



Reference

BASIC-K Summary



ICL endeavours to ensure that the information in this document is correct and fairly stated, but does not accept liability for any error or omission.

The development of ICL products and services is continuous and published information may not be up-to-date. Any particular issue of a product may contain part only of the facilities described in this document or may contain facilities not described here. It is important to check the current position with ICL.

Specifications and statements as to performance in this document are ICL estimates intended for general guidance. They may require adjustment in particular circumstances and are therefore not formal offers or undertakings.

Statements in this document are not part of a contract or program product licence save insofar as they are incorporated into a contract or licence by express reference. Issue of this document does not entitle the recipient to access to or use of the products described, and such access or use may be subject to separate contracts or licences.

Technical Publication 5640

© International Computers Limited 1979

First Edition June 1978

Second Edition January 1979
(Issued as an amended reprint of TP5640)

The BASIC-K software described in this publication was designed and implemented at the University of Kent and is

© University of Kent at Canterbury, 1977

ICL will be pleased to receive readers' views on the contents and organisation, etc. of this publication. Please write to

The Registry (Readership Survey)
UK Software and Literature Distribution Centre
International Computers Limited
60 Portman Road
Reading
Berks RG3 1NR

Distributed by
UK Software and Literature Distribution Centre
International Computers Limited
Registered Office: ICL House, Putney, London SW15 1SW
Printed by ICL Printing Services
Works Road, Letchworth, Herts SG6 1JY

Entry and exit

LOGIN(JNAME=jobname,
UNAME=username,
PROFILE=profilename)

Login to MAC

.

BASIC()

Start of BASIC session

.

BYE

End of BASIC session

.

LOGOUT

Logout, end of MAC session

Fundamentals

Line numbers

1 to 31999

Variable names

A, A0, Z9, A\$, A0\$, Z9\$

Constants

1, 1.3, 1.3E10, 1E-3, "STRING"

String sizes

0 to 160 characters

Array elements

A(3), B(1,J), A\$(3), *not* B\$(1,J)

Array size

Total arrays + strings < 256KB

Default array bounds

(10) or (10,10)

Numeric operators

+, -, *, /, † (or **)

String operator

& (concatenate)

Expressions

(-A†B*6)/D-E†+SQR(3)

Numerical range

Roughly 10⁻⁷⁸ to 10⁷⁵

Numerical accuracy

About 16 decimal digits

Character set

EBCDIC

Immediate mode

Statements without line numbers

File names

Letter plus 0-15 letters or digits,
optionally followed by user name
(X1,US=CUR099)

Initial values

All variables unassigned

Re-initializing

Values retained from one run to the next

USING clause

Left-justified string (<**), right-justified
string (>**), integer (+**), decimal
(-**,*), exponent (+**, *†††††),
currency symbol (£-**,**), and literal
characters anywhere except within † field
of exponent (ANSWER = +**, ***)

Statements

CHAIN "XYZ"

Delete current program, then compile and
run another

DATA 3.4E6, UNQUOTED.
DATA "QUOTED"

Data for a READ

DEF FNA(X)= X*X
DEF FNZ= 3.141592653590

User-defined function

DIM X(3,4), Y\$(10),C(3)

Declare size of arrays

DUMP

Print all current variables

END

Stop running

FILE *30: "PIG"

FILE *N+1: A\$&"1"

FILE*2: "PIG,U\$=CUR099"

Assign file name to channel

FOR K=1 TO 10

FOR P9=C+3 TO -Z STEP-P/Q

Start a loop

GOATEND *1: 200

Go if end of file is ever subsequently read

GOSUB 1050

Call a subroutine

GOTO 50

Go to a line

IF A>=B+1 THEN 100

IF A\$ <> B\$ THEN 20

Conditional GOTO

INPUT P,Q,C1\$

INPUT *1: P\$,Q1.R

Request input values

LET A=A+1

A=A+1

Assign value to variable

LET A\$=B\$(3)="PIG"

Set several variables to same value

LINPUT P\$,Q1\$

LINPUT *3:P\$

Input a whole line

MARGIN 40
MARGIN *3:P+1

Set line length for output

(Note that MAT statements apply to numeric arrays only. They ignore row (and column) 0. Arrays should be declared with DIM.)

MAT A=B	Matrix assignment
MAT A= B+C	Matrix add
MAT A=B-C	Matrix subtract
MAT A=B*C	Matrix multiply
MAT A=(6)*C	Matrix scalar multiply
MAT A=INV(B)	Matrix invert
MAT A=TRN(B)	Matrix transpose
MAT A=CON	Set matrix all ones
MAT A=IDN	Set matrix to identity (square)
MAT A=ZER	Set matrix all zeros
MAT INPUT A,B	Input matrices
MAT PRINT *3: USING N\$:A	Print matrices
MAT PRINT A,B;D;	
MAT READ C	Read matrices from DATA
NEXT P9	End a FOR loop
ON A GOTO 150,20,150	Multi-way branch
OPTIONBASE I	Set lowest array subscript
PRINT	Print values
PRINT A, B;	
PRINT 1+X,TAB(V),Y,"XY"&P\$	
PRINT *0: X	
PRINT USING N\$: X,X\$	
PRINT *P+Q: USING N\$: X,X\$	
RANDOMIZE	Make RND truly random
READ P,P\$	Read from DATA
REM A COMMENT	Comments
RESET *6	Reset channel to start
RESTORE	Restore DATA to start
RETURN	Return from subroutine
STOP	Synonym of END

Commands

Commands do not have line numbers.

ADD TEACHSUBS	Add a program file to current program
BYE	End a BASIC session
DELETE 20,2000	Delete part of current program
LIST	List current program
LIST 5,50	List part of current program
NEW	Create new current program (default name BASICPROGRAM)
NEW PIG	
OLD	Retrieve saved program (default name BASICPROGRAM)
OLD PIG	
OLD PIG, US=CUR099	
RSAVE	As SAVE, but re-sequence as well (default start 10 and step 10). Only the saved version is re-sequenced, not the current program.
RSAVE 100,50, MYPROG	
RSAVE 100,50,MYPROG,10,100	
RUN	Run the current program
SAVE	As LIST but put program on file (default name = current program name)
SAVE MYPROG	
SAVE MYPROG,10,100	
SCR	Synonym of NEW
SETOPTION I	Set compiler options (to E or I or L or R or T)
UNSAVE MYPROG	Delete a file

Use of files

File names can be local files or permanent files; local names take precedence. On output previous version of file is overwritten if it exists. Default I/O files are INPUT and OUTPUT, assigned to SOURCE under MAC, and corresponding to channel 0. In batch source is assigned by default to the journal; the line limit on the journal is 1000 lines.

Character codes in decimal (95 character set)

Character	Code	Character	Code	Character	Code
Space	64	d	132	H	200
[74	e	133	I	201
.	75	f	134	J	208
<	76	g	135	K	209
(77	h	136	L	210
+	78	i	137	M	211
!	79	j	145	N	212
&	80	k	146	O	213
	90	l	147	P	214
\$	91	m	148	Q	215
°	92	n	149	R	216
)	93	o	150	S	217
;	94	p	151	\	224
↑ (uparrow)	95	q	152	T	226
- (minus)	96	r	153	U	227
/	97	(overline)	161	V	228
(bar)	106	s	162	W	229
, (comma)	107	t	163	X	230
%	108	u	164	Y	231
_ (underline)	109	v	165	Z	232
>	110	w	166	0	233
?	111	x	167	1	240
` (grave)	121	y	168	2	241
:	122	z	169	3	242
£	123	{	192	4	243
@	124	A	193	5	244
' (apostrophe)	125	B	194	6	245
"	126	C	195	7	246
"	127	D	196	8	247
a	129	E	197	9	248
b	130	F	198		249
c	131	G	199		

Example of files and strings

The following example shows the use of strings and files. It replaces each occurrence of the string O\$ within file ENGLISH by the string N\$ and writes the result to the file FRENCH.

```

10 LET O$ = "BUT"
20 LET N$ = "MAIS"
30 FILE *1: "ENGLISH"
40 FILE *2: "FRENCH"
50 GOATEND *1: 500
60 INPUT *1: L$
70 LET X = POS(L$, O$)
80 IF X = 0 THEN 200
90 PRINT *2: SEG$(L$, 1, X-1); N$;
100 LET L$ = SEG$(L$, X+LEN(O$), 999)
110 GOTO 70
200 PRINT *2: L$
210 GOTO 60
500 END

```

Built-in functions

ABS(N)	Absolute value
ATN(N)	Arctangent in radians
CHR(S\$)	Numeric code for single character
CHR\$(N)	Inverse of CHR (.)
CLK\$	Current time of day
COS(N)	Cosine
DAT\$	Current date
DET	Determinant of last array inversion
EXP(N)	e^N
INT(N)	Greatest integer $\leq N$
LEN(S\$)	Length of string
LOG(N)	Log to the base e
POS("PIGGY", "IG")	Gives 2 (position of "IG" in "PIGGY")
RND	Pseudo-random number in [0,1)
SEG\$("PIGGY",2,3)	Gives "IG" (substring)
SGN(N)	1 if $N > 0$; -1 if $N < 0$; else 0
SIN(N)	Sine
SQR(N)	Square root
STR\$(N)	String that PRINT N; would give
TAN(N)	Tangent
TIM	Total process time for job (milliseconds)
VAL(S\$)	Numerical value of S\$

Some useful VME/K commands

DF(NAME=*DJB)	Delete a file
LF(NAME=XYZ)	List a file on the printer
AF(NAME=(LUR099).XYZ, LNAME=ABC)	Assign another user's file

BASIC in VME/K Batch

The following is the VME/K JCL for a simple batch job.

```
JOB(JNAME=RUNBASIC,UNAME=:CUR099,PROFILE,profilename)
BEGIN
INPUT(NAME=TEXT)
----DATA
10 INPUT X, Y
.
.
.
RUN
23, 24
.
.
.
BYE
++++DATA
AF(NAME=TEXT,LNAME=INPUT)
BASIC()
DELETEFILE(NAME=*INPUT)
END
EJ
*****
```