UNIVERSITY OF
EDINBURGH

Edinburgh
Regional
Computing
Centre

The Edinburgh IMP/FORTRAN System Library Manual

A description of the System Library Routines available in IMP and FORTRAN as implemented by the Edinburgh Regional Computing Centre

First edition
August 1974

The Edinburgh IMP/FORTRAN System Library Manual First Edition

Notes to readers

At a late stage in the production of this manual, a number of errors were detected which it has not been possible to correct in the body of the text. Instead, this errata sheet has been prepared to draw attention to the more important errors. These and other corrections will be incorporated in a subsequent update to the manual. The Centre apologises to readers for the inconvenience caused by these mistakes.

Page 5 The definition of CPUTIM should be as given on Page 14.

11 Line 6. Replace 'Y/X' by 'X/Y'.

21 Line 20 should start 'greater than or equal to 10^{15}'.

26 The parameter for ADDR should be %NAME A.
Alternative parameters for ARRAY are %INTEGER I, %ARRAYFORMAT J.
The 'type' should be %ARRAYMAP.

33 Line 7. Replace 'Y/X' by 'X/Y'.

41 The last line of DIV MATRIX should read:
'DIVMATRIX DATA FAULT N = X
where X is the invalid value of n or m.'

43 Line 22 should read: 'FRACPT (-4.6) is 0.4.'

47 Line 12 should read: 'n is an integer expression indicating the last column of the field.'

48 Line 9 should read: 'a is the name of the 80 element byte integer array in which the information from'.

49 Second line from the bottom should read: 'as INT(X) is interpreted as INTPT(X+0.5).'

50 Fourth line from the bottom. Replace -2^{32} by -2^{31}.

51 Replace last line by 'INVERT DATA FAULT N = n.'

52 Third line from the bottom. Replace 'layout' by 'length'.

56 Line 2. Replace 'A=BC where C' by 'A=BC^T where C^T'.

65 Line 15 should read: 'integer random sequence in the range 0 – $(2^{31}-1)$.'

82 Line 24 should read: 'SOLVE LN EQ DATA FAULT N = n.'

26th May, 1975

CONTENTS

PREFACE

This first edition of the Imp and Fortran System Library Manual defines in detail, the intrinsic and mathematical functions and subprograms available to the user in Fortran and the routines, functions and maps available to him in Imp. As well as detailed specifications of each Imp and Fortran routine, there are two quick reference tables; one preceding each detailed set of specifications.

The final section explains the calling of Imp routines from a Fortran program and Fortran Subroutines from Imp. Further information on the use of these functions can be found in:-

EDINBURGH IMP LANGUAGE MANUAL	(SECOND EDITION JUNE 1974)
and EDINBURGH FORTRAN LANGUAGE MANUAL	(FIFTH EDITION MAY 1974)

Acknowledgements are due to a number of Regional Computing Centre Staff who have contributed to this manual including G E Millard, Miss Dorothy Kidd and Mrs Anne Tweeddale.

Miss Lynda Carlton
August 1974

1

FORTRAN

1.1 REFERENCE TABLE AND INDEX FOR FORTRAN LIBRARY

GENERIC NAME	Function	Entry Name	No. of Arg.	Type of Arguments	Type of Function Value	Page
ABS	Absolute value	IABS ABS DABS CABS CDABS	1 1 1 1 1	Integer*4 Real*4 Real*8 Complex*8 Complex*16	Integer*4 Real*4 Real*8 Real*4 Real*8	6 6 6 6 6
ALOG	Natural Logarithm	ALOG DLOG CLOG CDLOG	1 1 1 1	Real*4 Real*8 Complex*8 Complex*16	Real*4 Real*8 Complex*8 Complex*16	7 7 7 7
ALOG 10	Common Logarithm	ALOG10 DLOG10	1 1	Real*4 Real*8	Real*4 Real*8	8 8
ARCOS	Arccosine	ARCOS DARCOS	1 1	Real*4 Real*8	Real*4 Real*8	9 9
ARSIN	Arcsine	ARSIN DARSIN	1 1	Real*4 Real*8	Real*4 Real*8	10 10
ATAN	Arctangent	ATAN ATAN2 DATAN DATAN2	1 2 1 2	Real*4 Real*4 Real*8 Real*8	Real*4 Real*4 Real*8 Real*8	10 11 10 11
CMPLX	Express two real args in complex form	CMPLX DCMPLX	2 2	Real*4 Real*8	Complex*8 Complex*16	11 11
CONJG	Obtain conjugate of complex Arg.	CONJG DCONJG	1 1	Complex*8 Complex*16	Complex*8 Complex*16	12 12
COS	Trigonometric Cosine (Argument in radians)	COS DCOS CCOS CDCOS	1 1 1 1	Real*4 Real*8 Complex*8 Complex*16	Real*4 Real*8 Complex*8 Complex*16	12 12 12 12
COSH	Hyperbolic Cosine	COSH DCOSH	1 1	Real*4 Real*8	Real*4 Real*8	13 13

GENERIC	Function	Entry Name	No. of Arg.	Type of Arguments	Type of Function Value	Page
COTAN	Trigonometric cotangent (argument in radians)	COTAN DCOTAN	1 1	Real*4 Real*8	Real*4 Real*8	13 13
DBLE	Express a Real *4 argument in Real*8 form	DBLE	1	Real*4	Real*8	14
DIM	Positive difference	DIM IDIM	2 2	Real*4 Integer*4	Real*4 Integer*4	15 15
ERF	Error function	ERF DERF	1 1	Real*4 Real*8	Real*4 Real*8	15 15
ERFC	Complemented error function	ERFC DERFC	1 1	Real*4 Real*8	Real*4 Real*8	16 16
EXP	Exponential	EXP DEXP CEXP CDEXP	1 1 1 1	Real*4 Real*8 Complex*8 Complex*16	Real*4 Real*8 Complex*8 Complex*16	16 16 16 16
FIX	Fix	IFIX HFIX	1 1	Real*4 Real*4	Integer*4 Integer*2	18 18
FLOAT	Float	FLOAT DFLOAT	1 1	Integer*4 Integer*4	Real*4 Real*8	17 17
GAMMA	Gamma	GAMMA DGAMMA	1 1	Real*4 Real*8	Real*4 Real*8	17 17
IMAG	Obtain imaginary part of complex argument	AIMAG	1	Complex*8	Real*4	6
INT	Truncation	INT AINT IDINT	1 1 1	Real*4 Real*4 Real*8	Integer*4 Real*4 Integer*4	18 18 18

GENERIC NAME	Function	Entry Name	No. of Arg.	Type of Arguments	Type of Function Value	Page
LGAMMA	Log-gamma	ALGAMA DLGAMA	1 1	Real*4 Real*8	Real*4 Real*8	7 7
MAX	Largest value	AMAX0 AMAX1 MAX0 MAX1 DMAX1	>=2 >=2 >=2 >=2 >=2	Integer*4 Real*4 Integer*4 Real*4 Real*8	Real*4 Real*4 Integer*4 Integer*4 Real*8	8 8 8 8 8
MIN	Smallest value	AMIN0 AMIN1 MIN0 MIN1 DMIN1	>=2 >=2 >=2 >=2 >=2	Integer*4 Real*4 Integer*4 Real*4 Real*8	Real*4 Real*4 Integer*4 Integer*4 Real*8	9 9 9 9 9
MOD	Modular Arithmetic	MOD AMOD DMOD	2 2 2	Integer*4 Real*4 Real*8	Integer*4 Real*4 Real*8	19 19 19
REAL	Obtain real part of complex arg.	REAL	1	Complex*8	Real*4	19
SIGN	Transfer of sign	SIGN ISIGN DSIGN	2 2 2	Real*4 Integer*4 Real*8	Real*4 Integer*4 Real*8	21 21 21
SIN	Trigonometric Sine (Argument in radians)	SIN DSIN CSIN CDSIN	1 1 1 1	Real*4 Real*8 Complex*8 Complex*16	Real*4 Real*8 Complex*8 Complex*16	21 21 21 21
SINH	Hyperbolic Sine	SINH DSINH	1 1	Real*4 Real*8	Real*4 Real*8	22 22
SNGL	Obtain most SIGNIFICANT PART of Real*8 arg.	SNGL	1	Real*8	Real*4	22

GENERIC NAME	Function	Entry Name	No. of Arg.	Type of Arguments	Type of Function Value	Page
SQRT	Square Root	SQRT DSQRT CSQRT CDSQRT	1 1 1 1	Real*4 Real*8 Complex*8 Complex*16	Real*4 Real*8 Complex*8 Complex*16	23 23 23
TAN	Trigonometric Tangent (Argument in radians)	TAN DTAN	1 1	Real*4 Real*8	Real*4 Real*8	23 23
TANH	Hyperbolic Tangent	TANH DTANH	1 1	Real*4 Real*8	Real*4 Real*8	24 24

/

ADDITIONAL SUPPLIED SUBROUTINES IN EDINBURGH FORTRAN

A number of additional subroutines are available to users of Edinburgh FORTRAN. these are not standard, but provide useful facilities which are often available with minor differences in other implementations of FORTRAN. No special library has to be declared to access any of these routines.

Entry	Definition	No. of Arguments	Type of Argument	Page No.
CPUTIM	CPU time in seconds used by program since first call to routine. (First call returns the value zero	1	Real*8	14
CTIME	Clock time as character string hh:mm:ss	1	Real*8	14
DIAG	Print routine trace back if option NODIAG not coded on EXEC statement	0		15
HDATE	Date as character string dd/mm/yy	1	Real*8	17
LABELS	List 50 most recent STATEMENT LABELS, sub-program entries and returns encountered, if LABELS coded on EXEC statemant	0		18
SETREC	Error recovery can be requested by calling the routine SETREC (usertn), where usertn is the name of the user's error recovery routine	1	Routine name	20

ABS

DEFINITION: This set of functions gives the value of the modulus (i.e. the absolute value) of the quantity specified on entry.

CALL: IABS(X) value is Integer*4, X is an Integer*4 expression
ABS(X) value is Real*4, X is a Real*4 expression
DABS(X) value is Real*8, X is a Real*8 expression
CABS(X) value is Real*4, X is a Complex*8 expression
CDABS(X) value is Real*8, X is a Complex*16 expression

The value of the modulus of X is returned via the function name. For complex arguments X=x+iy the absolute value = SQRT(x^2+y^2)

ERROR CONDITIONS: None

AIMAG

DEFINITION: This function obtains the imaginary part of the Complex quantity specified on entry.

CALL: AIMAG(X) value is Real*4, X is a Complex*8 expression

The imaginary part (i.e. the second word) of the Complex Argument X=x+iy, is returned via the function name.

ERROR CONDITIONS: None

ALGAMA

DEFINITION: These functions give the value of the log-gamma function of the quantity specified on entry.

CALL: ALGAMA(X) value is Real*4, X is a Real*4 expression
DLGAMA(X) value is Real*8, X is a Real*8 expression

The value of the logarithm to the base 'e' of the gamma function of the quantity specified by X is returned via the function name

$\log_e \Gamma$ (X)

ERROR CONDITIONS: If X is less than or equal to zero, or if X is greater than or equal to $4.2937*10^{73}$ the job terminates with the following message:

ARGUMENT NOT WITHIN PERMITTED RANGE
$0.<X<4.2913*10^{73}$

ALOG

DEFINITION: This set of functions gives the value of the logarithm to the base 'e' of the quantity specified on entry.

CALL: ALOG(X) value is Real*4, X is a Real*4 expression
DLOG(X) value is Real*8, X is a Real*8 expression
CLOG(X) value is a Complex*8, X is a Complex*8 expression
CDLOG(X) value is Complex*16, X is a Complex*16 expression

The value of the logarithm to the base 'e' of the quantity specified by X is returned via the function name. For complex arguments $\ln(x+iy)=a+ib$ where $a=\ln|x+iy|$ and b= the principal value of $\arctan(y/x)$.

ERROR CONDITIONS: If X is negative or zero, the job terminates with the following message:

LOG NEGATIVE

This is Run Time Fault 21. This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

ALOG10

DEFINITION: This Real*4 or Real*8 function gives the logarithm to the base 10 of the quantity specified on entry.

CALL: ALOG10(X) value is Real*4, X is a Real*4 expression
DLOG10(X) value is Real*8, X is a Real*8 expression

The value of the logarithm to the base 10 of the quantity specified by X is returned via the function name i.e. $\log_{10}(X)$

ERROR CONDITIONS: If X is negative or zero the job terminates with the following message:

LOG NEGATIVE

This is Run Time Fault 21 This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

AMAXO

DEFINITION: This set of functions gives the value of the largest argument specified on entry in the parameter list.

CALL: AMAX0 (X_1, X_2,...X_n) value is Real*4, X_n are INTEGER*4 expressions
AMAX1 (X_1, X_2,...X_n) value is Real*4, X_n are Real*4 expressions
MAX0 (X_1, X_2,...X_n) value is Integer*4, X_n are Integer*4 expressions
MAX1 (X_1, X_2,...X_n) value is Integer*4, X_n are Real*4 expressions
DMAX1 (X_1, X_2,...X_n) value is Real*8, X_n are Real*8 expressions

The value of the largest argument specified on entry is returned via the function name. i.e. MAX(Arg_1, Arg_2,...Arg_n) (where n >= 2).

ERROR CONDITIONS: None

AMINO

DEFINITION: This set of functions gives the value of the smallest argument specified on entry in the parameter list.

CALL: AMINO (X_1, X_2,...X_n) value is Real*4, X_n are Integer*4 expressions
AMIN1 (X_1, X_2,...X_n) value is Real*4, X_n are Real*4 expressions
MINO (X_1, X_2,...X_n) value is Integer*4, X_n are Integer*4 expressions
MIN1 (X_1, X_2,...X_n) value is Integer*4, X_n are Real*4 expressions
DMIN1 (X_1, X_2,...X_n) value is Real*8, X_n are Real*8 expressions

The value of the smallest argument specified on entry is returned via the function name. i.e. Min(Arg_1, Arg_2,...Arg_n) (where n>= 2).

ERROR CONDITIONS: None

ARCOS

DEFINITION: This Real*4 or Real*8 function finds the value of the angle whose cosine is specified on entry.

CALL: ARCOS(X) value is Real*4, X is a Real*4 expression
DARCOS(X) value is Real*8, X is a Real*8 expression

The value of the angle whose cosine is specified by X is returned via the function name. i.e. arcos(X) The value is given in radians, and lies in the range 0 to + π

ERROR CONDITIONS: If X is not in the range $-1 <= X <= 1$, the job terminates with the following message:

LIBRARY FN FAULT 2

This is Run Time Fault 25
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

ARSIN

DEFINITION: This Real*4 or Real*8 function finds the value of the angle whose sine is specified on entry.

CALL: ARSIN(X) value is Real*4, X is a Real*4 expression
DARSIN(X) value is Real*8, X is a Real*8 expression

The value of the angle whose sine is specified by X is returned via the function name i.e. arcsin (X) . The value is given in radians, and lies in the range $-\pi/2$ to $\pi/2$

ERROR CONDITIONS: If X is not in the range -1 <= X <= 1, the job terminates with the following message:

LIBRARY FN FAULT 1

This is Run Time Fault 25.
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted however, when trapping, that this is a general library function fault and may arise from any one of function calls. Careful programming is therefore required if it may arise from more than one source in a program.

ATAN

DEFINITION: This Real*4 or Real*8 function gives the value of the angle whose tangent is specified on entry.

CALL: ATAN(X) value is Real*4, X is a Real*4 expression
DATAN(X) value is Real*8, X is a Real*8 expression

The value in radians of the angle whose tangent is specified by X is returned via the function name i.e. arctan (X) .

ERROR CONDITIONS: None.

ATAN2

DEFINITION: This Real*4 or Real*8 function gives the value of the angle whose tangent is specified on entry.

CALL: ATAN2(X,Y) value is Real*4, X and Y are Real*4 expressions
DATAN2(X,Y) value is Real*8, X and Y are Real*8 expressions

The value in radians, of the angle whose tangent is specified by 'Y/X', is returned via the function name i.e. arctan (X/Y) . This value lies in the range $-\pi$ to π and is in the first or fourth quadrant if X > 0, and second or third quadrant if X < 0.

ERROR CONDITIONS: If the parameters passed to this routine are both zero, the job terminates with the following message:

```
LIBRARY FN FAULT 3
```

This is Run Time Fault 25
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

CMPLX

DEFINITION: This Complex*8 or Complex*16 function expresses the two real quantities specified on entry, in complex form.

CALL: CMPLX(X,Y) value is Complex*8, X and Y are Real*4 expressions
DCMPLX(X,Y) value is Complex*16, X and Y are Real*8 expressions

The two real arguments specified on entry are expressed in Complex*16 form and returned via the function name.

ERROR CONDITIONS: None.

CONJG

DEFINITION: This Complex*8 or Complex*16 function obtains the conjugate of the complex quantity specified on entry.

CALL: CONJG(X) value is Complex*8, X is a Complex*8 expression
DCONJG(X) value is Complex*16, X is a Complex*16 expression

The conjugate of the complex argument specified on entry is returned via the function name. i.e. Arg = x + i y; Conjugate = x - i y.

ERROR CONDITIONS: None

COS

DEFINITION: This set of functions gives the value of the cosine of the quantity specified on entry. This quantity must be in radians.

CALL: COS(X) value is Real*4, X is a Real*4 expression
DCOS(X) value is Real*8, X is a Real*8 expression
CCOS(X) value is Complex*8, X is a Complex*8 expression
CDCOS(X) value is Complex*16, X is a Complex*16 expression

The value of the cosine of X is returned via the function name.

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15}, the result would be wholly inaccurate, and the job terminates with the following message:

```
TRIG FN INACCURATE
```

This is Run Time Fault 22
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases, until at $|X| = 10^{15}$, the result would be quite meaningless.

COSH

DEFINITION: This Real*4 or Real*8 function gives the value of the hyperbolic cosine of the quantity specified on entry.

CALL: COSH(X) value is Real*4, X is a Real*4 expression
DCOSH(X) value is Real*8, X is a Real*8 expression

The value of the hyperbolic cosine of the quantity specified by X is returned via the function name. i.e. $\frac{1}{2}(e^{x}-e^{-x})$.

ERROR CONDITIONS: If the modulus of X is greater than or equal to 172.694 the job will terminate with the following message:

LIBRARY FN FAULT 5

This is Run Time Fault 25
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in the program.

COTAN

DEFINITION: This Real*4 or Real*8 function gives the value of the cotangent of the quantity specified on entry. This quantity must be in radians.

CALL: COTAN(X) value is Real*4, X is a Real*4 expression
DCOTAN(X) value is Real*8, X is a Real*8 expression

The value of the cotangent of X is returned via the function name. (i.e. COTAN(X) = COS(X)/SIN(X)).

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15} the result would be wholly inaccurate and the job terminates with the following message:

TRIG FN INACCURATE

This is Run Time Fault 22
If the argument passed to this function is near 0 or a multiple of π the job terminates with the following message:

TAN TOO LARGE

This is Run Time Fault 23.
These faults may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

CPUTIM

DEFINITION: This subroutine give the total CPU time used from an arbitrary starting point.

CALL: CPUTIM(X) X is a Real*8 expression

The total CPU time in seconds used since an arbitrary starting point is returned via the parameter list.

ERROR CONDITIONS: None

NOTE: This is an Edinburgh Fortran Option.

CTIME

DEFINITION: This subroutine yields the present time of day.

CALL: CTIME(X) X is a Real*8 expression

The time is returned in the form hh:mm:ss via the parameter list.

ERROR CONDITIONS: None

NOTE: This is an Edinburgh Fortran Option.

DBLE

DEFINITION: This function expresses the Real*4 quantity specified on entry in Real*8 form.

CALL: DBLE(X) value is Real*8, X is a Real*4 expression

The Real*4 argument is expressed in Real*8 form and returned via the function name.

ERROR CONDITIONS: None

DIAG

DEFINITION: This subroutine prints a routine traceback, so long as the option NODIAG is not coded on the EXEC statement of the job.

CALL: DIAG

NOTE: This is an Edinburgh Fortran Option.

DIM

DEFINITION: This Real*4 or Integer*4 function gives the positive difference of the quantities specified on entry.

CALL: DIM(X,Y) value is Real*4, X and Y are Real*4 expressions
IDIM(X,Y) value is Integer*4, X and Y are INTEGER*4 expressions

The value of Argl - Min(argl, arg2) is returned via the function name, where the value of min(argl, arg2) is the smaller of the two parameters.

ERROR CONDITIONS: None

ERF

DEFINITION: This Real*4 or Real*8 function gives the value of the error function of the quantity specified on entry.

CALL: ERF(X) value is Real*4, X is a Real*4 expression
DERF(X) value is Real*8, X is a Real*8 expression

The value of the error function

$$y = {}^{2}/\sqrt{\pi} \int_0^x e^{-u^2} du$$

of the quantity specified by X is returned via the function name.

ERROR CONDITIONS: None

ERFC

DEFINITION: This Real*4 or Real*8 function gives the value of the complemented error function of the quantity specified on entry.

CALL: ERFC(X) value is Real*4, X is a Real*4 expression
DERFC(X) value is Real*8, X is a Real*8 expression

The value of the complemented error function

$$y = {}^2/\sqrt{\pi} \int_x^{\infty} e^{-u^2} du$$

or

$$y = 1-\mathrm{erf}(X)$$

of the quantity specified by X is returned via the function name.

ERROR CONDITIONS: None

EXP

DEFINITION: This set of functions gives the value of 'e' raised to the power of the quantity specified on entry.

CALL: EXP(X) value is Real*4, X is a Real*4 expression
DEXP(X) value is Real*8, X is a Real*8 expression
CEXP(X) value is Complex*8, X is a Complex*8 expression
CDEXP(X) value is Complex*16, X is a Complex*16 expression

The value of 'e' raised to the power X is returned via the function name.
For complex arguments

$$e^{x+iy} = e^x.\cos(y) + i.e^x.\sin(y)$$

ERROR CONDITIONS: If the argument of this function is greater than 172.69, the job terminates with the following message:

EXP TOO LARGE

This is Run Time Fault 24
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

FLOAT

DEFINITION: This Real*4 or Real*8 function converts the integer expression specified on entry to a real value.

CALL: FLOAT(X) value is Real*4, X is an Integer*4 expression
DFLOAT(X) value is Real*8, X is an Integer*4 expression

The real value of the integer expression specified on entry is returned via the function name.

ERROR CONDITIONS: None

GAMMA

DEFINITION: This Real*4 or Real*8 function gives the value of the gamma function of the quantity specified on entry.

CALL: GAMMA(X) value is Real*4, X is a Real*4 expression
DGAMMA(X) value is Real*8, X is a Real*8 expression

The value of the gamma function

$$\Gamma(x)=\int_0^\infty u^{x-1}e^{-u}\,du$$

of the quantity specified by X is returned via the function name.

ERROR CONDITIONS: If X is less than $1*10^{-11}$, or if X is greater than 57, the job terminates with the following message:

```
ARGUMENT NOT WITHIN PERMITTED RANGE:
          1@-11<X<=57.
```

HDATE

DEFINITION: This subroutine yields the current date.

CALL: HDATE(X) X is a Real*8 expression

The date is returned in the form dd/mm/yy via the parameter list.

ERROR CONDITIONS: None

NOTE: This is an Edinburgh Fortran Option.

IFIX

DEFINITION: This Integer*4 or Integer*2 function converts the real expression specified on entry to an integer value.

CALL: IFIX(X) value is Integer*4, X is a Real*4 expression
HFIX(X) value is Integer*2, X is a Real*4 expression

The integer value of the real expression specified on entry is returned via the function name.

ERROR CONDITIONS: None

INT

DEFINITION: This Integer*4 or Real*4 function gives the value of the largest integer less than or equal to the absolute value of the quantity specified on entry. The function takes the same sign as the argument.

```
CALL:  INT(X)   value is Integer*4, X is a Real*4 expression
       AINT(X)  value is Real*4, X is a Real*4 expression
       IDINT(X) value is Integer*4, X is a Real*8 expression
```

The value of the sign of the argument times the largest integer <= |X| is returned via the function name.

ERROR CONDITIONS: None

LABELS

DEFINITION: This subroutine lists the 50 must recent statement labels, subprogram entries and returns encountered, if LABELS is coded on the EXEC statement.

CALL: LABELS

NOTE: This is an Edinburgh Fortran Option.

MOD

DEFINITION: This set of functions gives the value of X(MOD Y) of the quantities specified on entry.

CALL: MOD(X,Y) value is Integer*4, X and Y are Integer*4 expressions
AMOD(X,Y) value is Real*4, X and Y are Real*4 expressions
DMOD(X,Y) value is Real*8, X and Y are Real*8 expressions

The value of Argl (mod Arg2) = Argl -[X]*Arg2 is returned via the function name, where [X] is the largest integer whose magnitude does not exceed the magnitude of Argl/Arg2. The sign of the integer is the same as the sign of Argl/Arg2.

ERROR CONDITIONS: If a division by zero is attempted the job will terminate with the following message:

DIVIDE ERROR

This is Run Time Fault 41 for MOD, Fault 30 for AMOD and Fault 31 for DMOD. These faults may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

REAL

DEFINITION: This Real*4 function obtains the real part of the complex quantity specified on entry.

CALL: REAL(X) value is Real*4, X is a Complex*8 expression

The real part (i.e. the first word) of the Complex Argument X=x+iy is returned via the function name.

ERROR CONDITIONS: None.

SETREC

DEFINITION: This subroutine gives the user the opportunity to recover from certain Fortran errors.

CALL: SETREC (usertn)
usertn is the name of the user's error recovery routine.

A library routine, SETREC, is provided, which the user may call in order to nominate his own recovery routine. After any recoverable error, this user routine is entered with some of the parameters containing diagnostic information. The user subroutine may examine this information to decide whether or not execution should continue with or without diagnostic information first being provided. It sets the remaining parameters, as specified below, prior to return to the calling routine.

Since the parameter 'usertn' in the CALL is a subroutine name, it must be declared as such in the calling program by appearing in an external statement thus:-

```
EXTERNAL usertn
```

The specification of the user's routine has the form

```
SUBROUTINE usertn (errl, err2, exec, diags, dsnum)
```

where the parameters, all integer*4, are:

errl: is the Fortran run time fault number
err2: Additional information
exec: on entry is 0
on return{if 0 then the program stops
if 1 execution continues
diags: on entry is 0
on return{if 0 it requests diagnostics
if 1 no diagnostics
dsnum: is the data set number associated with an input/output error

The action taken on recovering from a particular fault is as detailed in the Edinburgh Fortran Language Manual, Appendix D.

ERROR CONDITIONS: None

NOTE: This is an Edinburgh Fortran Option

SIGN

DEFINITION: This set of functions transfers the sign of the second argument to the modulus of the first.

CALL: SIGN(X,Y) value is Real*4, X and Y are Real*4 expressions
ISIGN(X,Y) value is Integer*4, X and Y are Integer*4 expressions
DSIGN(X,Y) value is Real*8, X and Y are Real*8 expressions

The value of the sign of the second argument times the modulus of the first argument is returned via the function name. i.e. the value returned is the sign of Y*|X|

ERROR CONDITIONS: None

SIN

DEFINITION: This set of functions gives the value of the sine of the quantity specified on entry. This quantity must be in radians.

CALL: SIN(X) value is Real*4, X is a Real*4 expression
DSIN(X) value is Real*8, X is a Real*8 expression
CSIN(X) value is Complex*8, X is a Complex*8 expression
CDSIN(X) value is Complex*16, X is a Complex*16 expression

The value of the sine of X is returned via the function name.

ERROR CONDITIONS: If the modulus of the argument passed to this function is greater than or equal to 10 , the result would be wholly inaccurate, and the job terminates with the following message:

TRIG FN INACCURATE

This is Run Time Fault 22
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases until at $|X|=10^{15}$, the result would be quite meaningless.

SINH

DEFINITION: This Real*4 or Real*8 function gives the value of the hyperbolic sine of the quantity specified on entry.

CALL: SINH(X) value is Real*4, X is a Real*4 expression
DSINH(X) value is Real*8, X is a Real*8 expression

The value of the hyperbolic sine of the quantity specified by X is returned via the function name i.e. $\frac{1}{2}(e^{x}-e^{-x})$

ERROR CONDITIONS: If the modulus of X is greater than 172.694 the job will terminate with the following message:

LIBRARY FN FAULT 4

This is Run Time Fault 25
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC). It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in the program.

SNGL

DEFINITION: This Real*4 function obtains the most significant part of the Real*8 quantity specified on entry.

CALL: SNGL(X) value is Real*4, X is a Real*8 expression

The most significant part of the Real*8 argument is returned via the function name.

ERROR CONDITIONS: None

SQRT

DEFINITION: This set of functions gives the value of the positive square root of the quantity specified on entry.

CALL: SQRT(X) value is Real*4, X is a Real*4 expression
DSQRT(X) value is Real*8, X is a real*8 expression
CSQRT(X) value is Complex*8, X is a Complex*8 expression
CDSQRT(X) value is Complex*16, X is a Complex*16 expression

The value of the positive square root of the quantity specified by X is returned via the function name. i.e. SQRT(X)

ERROR CONDITIONS: If a negative argument is passed to this routine the job terminates with the following message:

SQRT NEGATIVE

This is Run Time Fault 20
This fault may be trapped by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

TAN

DEFINITION: This Real*4 or Real*8 function gives the value of the tangent of the quantity specified on entry. This quantity must be in radians.

CALL: TAN(X) value is Real*4, X is a REAL*4 expression
DTAN(X) value is Real*8, X is a Real*8 expression

The value of the tangent of X is returned via the function name.

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15} the result would be wholly inaccurate and the job terminates with the following message:

TRIG FN INACCURATE

This is Run Time Fault 22
It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases, until at $|X|=10^{15}$, the result would be quite meaningless. If the value of COS(X) is less than or equal to 10^{-14} then the job terminates with the message:

TAN TOO LARGE

This is Run Time Fault 23.
Both these faults are trappable by making a call on the Edinburgh Fortran error recovery routine (see SETREC).

TANH

DEFINITION: This Real*4 or Real*8 function gives the value of the hyperbolic tangent of the quantity specified on entry.

CALL: TANH(X) value is Real*4, X is a Real*4 expression
DTANH(X) value is Real*8, X is a Real*8 expression

The value of the hyperbolic tangent of the quantity specified by X is returned via the function name.

ERROR CONDITIONS: None

2

IMP

2.1 INTRINSIC, IMPLICIT, EXPLICIT Functions in IMP

Routines, functions and maps are divided into three classes INTRINSIC, IMPLICIT and EXPLICIT. Items in the first two classes can be used without explicit declaration since their names and characteristics are known to the compiler. Items in the EXPLICIT class, however, must be specified before they are used. The specification provides information to the compiler as to the name and parameter list of the routine, function or map. It also causes the compiler to generate an entry in a table to ensure that the necessary module is loaded when the program is run. It is most important to type the specification accurately, in particular the number and types of parameter must be correct (the names of parameters used are not significant).

The specification of items in this class are given in the following sections where required.

2.2 REFERENCE TABLE AND INDEX FOR IMP LIBRARY

NAME	TYPE	CLASS	PARAMETERS	PAGE NO.
ADD MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B,C, %INTEGER I,J	30
ADDR	%INTEGERFN	INTRINSIC	%INTEGER I	30
ARCCOS	%LONGREALFN	IMPLICIT	%LONGREAL A	31
ARCSIN	%LONGREALFN	IMPLICIT	%LONGREAL A	32
ARCTAN	%LONGREALFN	IMPLICIT	%LONGREAL A,B	33
ARRAY	%ARRAY MAP	INTRINSIC	%INTEGER I,%ARRAYNAME J	34
BITS	%INTEGERFN	EXPLICIT	%INTEGER I	34
BYTE INTEGER	%BYTEINTEGERMAP	INTRINSIC	%INTEGER I	35
CHARNO	%BYTEINTEGERMAP	INTRINSIC	%STRINGNAME S, %INTEGER I	35
CLOSE DA	%ROUTINE	EXPLICIT	%INTEGER I	36
CLOSE SQ	%ROUTINE	EXPLICIT	%INTEGER I	36
CLOSE STREAM	%ROUTINE	IMPLICIT	%INTEGER I	37
COPY MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B, %INTEGER I,J	37
COS	%LONGREALFN	IMPLICIT	%LONGREAL A	38
COT	%LONGREALFN	IMPLICIT	%LONGREAL A	39
CPUTIME	%LONGREALFN	EXPLICIT	NONE	39
DATE	%STRINGFN	EXPLICIT	NONE	40
DET	%LONGREALFN	EXPLICIT	%LONGREALARRAYNAME A, %INTEGER I	40
DIV MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B, %INTEGER I,J, %LONGREALNAME C	41
ERFN	%LONGREALFN	EXPLICIT	%LONGREAL A	41
ERFNC	%LONGREALFN	EXPLICIT	%LONGREAL A	42
EXP	%LONGREALFN	IMPLICIT	%LONGREAL A	42
EXP TEN	%LONGREALFN	EXPLICIT	%LONGREAL A	43
FRAC PT	%LONGREALFN	INTRINSIC	%LONGREAL A	43

NAME	TYPE	CLASS	PARAMETERS	PAGE NO.
FROM STRING	%STRINGFN	IMPLICIT	%STRINGNAME S,%INTEGER I,J	44
GAMMAFN	%LONGREALFN	EXPLICIT	%LONGREAL A	44
HYPCOS	%LONGREALFN	EXPLICIT	%LONGREAL A	45
HYPSIN	%LONGREALFN	EXPLICIT	%LONGREAL A	46
HYPTAN	%LONGREALFN	EXPLICIT	%LONGREAL A	46
IFD BINARY	%INTEGERFN	EXPLICIT	%SHORTINTEGERARRAYNAME I, %INTEGER J,K,%INTEGERNAME L	47
IFD ISO	%INTEGERFN	EXPLICIT	%BYTEINTEGERARRAYNAME I, %INTEGER J,K,%INTEGERNAME L	48
IMOD	%INTEGERFN	INTRINSIC	%INTEGER I	49
INT	%INTEGERFN	INTRINSIC	%LONGREAL A	49
INTEGER	%INTEGERMAP	INTRINSIC	%INTEGER I	50
INT PT	%INTEGERFN	INTRINSIC	%LONGREAL A	50
INVERT	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B, %INTEGER I, %LONGREALNAME J	51
ISO CARD	%ROUTINE	EXPLICIT	%BYTEINTEGERARRAYNAME K	52
LENGTH	%BYTEINTEGERMAP	INTRINSIC	%STRINGNAME S	52
LOG	%LONGREALFN	IMPLICIT	%LONGREAL A	53
LOGGAMMA	%LONGREALFN	EXPLICIT	%LONGREAL A	53
LOGTEN	%LONGREALFN	EXPLICIT	%LONGREAL A	54
LONG REAL	%LONGREALMAP	INTRINSIC	%INTEGER I	54
MOD	%LONGREALFN	INTRINSIC	%LONGREAL A	55
MULT MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B,C, %INTEGER I,J,K	55
MULT TR MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B,C, %INTEGER I,J,K	56
NEWLINE	%ROUTINE	INTRINSIC	NONE	56
NEWLINES	%ROUTINE	INTRINSIC	%INTEGER I	57
NEWPAGE	%ROUTINE	INTRINSIC	NONE	57
NEXT ITEM	%STRINGFN	INTRINSIC	NONE	58
NEXT SYMBOL	%INTEGERFN	INTRINSIC	NONE	58

NAME	TYPE	CLASS	PARAMETERS	PAGE NO.
NL	%INTEGERFN	INTRINSIC	NONE	59
NULL	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A, %INTEGER I,J	59
OPEN DA	%ROUTINE	EXPLICIT	%INTEGER I	60
OPEN SQ	%ROUTINE	EXPLICIT	%INTEGER I	61
PI	%LONGREALFN	INTRINSIC	NONE	61
PRINT	%ROUTINE	IMPLICIT	%LONGREAL A,%INTEGER I,J	62
PRINT CH	%ROUTINE	INTRINSIC	%INTEGER I	62
PRINT FL	%ROUTINE	IMPLICIT	%LONGREAL A,%INTEGER I	63
PRINT STRING	%ROUTINE	INTRINSIC	%STRING S	63
PRINT SYMBOL	%ROUTINE	INTRINSIC	%INTEGER I	64
RADIUS	%LONGREALFN	IMPLICIT	%LONGREAL A,B	64
RANDOM	%REALFN	EXPLICIT	%INTEGERNAME I,%INTEGER K	65
READ	%ROUTINE	IMPLICIT	%NAME A	66
READ CH	%ROUTINE	INTRINSIC	%NAME I	67
READ DA	%ROUTINE	EXPLICIT	%INTEGER I,%INTEGERNAME J, %NAME K,L	68
READ ITEM	%ROUTINE	INTRINSIC	%STRINGNAME S	69
READ SQ	%ROUTINE	EXPLICIT	%INTEGER I,%NAME J,K	70
READ STRING	%ROUTINE	IMPLICIT	%STRINGNAME S	71
READ SYMBOL	%ROUTINE	INTRINSIC	%NAME I	72
REAL	%REALMAP	INTRINSIC	%INTEGER I	73
RECORD	%RECORDMAP	INTRINSIC	%INTEGER I	74
RFD BINARY	%LONGREALFN	EXPLICIT	%SHORTINTEGERARRAYNAME I, %INTEGER J,K,%INTEGERNAME L	75
RFD ISO	%LONGREALFN	EXPLICIT	%BYTEINTEGERARRAYNAME I, %INTEGER J,K,%INTEGERNAME L	76
SELECT INPUT	%ROUTINE	INTRINSIC	%INTEGER I	77
SELECT OUTPUT	%ROUTINE	INTRINSIC	%INTEGER I	78
SET MARGINS	%ROUTINE	IMPLICIT	%INTEGER I,J,K	79

NAME	TYPE	CLASS	PARAMETERS	PAGE NO.
SHIFT C	%INTEGERFN	EXPLICIT	%INTEGER I,J	80
SHORT INTEGER	%SHORTINTEGERMAP	INTRINSIC	%INTEGER I	80
SIN	%LONGREALFN	IMPLICIT	%LONGREAL A	81
SKIP SYMBOL	%ROUTINE	INTRINSIC	NONE	81
SOLVE LN EQ	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B, %INTEGER I,%LONGREALNAME C	82
SPACE	%ROUTINE	INTRINSIC	NONE	82
SPACES	%ROUTINE	INTRINSIC	%INTEGER I	83
SQRT	%LONGREALFN	IMPLICIT	%LONGREAL A	83
STRING	%STRINGMAP	INTRINSIC	%INTEGER I	84
SUB MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B,C, %INTEGER I,J	85
TAN	%LONGREALFN	IMPLICIT	%LONGREAL A	86
TIME	%STRINGFN	EXPLICIT	NONE	86
TOSTRING	%STRINGFN	INTRINSIC	%INTEGER I	87
TRANS MATRIX	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A,B, %INTEGER I,J	87
UNIT	%ROUTINE	EXPLICIT	%LONGREALARRAYNAME A, %INTEGER I	88
WRITE	%ROUTINE	INTRINSIC	%INTEGER I,J	88
WRITE DA	%ROUTINE	EXPLICIT	%INTEGER I,%INTEGERNAME J, %NAME K,L	89
WRITE SQ	%ROUTINE	EXPLICIT	%INTEGER I,%NAME J,K	90

ADD MATRIX

DEFINITION: This routine adds two NxM matrices.

SPEC: %EXTERNALROUTINESPEC ADD MATRIX (%LONGREALARRAYNAME a,b,c, %INTEGER n,m)

CALL: ADD MATRIX (a,b,c,n,m)

a,b,c are the names of three longreal two-dimensional arrays. Arrays b and c contain the matrices to be added.

n,m are the integer expressions whose values, N and M, give upper bounds for the dimensions of a,b and c, the lower bounds being taken to be 1. Array a contains the matrix sum of matrices in arrays b and c. Arrays b and c are not corrupted.

ERROR CONDITIONS: If the array bounds of any of the matrices are zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

ADDR

DEFINITION: This function gives the absolute address of the %NAME type parameter specified on entry.

SPEC: Not required

CALL: ADDR(x)

x is the name of a variable whose address is to be returned.

The absolute address of x is returned via %RESULT.

ERROR CONDITIONS: None.

ARCCOS

DEFINITION: This longreal function finds the value of the angle whose cosine is specified on entry.

SPEC: Not required

CALL: ARCCOS(x)

x is a real or longreal expression.

The value of the angle whose cosine is specified by x is returned via %RESULT.
The value is given in radians, and lies in the range 0 to + π

ERROR CONDITIONS: If x is not in the range -1<=x<=1, the job terminates with the following message:

LIBRARY FN FAULT 2

This fault is trappable as FAULT 25.
It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

ARCSIN

DEFINITION: This longreal function finds the value of the angle whose sine is specified on entry.

SPEC: Not required

CALL: ARCSIN(x)

x is a real or longreal expression.

The value of the angle whose sine is specified by x is returned via %RESULT.
The value is given in radians, and lies in the range -π/2 to +π/2.

ERROR CONDITIONS: If x is not in the range -1<=x<=1, the job terminates with the following message:

LIBRARY FN FAULT 1

This fault is trappable as FAULT 25.
It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

ARCTAN

DEFINITION: This longreal function gives the value of the angle whose tangent is specified on entry.

SPEC: Not required

CALL: ARCTAN(x,y)

x,y are real or longreal expressions.

The value in radians, of the angle whose tangent is specified by 'y/x', is returned via %RESULT. This value lies in the range $-\pi$ to $+\pi$, and is in the first or fourth quadrant if x>0, and second or third quandrant if x<0.

ERROR CONDITIONS: If the parameters passed to this routine are both zero, the job terminates with the following message:

LIBRARY FN FAULT 3

This fault is trappable as FAULT 25.
It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

ARRAY

DEFINITION: This map can be used to map an array onto an area specified by the address of its start.

SPEC: Not required

CALL: ARRAY(I,J)

I is an integer expression specifying an address
J is the name of an %ARRAYFORMAT

The %ARRAYFORMAT statement is used to describe the characteristics of the array which is being mapped. As an alternative to using the name of an %ARRAYFORMAT for the second parameter, the name of another %ARRAY can be used, if one with suitable characteristics has been defined in the program.

In the following example the two-dimensional array ATWO is mapped on to an array AONE which is declared as a one- dimensional array:

```
%INTEGERARRAY AONE (1:10000)
%INTEGERARRAYNAME ATWO
%INTEGERARRAYFORMAT AFORM (1:100,1:100) ; !This  statement  describes
                                           the characteristics of the
                                           array ATWO
   ATWO==ARRAY(ADDR(AONE(1)),AFORM)
   ATWO(27,27)=928
```

ERROR CONDITIONS: None.

BITS

DEFINITION: This function finds the number of non-zero bits in the binary representation of a specified integer.

SPEC: %EXTERNALINTEGERFNSPEC BITS (%INTEGER n)

CALL: BITS(n)

n is the integer expression whose binary representation is to be examined.

The number of non-zero bits of n is returned via %RESULT.

ERROR CONDITIONS: None

BYTE INTEGER

DEFINITION: This map enables the user to acess a particular byte location whose address is specified as the parameter.

SPEC: Not required

CALL: BYTE INTEGER(n)

n is an integer expression giving the absolute address in core of the required location.

This map may be used to access the contents of a byte integer location or to store a value at the specified address:

```
%INTEGER I
%BYTEINTEGER J
I=X'12345678'                  ; !I is an integer containing a 4  byte
                                 binary pattern
J=BYTEINTEGER (ADDR(I)+1)      ; !J has the value X'34' and I is
                                 unchanged
BYTEINTEGER (ADDR(I)+3)=X'9A'  ; !I now contains the binary pattern
                                 X'1236569A' and J is unchanged
```

ERROR CONDITIONS: None.

CHARNO

DEFINITION: This map enables the user to access the ISO-numeric value of a character of a string.

SPEC: Not required

CALL: CHARNO(s,n)

s is the name (of type %STRING) of the location containing the string.

n is an integer expression indicating which character of the string is to be accessed.

ERROR CONDITIONS: The result is undefined unless 1<=n<=LENGTH(s).

CLOSEDA

DEFINITION: This routine closes the direct access file on the channel specified on entry.

SPEC: %EXTERNALROUTINESPEC CLOSEDA (%INTEGER n)

CALL: CLOSEDA(n)

n   is an integer expression which gives the channel number which defines the file, and whose value must lie between 1 and 80 inclusive.

The file is closed and may not be used again without re-opening.

ERROR CONDITIONS: If a file is closed twice or an attempt is made to close an unopened file, the job terminates with an appropriate message.

CLOSESQ

DEFINITION: The routine closes the sequential file on the channel specified on entry.

SPEC: %EXTERNALROUTINESPEC CLOSESQ (%INTEGER m)

CALL: CLOSESQ(m)

m   is an integer expression which specifies the channel number, and whose value must lie between 1 and 80 inclusive.

The file is closed and may not be used again without re-opening.

ERROR CONDITIONS: If a file is closed twice, an attempt is made to close a file which has not been opened, or an attempt is made to close a file which has been neither opened nor defined, the job terminates with an appropriate message.

CLOSE STREAM

DEFINITION: This routine closes a user-defined stream.

SPEC: Not required

CALL: CLOSE STREAM(n)

n is an integer expression giving the number of the stream to be closed.

The stream n is closed such that on the next call of either SELECT INPUT or SELECT OUTPUT for that stream, the stream is reset to the start. Note that n must not be the current input or output stream.

ERROR CONDITIONS: If the stream is not defined on entry to the routine, if n>80 or n<1, or if n is a current stream, then the job terminates with an appropriate message.

COPY MATRIX

DEFINITION: An NxM matrix is copied from one two-dimensional array onto another.

SPEC: %EXTERNALROUTINESPEC COPY MATRIX(%LONGREALARRAYNAME a,b, %INTEGER n,m)

CALL: COPY MATRIX (a,b,n,m)

a,b	are the names of two longreal two-dimensional arrays. Array b contains the rectangular matrix which is to be copied.
n,m	are the integer expressions whose values, N and M, give the upper limits for the dimensions of the given arrays, the lower limits being taken to be 1.

Array a contains a matrix identical to the matrix contained in b. Array b is unchanged.

ERROR CONDITIONS: If n or m is negative or zero, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

COS

DEFINITION: This longreal function gives the value of the cosine of the quantity specified on entry. This quantity must be in radians.

SPEC: Not required

CALL: COS(x)

x is a real or longreal expression.

The value of the cosine of x is returned via %RESULT.

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15}, the result would be wholly inaccurate, and the job terminates with the following message:

TRIG FN INACCURATE

This fault is trappable as FAULT 22.
It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases, until, at $|x|=10^{15}$, the result would be quite meaningless.

COT

DEFINITION: This real or longreal function gives the value of the cotangent of the quantity specified on entry. This quantity must be in radians.

SPEC: Not required

CALL: COT(x)

x is a real or longreal expression.

The real or longreal value of the cotangent of x is returned via %RESULT (cotan x=cos x/sin x).

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15} the result would be wholly inaccurate and the job terminates with the following message:

TRIG FN INACCURATE

This fault is trappable as Fault 22.
If the argument passed to this function is near 0 or a multiple of π, the job terminates with the following message:

TAN TOO LARGE

This fault is not trappable.

CPUTIME

DEFINITION: This function gives the total CPU time used from an arbitrary starting point.

SPEC: %EXTERNALLONGREALFNSPEC CPUTIME

CALL: CPUTIME

No parameters

The total CPU time used since an arbitrary starting point is returned via %RESULT as a positive long real number having seconds as its unit of time. Note that there is a CPU time overhead in calling the function itself.

ERROR CONDITIONS: None.

DATE

DEFINITION: This function yields the current date.

SPEC: %EXTERNALSTRINGFNSPEC DATE

CALL: DATE

No parameters.

The date is returned in the form dd/mm/yy via %RESULT, and may be printed out using PRINTSTRING.

ERROR CONDITIONS: None.

DET

DEFINITION: This function evaluates the determinant of an NxN matrix. The method of Gaussian elimination with partial pivoting is used. This is implemented using the routine SOLVE LN EQ.

SPEC: %EXTERNALLONGREALFNSPEC DET (%LONGREALARRAYNAME a, %INTEGER n)

CALL: DET (a,n)

a is the name of a longreal two-dimensional array containing the square matrix whose determinant is required.
Note that the contents of this matrix are destroyed.

n is the integer expression whose value N, gives the upper limit for the dimensions of a, the lower limits taken to be 1.

The value of the determinant is returned via %RESULT. Note that large errors may occur if the matrix is ill-conditioned, and if this is suspected, perturbed matrices should be examined.

ERROR CONDITIONS: If the matrix bound n is zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

DIV MATRIX

DEFINITION: Given an NxN matrix B and an NxM matrix A, the routine generates INV(B).A, an NxM matrix. Both B and A are destroyed.

SPEC:
```
%EXTERNALROUTINESPEC DIV MATRIX (%LONGREALARRAYNAME a,b,
                                 %INTEGER n,m, %LONGREALNAME det)
```

CALL: DIV MATRIX (a,b,n,m,det)

a,b are the names of two two-dimensional longreal arrays. Array a contains rectangular matrix A on input. Array b contains square matrix B on input.

n,m are the integer expressions whose values, N and M, give the upper limits for the dimensions of the given arrays, the lower limits being taken to be 1.)

det is the name of a longreal variable which is set by the routine to the value of the determinant of the matrix in array b.

The result of the division is placed in array a and matrix B is destroyed.
The value of the determinant of B is placed in det.
Note that large errors may occur if the matrix B is ill-conditioned, and if this is suspected, perturbed matrices should be examined.

ERROR CONDITIONS: If n or m is less than, or equal to zero, the job terminates with the following message:

DIV MATRIX DATA FAULT N = X

ERFN

DEFINITION: This real or longreal function gives the value of the error function of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC ERFN(%LONGREAL x)

CALL: ERFN(x)

x is a real or longreal expression.

The real or longreal value of the error function

$$y = 2/\sqrt{\pi} \int_0^x e^{-u^2} du$$

of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: None.

ERFNC

DEFINITION: This real or longreal function gives the value of the complemented error function of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC ERFNC (%LONGREAL x)

CALL: ERFNC(x)

x is a real or longreal expression

The real or longreal value of the complemented error function

$$y = 2/\sqrt{\pi} \int_{x}^{\infty} e^{-u^2} du$$

or

y = 1 - erfn(x)

of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: None.

EXP

DEFINITION: This real or longreal function gives the value of 'e' raised to the power of the quantity specified on entry.

SPEC: Not required

CALL: EXP(x)

x is a real or longreal expression.

The value of 'e' raised to the power x is returned via %RESULT.

ERROR CONDITIONS: If the argument of this function is greater than 172.69, the job terminates with the following message:

EXP TOO LARGE

This fault is trappable as FAULT 24.

EXPTEN

DEFINITION: This real or longreal function gives the value of 10 raised to the power of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC EXPTEN (%LONGREAL x)

CALL: EXPTEN(x)

x is a real or longreal expression.

The value of 10 raised to the power x is returned via %RESULT.

ERROR CONDITIONS: If the argument of this function is greater than 75.0 then the job terminates with the following message:

EXP TOO LARGE

This fault is trappable as FAULT 24.

FRAC PT

DEFINITION: This function gives the value of the fractional part of the quantity specified on entry.

SPEC: Not required

CALL: FRAC PT(x)

x is a longreal expression of which the fractional part is to be returned.

The fractional part of x is returned as a long real via %RESULT. If |x| exceeds 2^{55} -1, the value returned is always zero. The fractional part is always located as being greater than or equal to zero, hence FRACPT(i.e. -4.6) is 0.4.

ERROR CONDITIONS: None.

FROM STRING

DEFINITION: This function extracts characters from a string.

SPEC: Not required

CALL: FROM STRING (s,i,j)

s is the name (of type %STRING) of the location in which the string is stored.

i,j are integer expressions giving the lower and upper boundaries (inclusive) of the character string to be extracted.

The result is a string of characters (from i to j, inclusive) extracted from the string contained in s. s is unchanged.

ERROR CONDITIONS: Unless 1<=i<=j<=length(s), the job terminates with the following message:

STRING INSIDE OUT

GAMMAFN

DEFINITION: This longreal function gives the value of the gamma function of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC GAMMAFN(%LONGREAL x)

CALL: GAMMAFN(x)

x is a real or longreal expression

The real or longreal value of the gamma function

$$\Gamma(x) = \int_0^{\infty} u^{x-1} e^{-u} du$$

of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If x is less than $1*10^{-11}$, or if x is greater than 57, the job terminates with the following message:

ARGUMENT NOT WITHIN PERMITTED RANGE:
1@-11 < x <= 57.

HYPCOS

DEFINITION: This real or longreal function gives the value of the hyperbolic cosine of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC HYPCOS(%LONGREAL x)

CALL: HYPCOS(x)

x is a real or longreal expression

The real or longreal value of the hyperbolic cosine

$y = \frac{1}{2}(e^{x} + e^{-x})$

of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If the modulus of x is greater than or equal to 172.694 the job will terminate with the following message:

LIBRARY FN FAULT 5

This fault is trappable as Fault 25.
It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

HYPSIN

DEFINITION: This real or longreal function gives the values of the hyperbolic sine of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC HYPSIN(%LONGREAL x)

CALL: HYPSIN(x)

x is a real or longreal expression.

The real or longreal value of the hyperbolic sine

$$y = \tfrac{1}{2}(e^{x} - e^{-x})$$

of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If the modulus of x is greater than or equal to 172.694 the job will terminate with the following message:

LIBRARY FN FAULT 4

This fault is trappable as Fault 25.
It should be noted, however, when trapping, that this is a general library function fault and may arise from any one of several different function calls. Careful programming is therefore required if it may arise from more than one source in a program.

HYPTAN

DEFINITION: This real or longreal function gives the value of the hyperbolic tangent of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC HYPTAN(%LONGREAL x)

CALL: HYPTAN(x)

x is a real or longreal expression.

The real or longreal value of the hyperbolic tangent of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: None.

IFD BINARY

DEFINITION: This function interprets a card which has been stored as column binary in the array specified by the first input parameter to this routine. It evaluates the integer formed from the digits occurring in the field specified by the second and third input parameters.

```
SPEC:  %EXTERNALINTEGERFNSPEC IFD BINARY(%SHORTINTEGERARRAYNAME a,
                                         %INTEGER m,n,%INTEGERNAME x)
```

```
CALL:  IFD BINARY(a,m,n,x)
```

a is the name of the 80 element short integer array in which the information from the card has been stored.
m is an integer expression indicating the first column of the field.
n is an integer indicating the last column of the field.
x is the name of an integer variable which is used as an error indicator by this routine.

The integer formed from the digits occurring in the field specified from column 'm' to column 'n' is returned via %RESULT.

Note that the only legal punchings in the field specified by the input parameters in 'm' and 'n' are:

space(blank),0-9,+,-

ERROR CONDITIONS: This routine signals errors by setting a value in the error indicator specified on entry. In the event of error, the function assumes the value of the number of the column (i.e. the subscript of the array element) containing the illegal symbol.

The values of the error indicator are as follows:

0 the field contains a legal integer constant.
1 the field contains no digits, or spaces occur after the leading digit.
2 the field contains an illegal symbol.
3 the first column specified occurs after the last; in this case, the function takes the value zero.

IFD ISO

DEFINITION: This function interprets a card which has been stored in the array specified by the first input parameter to this routine. It evaluates the integer formed from the digits occurring in the field specified by the second and third input parameters.

SPEC:
```
%EXTERNALINTEGERFNSPEC IFDISO(%BYTEINTEGERARRAYNAME a,
                              %INTEGER m,n,%INTEGERNAME x)
```

CALL: IFD ISO(a,m,n,x)

a — is the name of the 80 byte integer in which the information from the card has been stored.

m — is an integer expression indicating the first column of the field.

n — is an integer expression indicating the last column of the field.

x — is the name of an integer variable which is used as an error indicator by this routine.

The integer formed from the digits contained in the field specified from column 'm' to column 'n' is returned via %RESULT. This routine assumes that the routine ISO CARD has been used to read a card into array 'a'.

Note that the only legal punchings in the field specified by the input parameters 'm' and 'n' are:

space(blank),0-9,+,-

ERROR CONDITIONS: This routine signals errors by setting a value in the error indicator specified on entry. In the event of error, the function assumes the value of the number of the column (i.e. the subscript of the array element) containing the first illegal symbol.

The values of the error indicator are as follows:

0 the field contains a legal integer constant.

1 the field contains no digits, or spaces occur after the leading digit.

2 the field contains an illegal symbol.

3 the first column specified occurs after the last; the function takes the value zero in this case.

IMOD

DEFINITION: This integer function gives the value of the modulus of the integer quantity specified on entry.

SPEC: Not required

CALL: IMOD(n)

n is an integer expression

The integer value of the modulus of n is returned via %RESULT. See MOD.

ERROR CONDITIONS: Integer overflow occurs if an attempt is made to take the modulus of the largest negative integer (-2147483648).

INT

DEFINITION: This function gives the value of the nearest integer to the quantity specified on entry.

SPEC: Not required

CALL: INT(x)

x is a longreal expression.

The value of the nearest integer to x is returned via %RESULT.

ERROR CONDITIONS: If x is less than -2^{31} or greater than $2^{31}-1$, the job terminates with the following message:

INTPT TOO LARGE

as INT is interpreted as INTPT (exprn>+0.5).
This fault is trappable as FAULT 27.

INTEGER

DEFINITION: This map enables the user to access a particular 4 byte integer location whose address is specified as the parameter.

SPEC: Not required

CALL: INTEGER(n)

n is an integer expression giving the absolute address in core of the required location.

This map may be used to access the contents of a correctly aligned integer location or to store a value at the specified address:

```
%INTEGER I,J
I=1234               ;        !I is an integer with value 1234
J=INTEGER(ADDR(I))   ;        !J is an integer with the value
                               1234(=I).  I is unchanged
INTEGER(ADDR(I))=5678 ;       !I now has the value 5678 J is
                               unchanged
```

ERROR CONDITIONS: If the specified location is not correctly full-word aligned, an address error occurs when the routine attempts to access it.

INT PT

DEFINITION: This integer function gives the value of the integral part of the quantity specified on entry.

SPEC: Not required

CALL: INT PT(x)

x is a real or longreal expression.

The integral part of x is returned via %RESULT.
The integral part is the integer that is less than or equal to the expression. Hence INT PT (-3.6) is -4.

ERROR CONDITIONS: If x is less than -2^{32}or greater than 2^{31}-1, the job terminates with the following message:

INT PT TOO LARGE

This fault is trappable as FAULT 27.

INVERT

DEFINITION: This routine inverts an NxN matrix, making use of the routines DIV MATRIX and UNIT. The method of Gaussian elimination with partial pivoting is used. The given matrix is destroyed.

SPEC:
```
%EXTERNALROUTINESPEC INVERT (%LONGREALARRAYNAME a,b, %INTEGER n,
                             %LONGREALNAME det)
```

CALL: INVERT (a,b,n,det)

a,b are the names of two longreal two-dimensional arrays. Array b contains the matrix to be inverted.

n is the integer expression whose value, N, gives an upper limit for the dimensions of the given arrays, the lower bounds being taken to be 1.

det is the name of a longreal variable which is set to the value of the determinant of the matrix in array b.

The required inverse is placed in a and the input matrix is destroyed.
The value of the determinant of the input matrix is placed in det.
Note that large errors may occur if the matrix is ill-conditioned, and if this is suspected, perturbed matrices should be examined.

ERROR CONDITIONS: If n is less than, or equal to zero, the job terminates with the following message:

```
INVERT DATA FAULT N = <plus'><unsigned integer>
```

ISO CARD

DEFINITION: This routine reads the next card or next card image from the currently selected input stream, and fills the 80-byte array starting at element 1 of the array specified on entry.

SPEC: %EXTERNALROUTINESPEC ISO CARD (%BYTEINTEGERARRAYNAME a)

CALL: ISO CARD (a)

a is the name of a byte integer array.

The array, a, contains the Edinburgh ISO equivalents of the card punching.

Note that valid extended Hollerith characters which have no ISO equivalent will be interpreted as SUB(X'1A'). Punchings which lie outside the 256 character extended Hollerith code will result in card rejection by the hardware.
The effect of using ISO CARD on paper tape input is undefined.

ERROR CONDITIONS: None

LENGTH

DEFINITION: This map accesses the location containing the current length of a string.

SPEC: Not required

CALL: LENGTH(s)

s is the name (of type %STRING) of the location containing the string whose length byte is to be accessed.

This map enables the number of characters in a string to be measured or adjusted without knowing the physical layout of the string.

Note if the layout of the string is increased the characters 'added' to the string will be undefined.

ERROR CONDITIONS: None

LOG

DEFINITION: This longreal function gives the value of the logarithm to the base 'e' of the quantity specified on entry.

SPEC: Not required

CALL: LOG(x)

x is a real or longreal expression.

The real or longreal value of the logarithm to the base 'e' of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If x is negative, or zero, the job terminates with the following message:

LOG NEGATIVE

This fault is trappable as FAULT 6.

LOGGAMMA

DEFINITION: This longreal function gives the value of the log-gamma function of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC LOGGAMMA(%LONGREAL x)

CALL: LOGGAMMA(x)

x is a real or longreal expression.

The real or longreal value of the logarithm to the base 'e' of the gamma function of the quantity specified by x is returned via %RESULT.

$$\log_e \Gamma(x)$$

ERROR CONDITIONS: If x is less than or equal to zero, or if x is greater than or equal to $4.2937*10^{73}$ the job terminates with the following message:

ARGUMENT NOT WITHIN PERMITTED RANGE
0<x<4.2913@73.

LOGTEN

DEFINITION: This real or longreal function gives the logarithm to the base 10 of the quantity specified on entry.

SPEC: %EXTERNALLONGREALFNSPEC LOGTEN (%LONGREAL x)

CALL: LOGTEN(x)

x is a real or longreal expression.

The real or longreal value of the logarithm to the base 10 of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If x is negative or zero, the job terminates with the following message:

LOG NEGATIVE

This fault is trappable as FAULT 6.

LONG REAL

DEFINITION: This map enables the user to access a particular long real location whose address is specified as the parameter.

SPEC: Not required

CALL: LONG REAL (n)

n is an integer expression giving the address of the location to be accessed.

This map may be used to access the contents of a corretly aligned long real location or to store a value at the specified address:

```
%LONGREAL X,Y
X=49.83                   ;      !X contains 49.83
Y=LONGREAL (ADDR(X))      ;      !Y has the value 49.83, X is
                                  unchanged
LONGREAL (ADDR(X))=52.36 ;      !X now contains 52.36, Y is unchanged
```

Note that LONG REAL is always double length and is not affected by %REALSNORMAL.

ERROR CONDITIONS: None

MOD

DEFINITION: This longreal function gives the value of the modulus (i.e. the absolute value) of the quantity specified on entry.

SPEC: Not required

CALL: MOD(x)

x is a real or longreal expression.

The real or longreal value of the modulus of x is returned via %RESULT. Note that MOD has a different effect from using modulus signs in that the latter leaves the type of the argument unchanged.

ERROR CONDITIONS: None

MULT MATRIX

DEFINITION: This routine forms the matrix product A=BC where B is NxP,C is PxM, and A, the product is NxM. The arrays used must be distinct.

```
SPEC:  %EXTERNALROUTINESPEC MULT MATRIX(%LONGREALARRAYNAME a,b,c,
                                          %INTEGER n,p,m)
```

CALL: MULT MATRIX (a,b,c,n,p,m)

a,b,c are the names of three longreal two-dimensional arrays. Arrays b and c contain the rectangular matrices to be multiplied together.

n,p,m are integer expressions whose values N,P and M, give the upper bounds of the dimensions for the given arrays, the lower limits being taken to be 1.

Matrix multiplication is performed on the matrices stored in arrays b and c. The resulting matrix is stored in array a. Arrays b and c are unchanged.

ERROR CONDITIONS: If the upper bounds of any of the arrays are zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

MULT TR MATRIX

DEFINITION: This routine forms the matrix product A=BC where C is the transpose of a given MxP matrix C. If matrix B is NxP, then A is NxM. The arrays used must be distinct.

SPEC: %EXTERNALROUTINESPEC MULT TR MATRIX (%LONGREALARRAYNAME a,b,c,
%INTEGER n,p,m)

CALL: MULT TR MATRIX (a,b,c,n,p,m)

a,b,c are the names of three longreal two-dimensional arrays. The array c contains the matrix to be transposed, and array b contains the matrix by which the transpose is to be multiplied.

n,p,m are the integer expressions whose values, N,P and M, give the upper limits of the dimensions of the given arrays, the lower limits being taken to be 1. Note that C has dimensions M and P in that order.

The transpose of the matrix in array c is multiplied by the matrix in b. The resulting matrix is stored in array a. Arrays b and c are unchanged.

ERROR CONDITIONS: If any of the upper bounds of the matrices is zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

NEWLINE

DEFINITION: This routine produces a 'newline' character on the currently selected output stream.

SPEC: Not required

CALL: NEWLINE

No parameters

A 'newline' character is produced on the output stream.

ERROR CONDITIONS: See PRINT SYMBOL

NEWLINES

DEFINITION: This routine produces the specified number of 'newline' characters on the currently selected output stream.

SPEC: Not required

CALL: NEWLINES(n)

n is an integer expression the last 8 bits of which indicate the number of 'newline' characters to be output.

n 'newline' characters are produced on the output stream. Note that n<=0 has no effect.

ERROR CONDITIONS: See PRINT SYMBOL

NEWPAGE

DEFINITION: This routine causes the 'form-feed' character to be put into the output stream.

SPEC: Not required

CALL: NEWPAGE

No parameters

The 'form-feed' character is put into the output stream. If the currently selected output stream goes to the Line Printer, then the paper is advanced to the head of a new page.

ERROR CONDITIONS: See PRINT SYMBOL

NEXT ITEM

DEFINITION: This function reads the next symbol from the currently selected input stream, as a one-character-string. The symbol may be read again, for example by NEXT ITEM or NEXT SYMBOL.

SPEC: Not required

CALL: NEXT ITEM

No parameters

The ISO-numeric value of the next symbol from the input stream is obtained. This symbol can be obtained again by either a 'READ ITEM' or a 'NEXT ITEM' instruction.

ERROR CONDITIONS: See READ SYMBOL

NEXT SYMBOL

DEFINITION: This function gives the ISO numerical value of the next symbol appearing on the currently selected input stream.

SPEC: Not required

CALL: NEXT SYMBOL

No parameters

The ISO numerical value of the next symbol appearing on the input stream is returned.
A subsequent call of READ SYMBOL or NEXT SYMBOL will obtain the same symbol.

ERROR CONDITIONS: See READ SYMBOL

NL

DEFINITION: This function returns as a result the value of the internal code of the newline character.

SPEC: Not required

CALL: NL

No parameters

The ISO numerical value of the newline character is returned.

ERROR CONDITIONS: None.

NULL

DEFINITION: An NxM null matrix is set up on a specified array.

SPEC: %EXTERNALROUTINESPEC NULL(%LONGREALARRAYNAME a, %INTEGER n,m).

CALL: NULL(a,n,m)

a is the name of a longreal two-dimensional array.
n,m are the integer expressions whose values, N and M, give upper bounds for the dimensions of the given array, the lower bounds being taken to be 1.

All elements of the array are set to zero.

ERROR CONDITIONS: If either of the matrix bounds of the array are found to be zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

OPENDA

DEFINITION: This routine opens the direct access file on logical channel (n) where (n) lies between 1 and 80. The logical channel number links to a physical file (on a disk) via the DD card defining the file.

SPEC: %EXTERNALROUTINESPEC OPENDA (%INTEGER n)

CALL: OPENDA(n)

n is an integer expression which gives the channel number specified on the DD card which defines the file, and whose value must be between 1 and 80 inclusive.

The file is made ready for reading or writing.

ERROR CONDITIONS: If a file is opened twice without being closed, the job terminates with the following message:

```
DA/CHANNEL OPENED TWICE
```

If an attempt is made to open a file which has not been defined, the job terminates with:

```
DA/MISSING DD CARD
```

If an attempt is made to open a file on a channel less than 1 or greater than 80, the job terminates with:

```
DA/ILLEGAL CHANNEL NUMBER
```

Although the logical channel number may lie in the range 1 to 80 it may not conflict with any channel numbers assigned to either stream I/O or to sequential files.

The user must declare his direct access file requirements to the operating system using Job Control Language or Command. Information on the necessary JCL is contained in the appropriate User's Guide.

OPENSQ

DEFINITION: This routine opens a sequential file on the logical channel indicated. The logical channel number links to a physical file via the DD card defining the file.

SPEC: %EXTERNALROUTINESPEC OPENSQ (%INTEGER m)

CALL: OPENSQ(m)

m is an integer expression which specifies the channel number, and whose value must lie between 1 and 80 inclusive.

The file is made ready for reading or writing, but note that both options may not be exercised.

ERROR CONDITIONS: If a file is opened twice without being closed, the job is terminated with the following message:

SQ/FILE OPENED TWICE

If the channel number specified lies outside the given range, the job terminates with:

SQ/ILLEGAL CHANNEL NUMBER

An attempt to open a file which has not been defined will cause the job to be terminated with:

SQ/MISSING DD CARD

Although the logical channel number may lie in the range 1 to 80 it may not conflict with any channel numbers assigned to either stream I/O or to direct access files.

The user must declare his sequential file requirements to the operating system using Job Control Language. Information on the necessary JCL is contained in the appropriate User's Guide

PI

DEFINITION: This function returns the value of π (3.141592653589793).

SPEC: Not required

CALL: PI

PRINT

DEFINITION: This routine prints the value of the specified real number on the currently selected output medium in fixed-point form.

SPEC: Not required

CALL: PRINT(x,m,n)

x is a longreal expression which is to be output.

m is an integer expression indicating how many digits have to be output before the decimal point.

n is an integer expression indicating how many digits have to be output after the decimal point.

x is printed via the current output stream in fixed-point form, with m digits before, and n digits after the decimal point. Insignificant leading zeros are replaced by spaces and the sign is right justified.
Positive sign is represented by a space.
If more than m significant figures occur before the decimal point, the point will be displaced to the right and extra digits inserted.

ERROR CONDITIONS: See PRINT SYMBOL

PRINT CH

DEFINITION: This routine places the seven least significant bits of the parameter as the next character in the output stream.

SPEC: Not required

CALL: PRINT CH (n)

n is an integer expression giving the ISO code representation of the character to be printed.

The character, represented in ISO code by the last 7 bits of n, is printed out on the currently selected output stream.

ERROR CONDITIONS: See PRINT SYMBOL

PRINT FL

DEFINITION: This routine prints the longreal value specified by the first parameter on the output medium in floating-point form。

SPEC: Not required

CALL: PRINT FL (x,n)

x is a real expression which is to be printed out.
n is an integer expression indicating the number of digits to be printed after the decimal point.

A floating-point real number is printed out using n+7 printing positions, with n digits after the decimal point. The number is standardised in the range 1<=x<10.

ERROR CONDITIONS: See PRINT SYMBOL

PRINT STRING

DEFINITION: This routine outputs a string.

SPEC: NOT REQUIRED

CALL: PRINT STRING(s)

s is the string expression to be output.

The string s is written to the currently selected output stream.

ERROR CONDITIONS: If the available amount of output, as specified by the job control statements (or by default), has been used, the job terminates with the following message:

OUTPUT EXCEEDED

This fault is not trappable.

PRINT SYMBOL

DEFINITION: This routine prints (on the currently selected output stream) the symbol whose location is specified on entry.

SPEC: Not required

CALL: PRINT SYMBOL(n)

n is an integer expression giving the ISO code representation of the symbol to be printed.

The symbol, represented in ISO code by the last 7 bits of n, is printed out on the currently selected output stream. If the symbol is not in the IMP extended character set, SUB is placed in the output stream.

ERROR CONDITIONS: If an attempt is made to output more data than has been specified for the current output stream, the job terminates with the following message:

OUTPUT EXCEEDED

This fault is not trappable.

RADIUS

DEFINITION: This longreal function gives the value of the radius of a circle whose equation is of the form $X^2+Y^2=R^2$(where R is the radius).

SPEC: Not required

CALL: RADIUS(x,y)

x,y are real or longreal expressions.

The real or longreal value of the radius (i.e. the value of the positive square root of (x^2+y^2)) is returned via %RESULT.

ERROR CONDITIONS: If x or $x^2+y^2$is greater than the largest permissible real number an overflow condition will occur and the job will terminate with the following message:

LIBRARY FN FAULT 10

This fault is trappable as Fault 25.

RANDOM

DEFINITION: This function produces a set of random numbers either in a rectangular or Gaussian distribution.

SPEC: %EXTERNALREALFNSPEC RANDOM(%INTEGERNAME i, %INTEGER n)

CALL: RANDOM(i,n)

i is the name of an integer variable which must be initalised to any odd integer. Note that a value of i<50001 may produce a sequence which is similar in the first few terms.

n is an integer expression whose value controls the type of distribution: n=1 gives a rectangular distribution, n>1 gives a Gaussian distribution with a mean value of n/2 and a standard deviation of $\sqrt{n/12}$. For a Gaussian distribution a value of n greater than 10 is recommended.

A real number in the range 0-n is returned. Note that i contains an integer random sequence in the range 0-(2 -1).

Users may wish to scale the distribution. To produce Z with Gaussian distribution, and mean X standard deviation S, call RANDOM with N=12 and scale thus:-

```
Z=(RANDOM(I,12)-6)*S+X
```

ERROR CONDITIONS: If N is negative the job terminates with the following message:

NEGATIVE ARGUMENT IN RANDOM.

READ

DEFINITION: This routine reads numerical data from the currently selected input stream and uses 'read symbol' (q.v.). The number read may be in fixed or floating point form.

SPEC: Not required

CALL: READ (x)

x is the name of an integer or real variable.

The next number is taken from the input stream and stored in x. Numbers whose modulus is less than $2.4\text{x}10^{-78}$ are treated as zero. Numbers whose modulus is greater than $7.2\text{x}10^{75}$ will overflow.

ERROR CONDITIONS: If the number assigned to an integer variable is not integral or is outside the range $-2^{31}+1$ to $2^{31}-1$, or an exponent is not an integer, the job terminates with the following message:

REAL INSTEAD OF INTEGER IN DATA

This fault is trappable as FAULT 16. If any characters other than digits +,-,@,., occur, the job terminates with the following message:

SYMBOL IN DATA n

This fault is trappable as FAULT 14. In the current implementation, if this fault is trapped the offending symbol may be obtained using READ SYMBOL.

If an attempt is made to read more data than has been provided on the input stream, the job terminates with the following message:

INPUT FILE ENDED

This fault is trappable as FAULT 9.

READ CH

DEFINITION: This routine reads a character from the input stream in internally coded form before any line reconstruction is done.

SPEC: Not required

CALL: READ CH (n)

n is the name of an integer variable.

The next character on the input stream, belonging to the full Edinburgh Standard Character Set, is read in and stored in the location specified by n. The character is stored (in internal code) in the 7 least significant bits of n.

ERROR CONDITIONS: If an attempt is made to read more data than has been provided on the input stream, the job terminates with the following message:

```
INPUT FILE ENDED
```

This fault is trappable as FAULT 9.

READDA

DEFINITION: This routine reads data from the channel specified into the specified area, starting from the specified block on the file.

SPEC:
```
%EXTERNALROUTINESPEC READDA (%INTEGER n, %INTEGERNAME sect,
                                      %NAME begin, end)
```

CALL: READDA (n,sect,begin,end)

n   is an integer expression which specifies the channel number and whose value must lie between 1 and 80.

sect   is an integername parameter whose value on entry specifies the block at which reading is to start and, on exit, contains the number of the last block read from.

begin }
end   } specify the area into which data is to be read.

Information is read from file n starting at section sect and is written into the area starting at begin and finishing at end. This area is normally an array.

ERROR CONDITIONS: If ADDR (end) < ADDR (begin), the job terminates with the following message:

DA/CORE ADDRESSES INSIDE-OUT.

If an attempt is made to access a block on the file which does not exist, the job terminates with the message:

DA/READ/WRITE OFF LIMITS.

If the specified channel is not open, the job terminates with:

DA/CHANNEL NOT OPEN

The recordsize for direct access files on both the 4/75 and the 370/158 is fixed at 1024 bytes and, on creation, the complete file is filled with the unassigned pattern.

READ ITEM

DEFINITION: This routine reads the next symbol from the currently selected input stream, and sets this up as a one-character string.

SPEC: Not required

CALL: READ ITEM (s)

s is the name (of type %STRING) of the location in which the symbol is stored.

The ISO-numeric value of the next symbol from the input stream is stored in s.

ERROR CONDITIONS: See READ SYMBOL

READSQ

DEFINITION: This routine reads information from the channel specified on entry into the array specified on entry.

SPEC: %EXTERNALROUTINESPEC READSQ (%INTEGER m, %NAME begin,end)

CALL: READSQ(m,begin,end)

m is an integer expression which specifies the channel number, and whose value must lie between 1 and 80 inclusive.

begin
end } specify the area into which the information is to be read.

Information is read from the channel specified on entry into the area specified on entry, starting at location begin, and finishing at location end. The area specified is normally an array.

ERROR CONDITIONS: If the specified channel is not open the job will terminate with the following message:

SQ/CHANNEL NOT OPEN

If ADDR(end) <ADDR(begin), the job will terminate with the message:

SQ/CORE ADDRESSES INSIDE-OUT

If an attempt is made to read beyond the end of a file, the job terminates with the message:

INPUT FILE ENDED

This fault is trappable as FAULT 9.

READ STRING

DEFINITION: This routine reads the next string from the currently selected input stream.

SPEC: Not required

CALL: READ STRING (s)

s is the name (of type %STRING) of the location in which the string of symbols is stored.

The string of symbols (which must be enclosed within quotes and must be <=255 symbols) is read into s.

ERROR CONDITIONS: If the next significant symbol (ignoring spaces and newlines) is not a quote, the job terminates with the following message:

SYMBOL IN DATA n

This fault is trappable as FAULT 14.
If the declared maximum length of the stringname parameter is insufficient to hold the string, the job terminates with the message:

CAPACITY EXCEEDED

See READ SYMBOL for other errors.

READ SYMBOL

DEFINITION: This routine reads the next symbol which appears on the currently selected input stream.

SPEC: NOT REQUIRED

CALL: READ SYMBOL(n)

n is the name of an integer, shortinteger or byteinteger variable.

The next symbol on the input stream is read in and stored in the location specified by n.
The symbol is stored (in internal code) in the 7 least significant bits of n.
All non-printing characters, (including CR), are ignored.

ERROR CONDITIONS: If an attempt is made to read more data than has been provided on the input stream, the job terminates with the following message:

INPUT FILE ENDED

This fault is trappable as FAULT 9.
Invalid characters in the input stream are replaced by SUB, and if SUB is detected by this routine, the job terminates with the following message:

SUBSTITUTE CHARACTER IN DATA

This fault is trappable as FAULT 18.

REAL

DEFINITION: This map enables the user to access a particular real location whose address is specified as the parameter.

SPEC: Not required

CALL: REAL(n)

n is an integer expression giving the address of the location to be accessed.

This map may be used to access the contents of a correctly aligned real location or to store a value at the specified address:

```
%REAL X,Y
X=234.62            ;          !X contains 234.62
Y=REAL(ADDR(X))     ;          !Y contains 234.62, X is unchanged
REAL(ADDR(X))=-52.1 ;          !X contains -52.1, Y is unchanged
```

Note that REAL is always single length and is not affected by %REALSLONG.

ERROR CONDITIONS: None.

RECORD

DEFINITION: This mapping function converts the absolute address specified on entry to a form such that it may be assigned to a %RECORDNAME variable using the '==' assignment operator.

SPEC: Not required

CALL: RECORD(addr)

addr is an integer expression giving the absolute address to be converted.

The address, in a form such that it can be assigned to a %RECORDNAME variable using the '==' assignment operator, is returned via %RESULT.

The following example illustrates the use of a record map:

```
%INTEGERARRAY NAME(1:10)
%RECORDFORMAT R1(%INTEGER I,J, %BYTEINTEGER K, %STRING (15) S,T)
%RECORDNAME R(R1)
R==RECORD (ADDR(NAME(1)))
X_I is a reference to NAME(1)
X_K is a reference to the top byte of NAME(3)
```

For more information on records see the Imp Language Manual.

ERROR CONDITIONS: If the address specified is not double word aligned, an address error is liable to occur.

RFD BINARY

DEFINITION: This function interprets a card which has been stored as column binary in the array specified by the first input parameter to this routine. It evaluates the fixed-point real constant formed from the digits and decimal point occurring in the field specified by the second and third input parameters.

SPEC:
```
%EXTERNALLONGREALFNSPEC RFD BINARY (%SHORTINTEGERARRAYNAME a,
                                    %INTEGER m,n, %INTEGERNAME x)
```

CALL: RFD BINARY (a,m,n,x)

a	is the name of an 80 element short integer array in which the information from the card has been stored.
m	is an integer expression indicating the first column of the field.
n	is an integer expression indicating the last column of the field.
x	is the name of an integer variable which is used as an error indicator by this routine.

The fixed-point real number formed from the digits occurring in the field specified from column 'm' to column 'n' is returned via %RESULT.

Note that the only valid punchings in the field specified by 'm' and 'n' are:

space(blank),0-9,.,+,-

ERROR CONDITIONS: This routine signals errors by setting a value in the error indicator specified on entry. In the event of error, the function assumes the value of the number of the column (i.e. the subscript of the array element) containing the first illegal symbol.

The values of the error indicator are as follows:

0	the field contains a legal real constant.
1	the field contains no digits, or spaces occur after the leading digit.
2	the field contains an illegal symbol.
3	the first column specified occurs after the last; in this case, the function takes the value zero.

RFD ISO

DEFINITION: This function interprets a card which has been stored in the array specified by the first input parameter to this routine. It evaluates the fixed-point real constant formed from the digits and decimal point occurring in the field specified by the second and third input parameters.

SPEC:
```
%EXTERNALLONGREALFNSPEC RFDISO(%BYTEINTEGERARRAYNAME a,
                               %INTEGER m,n,%INTEGERNAME x)
```

CALL: RFD ISO(a,m,n,x)

a	is the name of an 80 byte integer array in which the information from the card has been stored.
m	is an integer expression indicating the first column of the field.
n	is an integer expression indicating the last column of the field.
x	is the name of an integer variable which is used as an error indicator by this routine.

The fixed-point real number formed from the digits contained in the field specified from column 'm' to column 'n' is returned via %RESULT. This routine assumes that the routine ISO CARD has been used to read a card into the array.

Note that the only legal punchings in the field specified by the input parameters 'm' and 'n' are:

space(blank),0-9,.,+,-

ERROR CONDITIONS: This routine signals errors by setting a value in the error indicator specified on entry. In the event of error, the function assumes the value of the number of the column (i.e. the subscript of the array element) containing the first illegal symbol.

The values of the error indicator are as follows:

0	the field contains a legal real constant.
1	the field contains no digits, or spaces occur after the leading digit.
2	the field contains an illegal symbol.
3	the first column specified occurs after the last; the function takes the value zero in this case.

SELECT INPUT

DEFINITION: This routine connects the current input to the stream specified on entry.

SPEC: Not required

CALL: SELECT INPUT(n)

n is an integer expression giving the number of the input stream selected.

The current input is connected to the stream specified by n. The first call of an input routine subsequent to the use of this routine gives a 'newline' character, and all further input is obtained via the new stream.
Any unused information from the current record is lost during execution of this routine.

ERROR CONDITIONS: If no stream is defined on entry to this routine, or if the stream defined has previously been defined for output, the job terminates with the following message:

```
ILLEGAL STREAM
```

SELECT OUTPUT

DEFINITION: This routine connects current output data to the stream specified on entry.

SPEC: Not required

CALL: SELECT OUTPUT(n)

n is an integer expression giving the number of the output stream selected.

The current record is output via the current stream unless the last symbol output before the call of this routine was a 'newline' symbol. In the latter case, nothing is output. All subsequent records are sent to the new stream.

ERROR CONDITIONS: If no stream is defined on entry to this routine, or if the stream defined has been previously defined for input, the job terminates with the following message:

ILLEGAL STREAM

See PRINT SYMBOL

SET MARGINS

DEFINITION: This routine allows the margins of either the current input or the current output user-defined stream (i.e. any stream lying between 1 and 80 inclusive) to be changed.

SPEC: Not required

CALL: SET MARGINS (m,left,right)

m is an integer expression which gives the number of the stream whose margins are to be changed.

left is an integer expression specifying the left hand margin and can take values 1<=left<=right.

right is an integer expression specifying the right hand margin and can take values left<=right<=160 for input streams or left<=right<=132 for output streams.

If SET MARGINS operates on the current input stream, the default values for which are left=1, right=80, (with exception for Stream 99: right=132), left and right may take values which satisfy 1<=left<=right<=160. The effect of changing input margins is truncation of the input record at the beginning or truncation of the input record at the end, if right is reduced. If SET MARGINS operates on the current output stream, the default values for which are left=1, right=120, left and right may take values which satisfy 1<=left<=right<=132. The effect of altering the left hand margin is to inset the output record, and the effect of reducing the right hand margin is to output any overflow on the next line. The margins, once set on a given stream, remain as characteristics of that stream until the next call of SET MARGINS for the stream.

ERROR CONDITIONS: If m is not the current input or output stream; if m<1 or m>80; if left>right or if either lie outside their defined limits, the job terminates with an appropriate message.

SHIFTC

DEFINITION: This function cyclically shifts a given bit pattern by a specified number of bits.

SPEC: %EXTERNALINTEGERFNSPEC SHIFTC (%INTEGER n,m)

CALL: SHIFTC (n,m)

n is an integer expression whose binary pattern is to be shifted.

m is an integer expression indicating the number of places n is to be shifted. Note that m may be positive, indicating a shift to the left, or negative, indicating a shift to the right.

The bit pattern of the integer in location n is shifted m places, left or right according to the sign (+ or - respectively) of m.

Bits lost off one end of the word reappear in the same order the other end of the word.

ERROR CONDITIONS: If m is greater than 32, the job terminates with the following message:

ILLEGAL SHIFT

SHORT INTEGER

DEFINITION: This map enables the user to access a particular short integer location whose address is specified as the parameter.

SPEC: Not required

CALL: SHORT INTEGER(n)

n is an integer expression giving the absolute address in core of the required location.

This map may be used to access the contents of a 2 byte integer location or to store a value at the specified address.

```
%SHORT INTEGER I,J
I=1234                     ;   !integer containing 1234
J=SHORT INTEGER(ADDR(I))   ;   !J has value 1234, I is unchanged
SHORT INTEGER(ADDR(I))=3456 ;  !I has value 3456, J is unchanged
```

ERROR CONDITIONS: If the specified location is not correctly half-word aligned, an address error occurs.

SIN

DEFINITION: This longreal function gives the value of the sine of the quantity specified on entry. This quantity must be in radians.

SPEC: Not required

CALL: SIN(x)

x is a real or longreal expression.

The value of the sine of x is returned via %RESULT.

ERROR CONDITIONS: If the modulus of the argument passed to this function is greater than or equal to 10^{15} , the result would be wholly inaccurate, and the job terminates with the following message:

TRIG FN INACCURATE

This fault is trappable as FAULT 22.
It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases, until, at $|x|=10^{15}$, the result would be quite meaningless.

SKIP SYMBOL

DEFINITION: This routine passes over the next symbol which appears on the currently selected input stream, without reading it.

SPEC: Not required

CALL: SKIP SYMBOL

No parameters

The next symbol appearing on the input stream is passed over without reading.

ERROR CONDITIONS: See READ SYMBOL

SOLVE LN EQ

DEFINITION: This routine solves a linear system of equations Ax=b for vector x. A, an NxN matrix, and b, a vector, are destroyed. The method of Gaussian elimination with partial pivoting is used.

SPEC: %EXTERNALROUTINESPEC SOLVE LN EQ (%LONGREALARRAYNAME a,b,
%INTEGER n, %LONGREALNAME det)

CALL: SOLVE LN EQ (a,b,n,det)

a is the name of a longreal two-dimensional array which contains square matrix A on input.

b is the name of a longreal one-dimensional array which contains vector b on input.

n is the integer expression whose value, N gives an upper limit for the dimensions of the a and b, the lower limits being taken to be 1.

det is the name of a longreal variable which is set by the routine to the value of the determinant of the matrix in array a.

The solution of the equations, x, is placed in b, and matrix A is destroyed.
The value of the determinant of A is placed in det.
Note that large errors may occur if the matrix is ill- conditioned and, if this is suspected, perturbed matrices should be examined.

ERROR CONDITIONS: If n is less than, or equal to zero, the job terminates with the following message:

SOLVE LN EQ DATA FAULT N = X

If the matrix is not invertible, the value of det will be zero on exit.

SPACE

DEFINITION: This routine produces one 'space' character on the currently selected output stream.

SPEC: Not required

CALL: SPACE

No parameters

One 'space' character is produced on the output medmum.

ERROR CONDITIONS: See PRINT SYMBOL

SPACES

DEFINITION: This routine produces the specified number of 'space' characters on the currently selected output stream.

SPEC: Not required

CALL: SPACES(n)

n is an integer expression the last 8 bits of which indicate the number of 'space' characters to be produced.

n 'space' characters are produced on the output stream.

ERROR CONDITIONS: See PRINT SYMBOL

SQRT

DEFINITION: This longreal function gives the value of the positive square root of the quantity specified on entry.

SPEC: Not required

CALL: SQRT(x)

x is a real or longreal expression.

The real or longreal value of the positive square root of the quantity specified by x is returned via %RESULT.

ERROR CONDITIONS: If a negative argument is passed to this routine, the job terminates with the following message:

SQRT NEGATIVE

This fault is trappable as FAULT 5.

STRING

DEFINITION: This map enables the user to access a particular string location whose address is specified as the parameter.

SPEC: Not required

CALL: STRING(n)

n is an integer expression giving the absolute address in core of the required location.

This map may be used to access the contents of a string location or to store a string at the specified address:

```
%STRING(8) S,T
S='ABCDEFGH'                 ;   !the string 'ABCDEFGH' is assigned to
                                  S
T=STRING (ADDR(S))           ;   !the string 'ABCDEFGH' is assigned to
                                  T, S is unchanged
STRING (ADDR(S))='12345678'  ;   !the string '12345678' is assigned to
                                  S, T is unchanged
```

Note that the first byte of a string is taken as the length of the string.

ERROR CONDITIONS: None

SUB MATRIX

DEFINITION: This routine subtracts one NxM matrix from another.

SPEC: %EXTERNALROUTINESPEC SUB MATRIX (%LONGREALARRAYNAME a,b,c, %INTEGER n,m)

CALL: SUB MATRIX(a,b,c,n,m)

a,b,c are the names of three longreal two-dimensional arrays. The matrix in array c is subtracted from the matrix in b.

n,m are the integer expressions whose values, N and M, give the upper limits for the dimensions of a,b and c, the lower limits being taken to be 1.

Array a contains the matrix difference of matrices in arrays b and c. Arrays b and c are unchanged.

ERROR CONDITIONS: If the array bounds of any of the matrices are zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

TAN

DEFINITION: This longreal function gives the value of the tangent of the quantity specified on entry. This quantity must be in radians.

SPEC: Not required

CALL: TAN(x)

x is a real or longreal expression.

The real or longreal value of the tangent of x is returned via %RESULT.

ERROR CONDITIONS: If the modulus of the argument of this function is greater than or equal to 10^{15} the result would be wholly inaccurate, and the job terminates with the following message:

TRIG FN INACCURATE

This fault is trappable as FAULT 22.
It should be noted that the accuracy of the result falls off slowly as the modulus of the argument increases, until at $|x|=10^{15}$, the result would be quite meaningless. If the value of COS(x) is less than or equal to 10^{-14} then the job terminates with the message:

TAN TOO LARGE

This fault is trappable as FAULT 23.

TIME

DEFINITION: This function yields the present time of day.

SPEC: %EXTERNALSTRINGFNSPEC TIME

CALL: TIME

No parameters

The time is returned in the form hh.mm.ss via %RESULT, and may be printed out using PRINTSTRING.

ERROR CONDITIONS: None.

TO STRING

DEFINITION: This function creates a one-character string.

SPEC: Not required

CALL: TO STRING(n)

n the least significant 7 bits of the integer expression n represent the ISO code value of the character to be placed in the string.

A one-character string, containing the character specified by n, is created.

ERROR CONDITIONS: None.

TRANS MATRIX

DEFINITION: This routine sets up, on a specified two-dimensional array, the NxM transpose of a given MxN matrix. The specified arrays must be distinct.

SPEC: %EXTERNALROUTINESPEC TRANS MATRIX (%LONGREALARRAYNAME a,b, %INTEGER n,m)

CALL: TRANS MATRIX (a,b,n,m)

a,b are the names of two longreal two-dimensional arrays. Array b contains the rectangular matrix to be transposed.

n,m are the integer expressions whose values, N and M, give upper bounds for the dimensions of the specified arrays, the lower bounds being taken to be 1.

The transpose of the matrix, stored in array b, is set up on the array a. Array b is unchanged.

ERROR CONDITIONS: If the upper bounds of either matrix are zero or negative, the job terminates with the following message:

```
MATRIX BOUND ZERO OR NEGATIVE
```

UNIT

DEFINITION: An NxN unit matrix is set up on a specified array.

SPEC: %EXTERNALROUTINESPEC UNIT (%LONGREALARRAYNAME a, %INTEGER n).

CALL: UNIT (a,n)

a is the name of a two-dimensional longreal array.

n is the integer expression whose value, N, gives an upper bound for the dimensions of a, the lower bounds being taken to be one.

The diagonal elements of a are set to 1 and all other elements to zero.

ERROR CONDITIONS: If N is zero or negative, the job terminates with the following message:

MATRIX BOUND ZERO OR NEGATIVE

WRITE

DEFINITION: This routine prints the value specified by the first parameter on the output medium using one character position for the sign and the next n positions for the digits (where n represents a number specified on entry to the routine).

SPEC: Not required

CALL: WRITE(x,n)

x is an integer expression whose value is to be printed out.

n is an integer expression specifying the number of digits to be printed.

An integer is printed out with n significant figures preceded by a sign.
Non-significant leading zeros are suppressed, being replaced by spaces, and a positive sign is indicated by a space.
If x has more than n significant figures, or if n is zero or negative, all the figures are printed out, but the righthand end of the number is then out of alignment. The sign always immediately precedes the most significant digit.
x is accurate to 11 significant figures.

ERROR CONDITIONS: See PRINT SYMBOL

WRITEDA

DEFINITION: This routine writes the information held in the specified area on to the specified file, starting at the specified block.

SPEC:
```
%EXTERNALROUTINESPEC WRITEDA (%INTEGER n, %INTEGERNAME sect,
                              %NAME begin, end)
```

CALL: WRITEDA(n,sect,begin,end)

n   is an integer expression which specifies the channel number and whose value must lie between 1 and 80 inclusive.

sect   is an integername parameter whose value on entry specifies the block at which writing is to start and, on exit, it contains the number of the last block written.

begin }
end }   specify the area to which data is to be written

The information contained in the area begin to end is written to the direct access file on channel nn starting at section sect. Writing always commences at the start of a block and if the information written does not fill a complete block or blocks then the remainder of the last block written is filled with rubbish.

ERROR CONDITIONS: If ADDR (end) < ADDR (begin), the job terminates with the following message:

DA/CORE ADDRESSES INSIDE-OUT

If an attempt is made to access a non-existent block on the specified file, the job terminates with the message:

DA/READ/WRITE OFF LIMITS

If the specified channel is not open, the job terminates with:

DA/CHANNEL NOT OPEN

WRITESQ

DEFINITION: This routine writes the information held in the area specified on entry on to the next logical record of the file on the channel specified on entry.

SPEC: %EXTERNALROUTINESPEC WRITESQ (%INTEGER m, %NAME begin,end)

CALL: WRITESQ(m,begin,end)

m is an integer expression which specifies the channel number whose value must lie between 1 and 80.

begin, end } specify the core area from which the information is to be taken.

The information contained in core, between begin and end inclusive, is written in a new logical record on to the file specified by channel m. The area specified is normally an array.

ERROR CONDITIONS: If the specified channel is not open the job terminates with the following message:

```
SQ/CHANNEL NOT OPEN
```

If ADDR(end)<ADDR(begin), the job terminates with the message:

```
SQ/CORE ADDRESSES INSIDE-OUT
```

3

Cross Calling Between FORTRAN and IMP

3. Calling IMP routines from FORTRAN programs and FORTRAN subprograms from IMP

One of the design aims of the Edinburgh IMP and Edinburgh FORTRAN compiler implementations has been to allow programs to include routines from both languages. To this end routines in both languages obey the same linkage conventions and, where possible, the same conventions for passing parameters. This section defines the means of communicating between routines written in different languages.

Routine and function calls

IMP routines which are to be called explicitly from a FORTRAN routine must be defined as %EXTERNAL, and similarly FORTRAN routines must be specified as %EXTERNAL within IMP routines from which they are called.

e.g. an IMP routine

```
%EXTERNALROUTINE SUB(...)
.
.
.
END
```

could be called from a FORTRAN routine by

```
CALL SUB(...)
```

Functions are treated similarly. The permitted IMP function types are %INTEGERFN, %REALFN and %LONGREALFN corresponding to FORTRAN INTEGER*4 FUNCTION, REAL*4 FUNCTION and REAL*8 FUNCTION respectively.

Scalar parameters

FORTRAN passes all its scalar parameters by reference. This restricts the permitted IMP parameter types to %SHORTINEGERNAME, %INTEGERNAME, %REALNAME and %LONGREALNAME, corresponding to FORTRAN parameters INTEGER*2, INTEGER*4, REAL*4, and REAL*8.

e.g. the FORTRAN routine defined by

```
SUBROUTINE SUB1 (X,Y,I)
REAL*8 X,Y
.
.
.
END
```

could be accessed from an IMP routine containing the specification

```
%EXTERNALROUTINESPEC SUB1 (%LONGREALNAME X,Y,%INTEGERNAME I)
```

Array parameters

FORTRAN and IMP conventionally set up array parameters in very different ways. FORTRAN passes the address of the first element of an array and defines the structure within the called routine while IMP passes a descripter defining the location and structure of the array.

The following techniques are recommended for handling array parameters.

(a) IMP calling FORTRAN

In the routine specification instead of specifying an %ARRAYNAME parameter define a %NAME parameter of the appropriate type and set as actual parameter the first element of the array.

e.g. Where the called routine is

```
SUBROUTINE R(A,...)
DIMENSION A(50)
.
.
.
END
```

the calling routine should include statements of the form

```
%EXTERNALROUTINESPEC R(%REALNAME X...)
%REALARRAY B(1:50)
.
.
.
R (B(1),...)
```

References to elements of the array A in FORTRAN routine R will access the array B in the IMP routine.

(b) FORTRAN calling IMP

Passing array parameters FORTRAN passes only the address of the first element hence a called IMP routine has to construct its own description. This may be done as follows:
FORTRAN calling routine

```
REAL Y(50,10)
.
.
.
CALL SUB 2(Y,...)
```

IMP called routine

```
%EXTERNALROUTINE SUB 2(%INTEGER AX,...)
%REALARRAYNAME X
%REALARRAYFORMAT P (1:50, 1:10)
X == ARRAY (AX, P)
```

References to elements of array X within the IMP routine will access the array Y defined in the FORTRAN routine. The FORTRAN call sets the address of Y(1,1) i.e. the first element, as the actual parameter.

By defining the corresponding IMP parameter as %INTEGER, AX is a local variable into which this address is stored and which, with the array format specification P, is used by the special mapping function ARRAY to construct an array descriptor for X.

IMP follows the FORTRAN convention for the storage of multi-dimensional arrays, i.e. such that the first subscript varies the fastest.

Routine and function parameters

These may be freely passed between IMP and FORTRAN subject to the type restriction defined above. In the case of IMP a routine or function passed as a parameter may be either external or internal.