+/	-**
**+	-*	
**-	-/	
	+**	

Size: 416 words

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -187

Program Title: (For Lib. List)	Maximum or Minimum Integer Value
Classification Letter:	U
Program Name:	MAX/MIN
Version Number:	20
Type of Program:	Function Subroutine
Operating System:	MPX
Language:	Assembler
Programmer/date:	T.Voss/29.2.72
Program Passed By:	E. Page
Description:	Determines maximum or minimum value from a list of integer variables or constants.
System Prerequisites:	Uses MACRO 'RETRN' at assembly i.e. must have library UTMAC available.
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

M = MAX(N,L1,L2,L3,.....)

or

M = MIN(N,L1,L2,L3,.....)

M is set to the maximum or minimum of N arguments L1, L2, L3 etc. N must be ≥ 2 and has virtually no upper limit. It must be constant.

Size 51 words

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -199

Program Title: (For Lib. List)	To convert a real number to characters in a specific format
Classification Letter:	U
Program Name:	CONF
Version Number:	2Ø
Type of Program:	Subroutine
Operating System:	MPX
Language:	Fortran IV
Programmer/date:	M. Olliff 18.5.72
Program Passed By:	R. Wells
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	PUT, NZONE
a	
Programs Called:	
Files Called:	
Operation and Method:	

The subroutine converts a real number into A1 format.

To Use CALL CONF(X,Y,ICHAR)

where X is a real variable containing the no. to be converted

Y is a real variable containing the format specification

ICHAR is an integer array into which the characters are placed.

for example

If X = 12.65 and Y = 6.3, X is converted to EBCDIC characters, and adjusted to have three decimal places and six characters altogether (i.e. the value in Y is treated as the numerical value in the format statement Type F99.9)

The resulting characters are placed in ICHAR as 12.65Ø, one character (left justified) per word.

ICHAR should be defined in the calling program to be large enough to accomodate the characters, maximum size is 20 words.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -201

Program Title: (For Lib. List)	To convert a real number into EBCDIC chars. in E format
Classification Letter:	U
Program Name:	CONE
Version Number:	20
Type of Program:	Subroutine
Operating System:	MPX
Language:	Fortran IV
Programmer/date:	M. Olliff 18.5.72
Program Passed By:	R. Wells
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	LOADF, FBTD, ISLA
Programs Called:	
Files Called:	
Operation and Method:	

The subroutine converts a real number into a specified 'E' type format.

To Use CALL CONE(X,Y,ICHAR)

where X is a real variable to be converted

Y is a real variable indicating the format specification and

ICHAR is an integer array into which the chars. are placed.

for e.g.

If X = 12.65 and Y = 10.3 then the resulting characters in EBCDIC would be ten in number with three decimal places.

So ICHAR would contain +1.265E+Ø1, one char. per word, left justified.

ICHAR should be defined in the calling program to be large enough to acconodate the characters, maximum size is 16 words.

The value in X will be rounded to the no. of decimal places indicated by Y.

N.I.O.Subprogram -79 Classification U

Title	Pause if data switch 0 is on
Name	Subroutine DASWO
Machine	IBM 1800
Type of Program	Nonprocess subroutine
Language	Fortran IV
Purpose	General, but it is recommended to use the subroutine before reading or punching paper tape. This enables tape to be placed in the reader a record at a time, or the punch to be reloaded before it runs out.
Programmer	J. Sherwood
Restriction	Non-process programs only.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -159

Program Title: Change Plotter Pen
(For Lib. List)

Classification Letter: U

Program Name: XPEN

Version Number: 1

Type of program: Subroutine

Operating System: MPX/TSX

Language: 1800 FORTRAN IV

Programmer/date: T.Voss

Program Passed by: C.Spackman

Description: To instruct operator to change plotter pen.

System Prerequisites: V1 not suitable under MPX 1627 spooling

Inskel Common Variables:

Subroutines Called: RAISE

Programs Called: None

Files Called: None

Operation and Method: CALL XPEN (S,K)

where S gives the new pen

greater than 0 Ink pen - size in m.m.
equal to 0 Ball pen
less than 0 Fibre pen

and K is the rquired colour as follows

1	RED
2	GREEN
3	BLUE
4	BLACK
5	BROWN
6	PURPLE

The routine types instructions to the operator (main program must have *IOCS(TYPEWRITER)), raises the pen and waits. After start is pressed the pen is again raised before the routine returns.

If the pen requested is the same as in the last call, the call is ignored.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -162

Program Title: (For Lib. List)	Fast Intra-Core Copy
Classification Letter:	U
Program Name:	COPU/COPYD
Version Number:	20
Type of Program:	Subprogram
Operating System:	M.P.X./T.S.X.
Language:	1130/1800 Assembler
Programmer/date:	T.Voss 3.11.71
Program Passed By:	C. Spackman
Description:	Provides a method of moving data in core, word for word
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	

Operation and Method:

To copy NW words from IF to IT

(i) incrementing upwards

```
CALL COPU(IT,IF,NW)
```

(ii) incrementing downwards

```
CALL COPYD(IT,IF,NW)
```

Uses are best illustrated in examples:

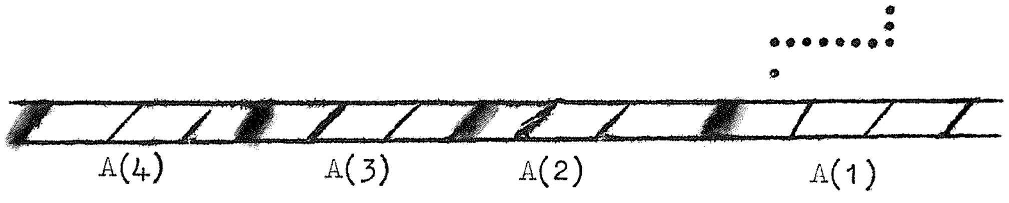
(a) It is required to copy the array J(1ØØ) into the array K(1ØØ)

```
CALL COPYD(K,J,1ØØ)
```

The compiler passes the address of the high core end of K and J. It is therefore necessary to copy downwards.

(b) It is required to copy the array A(5Ø) into the array B(5Ø)

Consideration of the core layout shows that passing A or B as a parameterwould not reference the high-core end of the array but the "high-core end - 2".



It is therefore necessary to pass the low core end address:

```
CALL COPYU (B(5Ø),A(5Ø),15Ø)
```

The 5Ø values (15Ø words with extended precision) will be copied without recourse to the extended precision subroutines as required by statements such as

```
B(N) = A(N)
```

(c) Array shifting: Consider the array J(1ØØ) which contains 8Ø values. It is required to insert an extra value after the 4Øth one

```
CALL COPYD(J(81),J(8Ø),39)
J(41) = NEW
```

(d) A program starts

```
DIMENSION A(5Ø),J(1ØØ),B(5Ø)
```

It is required to temporarily save A into J and B.

```
CALL COPYU(J(1ØØ),A(5Ø),15Ø)
```

(e) Imagine the subroutine

```
CALL SUB(PARM,NTYPE)
```

In which PARM naybe either integer or real and is to be used to return a value. NTYPE is Ø if it is integer and 1 if it is real.

The calls would be

```
CALL SUB(J,Ø)
    or
CALL SUB(F,1)
```

This could be managed in the subroutine thus:

```
     A=ANS
     IF(NTYPE-1)2Ø,3Ø,2Ø
2Ø   CALL COPYU(PARM,IFIX(A),1)
     GO TO 4Ø
3Ø   PARM = A
4Ø   RETURN
```

Checks: Call ignored if NW not positive

Size: 44 words.

Appendix to Progran -162

When COPYU is used in such a way as to propogate an area upwards in core, particular caution should be applied. More explicity, care should be taken if

$$\text{Address}_{\text{TOADR}} - \text{Address}_{\text{FRADR}} \leq \text{NWORD}$$

and

CALL COPYU

The direction of copy (upwards) specifies the direction of propogation of the TOADR and FRADR addresses. The actual transfers are always started at the high-core end e.g. consider the case:

			FRADR					TOADR						
			10	11	12	13	14	15	16	17	18	19	20	

a CALL COPYU(FOADR, FRADR, 7) would cause the value 16 to be noved first and 10 last, to give

10	11	12	13	14	10	11	12	13	14	15	16	

N.I.O. Program 154

Title	Tape copying utility program
Name	TCUP (V2)
Machine	I.B.M. 1800
Language	Assembler
Operating System	T.S.X.
Purpose	To read, compare and punch paper tape
Input	Any program or data on 8-hole paper tape, with the restriction of two deletes being /7F7F
Output	Paper tape copy of the input data

Job Description

```
                          19
// b JOB                  X
// b (Job No./Name/Title)
                        1617
// b XEQ b TCUP         FX
```

A drive Ø disk is used with a drive 1 containing sufficient working storage to store the tape.

Operation

When the program is executed, a general message appears on the typewriter giving the data switch options available.

This is the only message that appears during execution except for a 'tape misread' and 'tape successfully read' message. A tape may be read, reread to check that no errors occurred in the initial reading, and any number of copies punched out. The punched tape may be read to check for any punching errors. A new tape may be read and the procedure continued. If a tape is successfully compared a message is printed after start has been pressed to punch the tape.

The following data switches are used:-

Data Switch 15

Used to read in the paper tape. Also to read in a new paper tape. Put 15 on at beginning of reading the tape, and off when the last character has been read. The data read is stored on disk.

Data Switch 14

Used for rereading a tape, to ensure that the first reading was completed without error (i.e. errors due to the tape reader). If the tape is successfully reread a message is printed after start is pressed for the next operation. However if an error is found the message 'TAPES DO NOT COMPARE' is printed, and the tape may be reread from the beginning again.

Data Switch 12

Use this in conjunction with one of the above. 12 off - the program will stop on deletes. 12 on - the program will stop on any character other than deletes.

By putting 12 on and off at the appropriate time, with 15 or 14 on, the whole of the tape can be read in, or reread quickly. When a pause occurs, press start with the appropriate switches on.

Data Switch 13

Used to punch the tape. As many copies as necessary may be punched.

When the tape is punched it may be read, to check against the original data on disk, to see no errors have occurred. Foe this use data switch 14.
The program may be recycled after punching to read in a new tape.

Data Switch 11

If a tape has benn read in and reread, and you do not wish to punch a copy, but you do wish to either reread the tape again (eg. compare 3 copies of the same tape) or read a new tape in (eg. using the program only to compare sets of tapes) then at the end of the reread, put Data Switch 11 on, and all others off, and press start. A successfully compared message is printed and the system pauses. Then put 11 off, and reset the appropriate switches ie. 15 or 14, and continue.

Method The tape is read, two characters at a time. When 64Ø characters are read, a sector is written to working storage using DRITE.

When the tape is reread, it is checked a sector at a time, and if wrong, stops at the end of the sector in error with a message.

Any number of copies may be punched out, after the initial read, or reread.

Restrictions The program is restricted to the number of sectors of working storage available (64Ø characters = 5ft. 4in. of paper tape = 1 sector).

Execution Time Approximately 1Ø sectors in 3Ø seconds.

Programmer R.Howarth.

National Institute of Oceaongraphy

Data Processing Group

Laboratory Program Description

N.I.O. Program 173

Program Title: (For Lib. List) Edit paper tapes conversationally

Classification Letter: C

Program Name: PUNED

Version Number: 2Ø

Type of Program: Non-Process

Operating System: MPX/MPXIT

Language: 18ØØ Fortran IV

Programmer/date: W. Strudwick / 5.8.72

Program Passed By: E. Page

Description: To copy, skip and alter records on an input tape to produce an output tape

System Prerequisites:

Inskel Common Variables:

Subroutines Called: PAPTN, TYPEN, PAPPR, HOLEB, PLEAD, PLAGG, PAPEB

Programs Called:

Files Called:

Operation and Method: The user has the following options:-

1) To copy records from the input tape to the output tape, type:-
   Cn , where n is a single digit number specifying the number of records to be copied.
2) To skip records on the input tape, type:-
   Sn , where n is a single digit number specifying the number of records to be skipped.
3) To produce a leader tape, type:-
   LE
4) To produce a lagger, type:-
   LA
5) To add a record to the output tape, type:-
   Aþ followed by the record.
6) To exit, type:-
   E

Notes:-

(i) If C or S is specified the last record read is output to the typewriter.

(ii) An option which does not conform to the above mentioned is ignored.

NATIONAL INSTITUTE OF OCEANOGRAPHY
DATA PROCESSING GROUP
LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 229

Program Title: List BCD tapes
(For Lib. List)

Classification Letter: U

Program Name: LIMP

Version Number: 20

Type of Program: Mainline

Operating System: T.S.X.

Language: Fortran

Programmer/Date: W. Strudwick

Program Passed By: E. Page

Description: To list on the lineprinter, files from magnetic tape which have been written in BCD.

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called:

Operation and Method:

Data card as follows:-

1. Card column 1 tape unit number on which tape to be listed is mounted
   = Ø for first unit
   = 1 for second unit

   Card column 3-5 tape density, which can be 556, 800 or 200 bpi.

   Card column 7 tape parity.
   = E for Even
   = O for Odd

   Card column 9-12 block length in characters.

2. Card column 1-6 first file to be listed.
   9-13 last file to be listed.
   (both are integer right justified).
3. More of type 2 if required.
4. Blank card.

Output: File listing on lineprinter.

Use:

```
//ƀJOB
//ƀXEQƀLIMP
*CCEND
Data cards
```

Execution Time: Approximately $(4+F*(\frac{ICHAR*NOFIL}{240*60}))$ mins.

Where F = first file No.

ICHAR = No. of characters/block

NOFIL = No. of files to be listed.

N.I.O. PROGRAM 230 CLASSIFICATION U

Title List FORTRAN formatted magnetic tapes

Name TALIS

Language 1800 Assembler

Machine IBM 1800

Operating System T.S.X.

Purpose To list on the lineprinter files from magnetic tape which have been written using the formatted write statement in 1800 FORTRAN IV

Input Data Cards as follows

1) Card column 1: tape unit number on which tape to be listed is mounted :-
   Ø for the first unit
   1 for second unit

2) Card Column 1 - 6 First file to be listed
   9 - 13 Last file to be listed

   (both are integer right justified)

3) More of type 2 if required

4) Blank card

Output File listing on lineprinter

Use The program is stored on disk 22
To use :

```
//bJOB
//b*PROJECT No./NAME/TITLE
//bXEQbTALIS
*CCEND
Data cards (See under Input)
```

Execution Time Approximately $(2+F+(\frac{NOREC}{240} * NOFIL))$ mins

where F = first file No.

NOREC = average No. of records in files to be listed

NOFIL = No. of files to be listed

Programmer W. Strudwick

N.I.O. PROGRAM 235 CLASSIFICATION U

Title: To compare two program card decks

Name: FCOMP

Language: 1800 Fortran IV

Purpose: To compare two decks of program cards and verify that they are the same.

Subroutines Called: DSLSH, RSLSH

Operating Systems: TSX and MPX

Job Description:

```
                           19
//bJOB                     X

//bXEQbFCOMP

*FILES (1,,1)

*CCEND

PROGRAM DECK 1

=  =

PROGRAM DECK 2

=  =
```

Input: Programs inserted after *CCEND, and each card deck is terminated by = = in Card Column 1 and Card Column 2 card.

Output: This is self explanitory as to the differences in the program card deck.

Method: All control cards, blanks and comment cards are ignored and only program cards are compared. The first card deck is read and any card which does not compare with its corresponding card in the second deck, results in a lineprinter output of the two decks.

Execution Time: To compare two card decks of one thousand cards each the time required is eleven minutes.

Programmer: P. Lindsay.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 253

Program Title: (For Lib. List)	To find size of a BCD magnetic tape block.
Classification Letter:	U
Program Name:	BKSIZ
Version Number:	2Ø
Type of Program:	Non-Process
Operating System:	M.P.X.
Language:	Assembler
Programmer/Date:	D.Brown M.Olliff 10.1.72
Program Passed By:	C.Spackman
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	MAGT, CARDN, PRNTN
Programs Called:	
Files Called:	

Operation and Method: The program reads in a data card giving MAGT parameter and block size in words. A block is read from mag. tape, the correct block size is obtained from the I/O list after the call to MAGT. The data is printed out in hexadecimal form. If the block size given on the data card is too large, only the number of words in the correct block size is printed out. If the block size is too small the number of words in this block size is printed out.

If data Switch Ø is on the tape is backspaced.

If data Switch 15 is on the tape is moved forward.

If data Switch 14 is on the program exits.

The correct blocksize is printed at the start of the printout of each block of data.

Inputs:

```
                   cc19
// JOB                X
              cc16
// XEQ BKSIZ    FX
```

followed by the data card:

```
cc1  5
 XXXXb̸YYYYYY
```

where XXXX is always $2X_1X_2X_3$ for read from mag. tape

X_1 = 1 for even parity, Ø for odd parity.

X_2 = Ø for three bytes/word, 800 b.p.i.

1 " " " " 200 b.p.i.

2 " " " " 556 b.p.i.

4 " two " " 800 b.p.i.

5 " " " " 200 b.p.i.

6 " " " " 556 b.p.i.

X_3 = Ø for 1st. magnetic tape unit.

1 for 2nd. magnetic tape unit.

YYYYYY = blocksize in words (decimal)

Outputs:

Data from magnetic tape to lineprinter in hexadecimal form.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

NIO PROGRAM 254

Program Title: (for Lib.List)	Dump Program From Mag. Tape to Cards and Line Printer
Classification Letter:	U
Program Name:	CDUMP
Version Number:	1
Type of Program:	Non-Process
Operating System:	TSX/SPASM
Language:	1800 Assembler
Programmer/date:	A. Thornborough July 1971
Program Passed By:	Eileen Page
Description:	Program operates under SPASM (see DPG/P/19) to dump a program stored on mag. tape to cards + lineprinter
System Prerequisites:	Cold Start to SPASM
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files called:	
Operation and Method:	To execute the program:

```
//JOB
//XEQ CDUMP
        cc10                 cc73
*SPASM    X                    PROGNAME
*CCEND
Blank cards
```

Where PROGNAME is the name of the program (stored on magnetic tape) to be dumped. The SPASM Executive locates the program PROGNAME on the magnetic tape and switches the main input device of the 1800 from the card read/punch to the magnetic tape. Hence the program is in fact reading from magnetic tape even though it is coded as for the card/punch. For further documentation on this facility please read DPG/P/19 section 8.2.

The program reads each'card' from magnetic tape, punches it onto a card and prints it on 1443 printer. Tests are then made for the statement END in columns 7 - 9 (for a Fortran Program) or columns 27-29 (For an Assembler Program).

When this statement is found the program exits.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 264

Program Title: (For Lib. List)	Correct a binary card
Classification Letter:	U
Program Name:	BREAD
Version Number:	20
Type of Program:	Nonprocess
Operating System:	MPX or MPXPT
Language:	1800 Fortran IV
Programmer/date:	W. Strudwick/26.4.72
Program Passed By:	R. Wells
Description:	To correct a binary card via the keyboard
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	CARDN
Programs Called:	
Files Called:	

Operaticn and Method:

The program is invoked by the commands

```
//ƀJOB
//ƀXEQƀBREADƀƀƀFX
```

the program then reads one binary card.

The message:-

```
COLUMN
```

is then typed and the user may enter via the keyboard the column number in the card he wishes to change. This is entered as a two digit integer, right justified.

The typewriter then responds with the message:- XYØ1234656789 The user then types in under each character in turn either a 1 for a punch or a zero for a no-punch.

Therefore to punch holes in columns X, 2 and 9 type:- 1ØØØ1ØØØØØØ1

The program loops round asking for column numbers and rows to be punched until a column number of zero is typed in.

The program then punches the new card.

NATIONAL INSTITUTE OF OCEANOGRAPHY
DATA PROCESSING GROUP
LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM	-160
Program Title: (For Lib. List)	Search for Literal Characters
Classification Letter:	U
Program Name:	SKIPD
Version Number:	2Ø
Type of Program:	Subroutine
Operating System:	M.P.X./T.S.X.
Language:	18ØØ Fortran IV
Programmer/Date:	T.Voss 3.11.71
Program Passed By:	C.Spackman
Description:	Search For Any One Given List Of Literals
System Prerequisites:	None
Inskel Common Variables:	None
Subroutines Called:	None
Programs Called:	None
Files Called:	None

Operation and Method:

CALL SKIPD(IMAGE,NS,NL,LIST,L,NP,NTYPE,NX)

On call: IMAGE is the A1 literal field.
NS is starting point of search) in IMAGE
NL is the last point of search)
LIST is list of literals in search (A1)
L is length of LIST

On exit: NP points to position in IMAGE for which match was obtained.
NTYPE points to character in LIST which matches IMAGE(NP).
NX =NP+1 except that if no match was obtained NP=NL+1 and NX=Ø.

NX and NP may have the same location.

Inputs:	None
Outputs:	
Size:	106 words

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -164

Program Title: (For Lib. List)	DYNAMIC ALLOCATION OF TIMERS
Classification Letter:	U
Program Name:	LOTIM
Version Number:	V20
Type of Program:	Function Subprogram
Operating System:	MPX (not V1 or 2) Real Time Systems Only
Language:	1800 Assembler
Programmer/date:	T. Voss / 3.12.71
Program Passed By:	E. Page
Description:	Dynamic allocation of MPX software timers for occasional use
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	ITIMR, DELAY, MASK, UNMK
Programs Called:	
Files Called:	

Operation and Method:

The use of timers under MPX is greatly simplified, compared with TSX. Since a larger number of programs will be using timers it is not practical to keep a timer reserved for all users. Occasional users of timers may obtain an allocation by calling J=LOTIM(NDUM) where NDUM is a dummy argument (so that the compiler recognizes a function subprogram) which is not changed. J will be set to the number of a timer which is free. The timer will be reserved until it is used, although if it is not used within a given time the reservation lapses.

Example of Use

It is required to raise an interrupt on level 4 bit 8 after 5 seconds.

```
CALL DELAY(4,8,LOTIM(IDUM),5)
```

Error Message

Should no timer be available the message

```
LOTIM NO TM FREE PNAME LLAA
```

is printed, where PNAME is the load name
LLAA is the level and area of the call

A restart or reload is then performed.

Note

Before loading this program the systems analyst/programmer responsible for the system should set the following constants in LOTIM

RTIM The lowest number timer that may be allocated
(timers ≥ this value should always be used via LOTIM)

LEVL) The level and bit on which an interrupt is to be given if the timer is not
BIT) used within SECS seconds. LEVL and BIT must be valid for the system
SECS) but need not be associated with a program.

NATIONAL INSTITUTE OF OCEANGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -194

Program Title: (For Lib. List) Find a record for a given time in a dynamic data file

Classification Letter: U

Program Name: FINDT

Version Number: 20

Type of Program: Subroutine

Operating System: TSX,MPX

Language: Fortran IV

Programmer/Date: J. Sherwood/Jan 1972

Program Passed By: C. Spackman

Description:

System Prerequisites:

Inskel Common Variables:

Subroutines Called: ITMLM, Fortran Integer Subroutines, IFREC

Programs Called:

Files Called: File is called by symbolic file number provided by calling program

Operation and Method:

CALL FINDT(IFIL,IDAY,IMINT,NREC1,NREC2)

where IFIL is the number in the Define File Card

IDAY day

IMINT time (1/10 minutes)

NREC1 returned number of highest record with time below that requested

NREC2 returned number of lowest record with time above that requested

If a time is found identical to requested time NREC1 and NREC2 are set equal to that time.

If the time is not on the file, NREC1 = NREC2 = Ø

The file must have a standard header

Method: NREC1 and NREC2 are initially set to the Start and End records of the file. After checking that the required time lies on a record between these NREC is set to a record halfway between NREC1 and NREC2. It is then determined whether the required time is before or after NREC. If it is before, NREC2, is set to NREC, if after NREC1 is set to NREC. The process is repeated until NREC1 and NREC2 differ only by 1.

Records with negative day numbers are ignored.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM -198

Program Title: (For Lib. List) Test if DAY1, TIME1 is before DAY2, TIME2

Classification Letter: U

Program Name: ITMLM

Version Number:

Type of Program: Function Subroutine

Operating System: TSX/MPX

Language: Fortran IV

Programmer/date: D. Brown

Program Passed By: E. Page

Description:

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called:

Operation and Method:

The Function ITMLM(IDAY1,ITIM1,IDAY2,ITIM2)

```
returns      ITMLM = -1 IF IDAY2, ITIM2 is before IDAY1, ITIM1
                      0 "    "      "     at       "      "
                     +1 "    "      "     after    "      "
```

IDAY1, IDAY2 are day numbers

ITIM1, ITIM2 are times (1/10th minute)

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. Program -202

Program Title: (For Lib. List)	Load FAC
Classification Letter:	U
Program Name:	LOADF
Version Number:	20
Type of Program:	Subroutine
Operating System:	M.P.X.
Language:	1800 Assembler
Programmer/date:	M. Olliff 18.5.72
Program Passed By:	R. Wells
Description:	
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	ELD
Programs Called:	
Files Called:	

Operation and Method:

The subroutine executes a LIBF to the IBM subroutine ELD to load a real number into the floating point accumulator, FAC. The use of this subroutine enables real numbers to be loaded into FAC from a fortran program. (Useful when using IBM Subroutine FBTD, FLBPE).

To use,

```
      CALL LOADF(X)
```

where X is a real variable containing the number to be loaded.

NATIONAL INSTITUTE OF OCEANOGRAPHY

DATA PROCESSING GROUP

LABORATORY PROGRAM DESCRIPTION

N.I.O. PROGRAM 268

Program Title: Plot Ship's Data Files
(For Lib. List)

Classification Letter: U

Program Name: GRAPH

Version Number: 2Ø

Type of Program: Non-process

Operating System: MPX

Language: 1800 Fortran IV

Programmer/Date: W. Strudwick/27.6.72

Program Passed By: C. Spackman

Description: To plot data from any of the ship's data files with calibration if required.

System Prerequisites:

Inskel Common Variables:

Subroutines Called: DFT, DFTCH

Programs Called: GPLOT

Files Called: User Option

Operation and Method: The program is stored on the DISTRESS disk and is invoked by the following cardsi-

```
cc1          14    19    24
 //ƀJOB       X     X     X
 //ƀXEQƀGRAPHƀƀƀFX1
```

Followed by the following parameter cards:-

(i)

cc

1-2 Number of graphs to be plotted, (maximum of 20).

4-5 Time increment, in minutes, of data to be plotted.

7-9 First day to be plotted.

11-14 Start time for first day, (form HHMM).

16-18 Last day to be plotted.

20-23 Stop time for last day (form HHMM).

25-26 Number of inches along X-axis to represent one day.

28-29 Time interval of tick marks along X-axis (minutes).

31-32 Number of data files to be used.

(All the above values are integers, right justified).

(ii)

cc

1-5

7-11 Names of the files in which the data to be plotted is (must be in chronological
⋮ order), each name left justified in its field.

55-59

(Maximum of ten files)

(iii)

1-6 Minimum value of variable.

8-13 Maximum value of variable. (a value which lies outside this range will be set to the minimum value).

15-16 Total Y-axis length of graph, in inches.

18-23 Distance between tick marks, measured in the units of the variable.

25-30 Distance between Y-axis annotating numbers, measured in the units of the variable.

32-33 Distance from finish of graph to beginning of next graph, in tenths of inches. (Positive for over-plotting, negative for gap).

(All the above are integers, right justified).

35-52 X-axis title.

54-71 Y-axis title.

(iv)

cc

1-2 Word position in record of variable.

Value obtained from disk may be calibrated by either:-

a)

cc

4=1, if variable on disk is a digit value, otherwise blank or zero.

6-15 Calibration a) decimal point anywhere

17-26 Calibration b) in field.

(Calibration of ax+b performed).

or

b) If special S/R for calibration

cc

28=1,2,3,4,5 or 6

(There are six special subprograms on disk called S1,S2,S3,S4,S5 and S6
S/R's S1,S2 and S3 receive the value in the word position specified as an integer.
S/R's S4,S5 and S6 receive an integer array which contains all of the record which has been read from disk.
All subprograms return one real answer. Cards that contain 4,5 or 6 in column 28 must have Ø1 in columns 1 and 2).

cc

30= if computed values are to be printed, else zero.

(v) Type (iii)'s and (iv)'s repeated for as many graphs as are requested.

Notes:-

Program GPLOT does the actual plotting of the variable.

Data is assumed to be on drive 2.

The graphs proceed across the plotter left to right. The variable obtained from disk, is calibrated and then plotted as a function of time. The time increment other than that of the actual time increment on disk may be specified. Thus one may have for example a selective look at half-hourly intervals. Spurious day numbers and times may cause errors.

Special Subprograms

The special subprograms are written by the user for his own application where the standard calibration formula is inapplicable.

The format for S/R's S1, S2 and S3 are as follows:-

```
cc1                    19   24
 //ƀJOB                 X    X
 //ƀFORƀSn
 *LIST ALL
 *EXTENDED PRECISION
      FUNCTION Sn(K)
      calculations on variable K

      RETURN
      END

 //ƀFORƀGPLOTƀZPLOT
                   19  21
 *DELETE            1   GPLOT
 *STORECI           1   GPLOTƀGPLOT
 *CCEND
```

(Note new version of GPLOT with users own S/R Sn, where n is 1,2 or 3, replaces previous version).

The DISTRESS source file disk is required on drive 2 and the DISTRESS program disk on drive 1.

For S/R's 4, 5 and 6 the first statements should be :-

```
SUBROUTINE Sn(IDATA)
INTEGER IDATA(64)
```

If the new program is to be stored under a different name all references to GPLOT should be replaced with the required name.