

This reference card lists all the STOS Basic commands with their correct syntax. Parameters placed inside square brackets are optional and can be omitted.

## Naming conventions

All STOS Basic variables must begin with a letter and they must **not** contain any of the following commands:

TO, STEP, THEN, ELSE, XOR, OR, AND, GOTO, GOSUB, MOD, AS

Maximum length of a variable is 31 characters

## Types of variables

Integers P, NUM, T1  
Real numbers P\$, NUM\$, T1\$  
Strings P\$, NUM\$, T1\$

Arrays P(10), NUM\$(3), T1\$(100,10)

Stored in four bytes and range from: -214748348 to +214748348  
Stored in eight bytes and range between: -1.797692 E+308 to +1.797693 E+307  
Always suffixed with a \$ character and can range from 0 - 65500 characters long.  
They are not terminated with a chr\$(0).  
Any of the above variables can be incorporated into a table called an array.  
These arrays can be created using the DIM statement.

## Variable and string commands

=ASC(a\$)  
CHR\$(n)  
DEC var  
DIM var(size),...  
=FLIP\$(n\$)  
INC var  
=INPUT list / "chars",delay  
=INPUT\$(n)  
=INSTR(str1\$,str2\$(,p))  
KEY(x)=a\$  
=LEFT\$(str1\$,n)=  
=LEN(a\$)  
LET x=y  
=MATCH(var(0),value)  
=MID\$(v\$,s,n)=  
=RIGHT\$(str1\$,n)=  
SORT A\$(0)  
=SPACE\$(n)  
=STR\$(n)  
=STRING\$(a\$,n)  
=UPPER\$(n\$)  
=VAL(x\$)

Get Ascii code  
Return Ascii character  
Subtract one from an integer variable  
Dimension an array  
Invert string  
Add one to an integer variable  
Input a number or some text into a string  
Function to input *n* characters into a string  
Search for occurrences of a string within another string  
Function to assign a string to a function key  
Return the leftmost characters of a string  
Get length of string  
Load some information into a variable  
Find the closest match to a value in an array  
Return a string of characters from within a string expression  
Return the rightmost characters of a string  
Sorts all elements in an array  
Create a string full of spaces  
Convert a number to a string  
Create a string full of a\$  
Convert n\$ to upper case  
Convert a string to a number

## Editor and system commands

AUTO [start] [,inc]  
CHANGE a\$ TO b\$ [start-end]  
CLEAR  
CONT  
DEFAULT  
DELETE first-last  
ENGLISH  
FLOAD path\$  
FOLLOW  
FOLLOW OFF  
FOLLOW first-last  
FOLLOW variable list  
FOLLOW variable list,first-last  
FRANCAIS  
=FREE  
FREQUENCY  
FSAVE path\$  
FULL  
GRAB n [first-last]  
=LANGUAGE  
KEYLIST  
LIST  
LIST -last  
LIST first-  
LIST first-last  
LOWER  
MERGE file\$

Automatic line numbering  
Change all occurrences of string in a program  
Clear all the program variables  
Restart a program exited by STOP or Control-C  
Reset the editor and redraw current windows  
Delete some or all of a program  
Select English syntax (Default)  
Load a file using the file selector  
Track through a STOS Basic program  
Turns off any FOLLOW command  
Traces the lines between *first* and *last*  
Traces variables and prints to the screen  
Variables are traced and printed between lines *first* and *last*  
Select French syntax  
Return the amount of *free* memory  
Change scan rate from 50 to 60 Hertz  
Save a Basic file chosen with the file selector  
Expand current window into the full screen area  
Copy all or part of a program segment into the current program  
Variable holding the language used for syntax  
List the current function key assignment  
List the entire program  
List the start of the program to line *last*  
List lines from *first* onwards  
List lines from *first* to *last*  
Change editor mode to lower case  
Merge a file into the current program

MULTI n  
NEW  
RENUM [number][,inc]  
RENUM number,inc,start-end  
RESET  
RUN  
RUN file\$  
RUN line  
SEARCH s\$  
SEARCH s\$,start-end  
SEARCH  
SYSTEM  
UNNEW  
UPPER

Display a number of programs simultaneously  
Erase the current program  
Renummer all of a program  
Renummer part of a program  
Reset the editor  
Run the program starting from the first line  
Load and run a Basic file  
Run the program starting from a specific line  
Searches for a string from the start of a Basic program  
Search for a string between lines *start* and *end*  
Continue search from last match  
Exit back to Gem  
Attempt to recover from a NEW and restore the previous program  
Change listing mode to uppercase

### Program control commands

BREAK ON/OFF  
END  
-ERRL  
-ERRN  
ERROR n  
FOR...NEXT  
GOSUB line / expression  
GOTO line / expression  
IF...THEN [ELSE]  
ON ERROR GOTO line  
ON var GOSUB line1,line2...  
ON var GOTO line1,line2...  
POP  
REPEAT...UNTIL  
RESUME  
RESUME NEXT  
RESUME n  
RETURN  
STOP  
WHILE...WEND

Turn on or off the Control+C break key  
Exit from the program  
Variable holding the location of the last error  
Variable containing the number of the last error  
Generate an error and return to the STOS editor  
Repeat a section of code a specific number of times  
Jump to a subroutine  
Jump to a new line number  
Choose between alternative actions  
Trap an error within a Basic program  
GOSUB one of a list of routines depending on a variable  
Jump to one of a list of lines depending on a variable  
Remove the Return information after a GOSUB  
Repeat a section of code until a condition is true  
Jump back to the statement which caused the error and try again  
Jump to statement following the one which caused the error  
Resume the program at the line number *n*  
Return from a subroutine and execute the next instruction  
Stop running the program and return to the editor  
Repeat a section of code while a condition is true

### Loading and Saving

ACCLoad file\$  
ACCLoad \*\*\*  
BLOAD file\$,addr/#bank  
BSAVE file\$,start TO end  
LOAD file\$  
SAVE file\$

Load an accessory  
Load all accessories from the current disc  
Load binary information into a specified address or bank  
Save a block of memory between *start* and *end* in binary format  
Load part or all of a STOS Basic program  
Save part or all of a STOS Basic program

### Legal file extensions

.BAS  
.ACB  
.PI1, PI2, PI3  
.NEO  
.MBK  
.MBS  
.VAR  
.ASC  
.PRG

Basic programs  
Accessories (Use ACCLoad)  
Degas picture format  
Neochrome format  
One memory bank  
All current banks  
Currently defined variables  
Ascii format  
Executable from Gem Desktop

### Directories

=DFREE  
=DIR FIRST\$(path\$,flag)  
=DIR NEXT  
DIR [path\$]  
DIR\$=path\$  
DIR\W[path\$]  
=DRIVE  
=DRIVE\$=  
=DRVMAP  
=FILESELECT\$(path\$[title\$[border]])  
KILL file\$  
MKDIR folder\$  
PREVIOUS  
RENAME old\$TO new\$  
RMDIR folder\$

Variable containing the free space on the current disc  
Get first file in directory satisfying pathname  
Get the next file satisfying current pathname  
List out the directory of the current disc  
Set the current directory  
List the directory across the screen  
Variable containing the current drive  
String variable holding current drive  
Variable holding a list of the drives connected  
Select a file using the file selector  
Erase a file from the disc  
Create a new folder  
Sets the current path up one directory  
Rename a file  
Delete a folder

### Filing commands

CLOSE #channel  
=EOF (#channel)  
FIELD #channel,len1 AS field1\$,...  
GET #channel,r  
INPUT #channel,variable list  
=INPUT\$(#channel,count)  
LINE INPUT #channel,separator\$,  
variable list  
LINE INPUT #channel,variable list  
=LOF (#channel)  
OPEN #channel,"AUX"

Close a file  
Test for end of file  
Define record structure  
Input a record from a random access file  
Input a list of variables from a file or device  
Inputs a number of characters from a device  
  
Input a list with a different separator to ","  
Inputs a list of variables not separated by ","  
Length of open file  
Open a channel to the RS232 port

OPEN #channel,"MIDI"  
 OPEN #channel,"PRT"  
 OPEN #channel,"R",file\$  
 OPEN IN #channel,file\$  
 OPEN OUT #channel,file\$[,attribute]  
 =POF (#channel)=  
 =PORT (#channel)  
 PRINT #channel,variable list  
 PUT #channel,r

Open a channel to the MIDI interface  
 Open a channel to the printer  
 Open a random access file  
 Open a file for input  
 Open a file for output  
 Variable holding current position of file pointer  
 Function to test if channel waiting  
 Print a list to a file or device  
 Output a record to a random access file

### Memory bank commands

BCOPY #source TO #dest  
 BGRAB prgno [,b]  
 ERASE b  
 HEXA ON/OFF  
 =LENGTH(b)  
 =LENGTH(prgno,b)  
 LISTBANK  
 RESERVE AS DATA bank,length  
 RESERVE AS DATASCREEN bank  
 RESERVE AS SCREEN bank  
 RESERVE AS SET bank,length  
 RESERVE AS WORK bank,length  
 =START (b)  
 =START (prgno,b)

Copy the contents of a bank to another bank  
 Copy some or all banks from a program to the current program  
 Delete a bank  
 Toggle hexadecimal listing  
 Get the length of a bank  
 Get the length of a bank from another program  
 List the banks in use  
 Reserve a permanent bank as a workspace  
 Reserve a permanent bank of memory as a screen  
 Reserve a temporary bank as a screen  
 Reserve a permanent bank of memory for use as a character set  
 Reserve a temporary bank as a workspace  
 Get the start address of a bank  
 Return the start address of a bank from a program

### Text commands

CDOWN  
 CENTRE a\$  
 CLEFT  
 CRIGHT  
 CUP  
 CURS ON/OFF  
 HOME  
 INVERSE ON/OFF  
 LOCATE x,y  
 PAPER index  
 PEN index  
 =SCRN (x,y)  
 SET CURS top,base  
 SHADE ON/OFF  
 SQUARE wx,hy,border  
 TAB(n)  
 UNDER ON/OFF  
 WRITING effect  
 =XCURS  
 =XGRAPHIC(x)  
 =XTEXT(x)  
 =YCURS  
 =YGRAPHIC(y)  
 =YTEXT(y)

Cursor down  
 Print a line of text centred on the screen  
 Cursor left  
 Cursor right  
 Cursor up  
 Enable/disable text cursor  
 Cursor home  
 Switch the pen and ink values  
 Position the cursor  
 Set colour of the background text  
 Set colour of the text  
 Return the character at the position x,y  
 Set text cursor size  
 Shade all subsequent text, or return to normal printing  
 Draw a rectangle at the current cursor position  
 Move the cursor to the right by n places  
 Set underline mode on or off  
 Change text writing mode (1=replacement 2=OR (Default) 3=XOR)  
 Variable holding the X coordinate of the text cursor  
 Convert x text to x graphic coordinate  
 Convert x graphic to x text coordinate  
 Variable holding the Y coordinate of the text cursor  
 Convert y text to y graphic coordinate  
 Convert y graphic to y text coordinate

### Window commands

BORDER n  
 CLW  
 QWINDOW n  
 SCROLL DOWN  
 SCROLL ON/OFF  
 SCROLL UP  
 TITLE a\$  
 WINDEL n  
 WINDMOVE x1,y1  
 =WINDOW  
 WINDOWPEN n,x1,y1,w,h[,border[,set]]  
 WINDOW n

Set the border of the current window  
 Clears the current window  
 Activate window without redrawing it  
 Scroll the current window down one line  
 Switch window scrolling on and off  
 Scroll the current window up  
 Define a title for the current window  
 Delete a window  
 Move a window  
 Variable containing number of the current window  
 Create a window  
 Activate window

### Graphics functions

ARC x1,y1,r,start,end  
 BAR x1,y1 TO x2,y2  
 BOX x1,y1 TO x2,y2  
 CIRCLE x1,y1,r  
 CLIP x1,y1 TO x2,y2  
 COLOUR index,\$RGB  
 =COLOUR(index)  
 DIVX  
 DIVY  
 DRAW x1,y1 TO x2,y2  
 DRAW TO x3,y3  
 EARC x1,y1,r1,r2,start,end  
 ELLIPSE x1,y1,r1,r2  
 EPIE x1,y1,r1,r2,start,end  
 FLASH index,\*(col,delay)(col2,delay)...\*  
 GR WRITING x  
 INK index

Draw a circular arc  
 Draw a filled rectangle  
 Draw a hollow rectangle on the screen  
 Draw a filled circle  
 Restrict all graphics to part of the screen  
 Assign a colour to an index  
 Read the colour assignment  
 Variable fraction for x axis plotting  
 Variable fraction for y axis plotting  
 Draw a line between x1,y1 and x2,y2  
 Draw a line from the last point to x3,y3  
 Draw an elliptical arc  
 Draw a filled ellipse  
 Draw an elliptical pie  
 Set flashing colour sequence  
 Set method of drawing (1=replacement 2=transparent 3=XOR mode 4=Inverse transparent)  
 Set colour of graphic drawing operations

=MODE n  
 PAINT x1,y1  
 PALETTE \$RGB,\$RGB...  
 PIE x1,y1,r,start,end  
 PLOT x,y,[index]  
 POINT(x1,y1)  
 POLYGON x1,y1 TO x2,y2 TO x3,y3...  
 POLYLINE x1,y1 TO x2,y2 TO x3,y3...  
 POLYMARK x1,y1;x2,y2;x3,y3;...  
 RBAR x1,y1 TO x2,y2  
 RBOX x1,y1 TO x2,y2  
 SET LINE mask,thick,start,end  
 SET MARK type,size  
 SET PAINT type,pattern,border  
 SET PATTERN a\$  
 SET PATTERN address of pattern  
 SHIFT delay [,start]

#### Screen commands

APPEAR x[,y]  
 =BACK=  
 CLS  
 CLS scr  
 CLS scr,col  
 CLS scr,col,x1,y1 to x2,y2  
  
 DEF SCROLL n,x1,y1 TO x2,y2,dx,dy  
 DEFAULT BACK  
 DEFAULT LOGIC  
 DEFAULT PHYSIC  
 FADE speed  
 FADE speed TO sbank  
 FADE speed,col1,...,col4,...  
 GET PALETTE(n)  
 =LOGIC=  
 PACK scr,bnk  
 =PHYSIC=  
 REDUCE scr1 TO [scr2],x1,y1,x2,y2  
 RESERVE AS DATASCREEN n  
 RESERVE AS SCREEN n  
 SCREEN COPY scr1 TO scr2  
 SCREEN COPY scr1,x1,y1,x2,y2  
 TO scr2,x3,y3  
 SCREEN SWAP  
 SCREEN\$(scrn,x,y)=a\$  
 =SCREEN\$(scrn,x1,y1 TO x2,y2)  
 SCROLL n  
 SYNCHRO  
 SYNCHRO OFF  
 SYNCHRO ON  
 UNPACK bnk,scr  
 WAIT VBL  
 ZOOM scr1,x1,y1,x2,y2 TO  
 [scr2],x3,y3,x4,y4

#### Sprite commands

ANIM FREEZE [n]  
 ANIM ON/OFF [n]  
 ANIM n,a\$  
 AUTOBACK ON/OFF  
 COLLIDE (n,w,h)  
 =DETECT(n)  
 FREEZE  
 GET SPRITE x,y,i,[mask]  
 LIMIT SPRITE x1,y1 TO x2,y2  
 MOVE FREEZE  
 MOVE ON/OFF [n]  
 MOVE X n,m\$  
 MOVE Y n,m\$  
 MOVEON (n)  
 OFF  
 PRIORITY ON/OFF  
 PUT SPRITE n  
 REDRAW  
 RESET ZONE [n]  
 SET ZONE z,x1,y1 TO x2,y2  
 SPRITE n,x,y,p  
 UNFREEZE  
 UPDATE [ON/OFF]  
 =X SPRITE (n)  
 =Y SPRITE (n)  
 ZONE (n)

#### Character and Icon commands

CHARCOPY s TO b

Change the graphics mode  
 Contour fill  
 Set the current screen colours  
 Produce a pie chart  
 Plot a single point  
 Get the colour of a point  
 Draw a filled polygon  
 Multiple line drawing  
 Plot a list of polymarkers  
 Draw a filled rounded rectangle  
 Draw a rounded hollow box  
 Set the line styles  
 Set the marker used by polymark  
 Select fill pattern  
 Set user defined pattern from data in a string  
 Set a user-defined fill pattern with data at address  
 Colour rotation

Fade between two pictures  
 Address of the background screen  
 Clear the whole screen  
 Clears the screen at scr  
 Fills the screen scr with colour col  
 Replaces the rectangle at scr at coordinates x1,y1,x2,y2 with a block of colour col  
 Define a scrolling zone  
 Return the initial value of BACK  
 Return the initial value of LOGIC  
 Return the initial value of PHYSIC  
 Fade all colours to black  
 Fade present colours to those of the screen in bank sbank  
 FADE separate colours to new values  
 Set the palette from a screen bank  
 Address of the logical screen  
 Pack a screen  
 Address of the physical screen  
 Reduce a screen to a smaller size  
 Reserve a permanent screen  
 Reserve a bank as a temporary screen  
 Copy screen scr1 to screen scr2

Copies a section of a screen to a point in another screen  
 Swaps the address of the logical and physical screens  
 Copy screen area in a\$ to the screen scrn  
 Load an area of a screen into a string  
 Scroll a screen zone  
 Execute all sprite movements once  
 Turn off normal sprite interrupt  
 Revert the sprite movements to normal  
 Unpack a screen  
 Wait for a vertical blank

Magnify a section of the screen

Freeze an animation  
 Start an animation  
 Animate a sprite  
 Set screen for graphics operation  
 Detect collisions between two sprites  
 Find colour of pixel underneath sprite  
 Pause sprite and music operations  
 Copy a section of the screen into the sprite bank  
 Limits sprite to a specific area  
 Temporarily suspend sprite movements  
 Start/stop sprite movement  
 Move a sprite horizontally  
 Move a sprite vertically  
 Return sprite state  
 Turn off sprites  
 Change between priority modes  
 Put a copy of a sprite on the screen  
 Redraw the sprites  
 Erase a zone  
 Set a zone for testing  
 Displays a sprite on the screen  
 Restart sprite and music operations  
 Change automatic sprite updates  
 Get X coordinate of sprite  
 Get Y coordinate of sprite  
 Tests a sprite to see if it is in a zone

Copy a character set into a particular bank

=CHARLEN(n)  
 ICON\$(n)  
 RESERVE AS SET n,len

Get the length of a character set  
 Generate an icon at the current cursor position  
 Reserve a bank of memory for a character set

### Mouse and joystick commands

CHANGE MOUSE m  
 =FIRE  
 HIDE [ON]  
 =JDOWN  
 =JLEFT  
 =JOY  
 =JRIGHT  
 =JUP  
 LIMIT MOUSE x1,y1 TO x2,y2  
 =MOUSE KEY  
 SHOW [ON]  
 =X MOUSE  
 =Y MOUSE

Change the shape of the mouse pointer  
 Test fire button state  
 Remove mouse pointer from the screen  
 Test joystick movement down  
 Test joystick movement left  
 Read joystick  
 Test joystick movement right  
 Test joystick movement up  
 Limit mouse to a section of the screen  
 Get status of the mouse keys  
 Activate the mouse pointer  
 Get the X coordinate of the mouse pointer  
 Get the Y coordinate of the mouse pointer

### Music and Sound commands

BELL  
 BOOM  
 CLICK ON/OFF  
 ENVEL type,speed  
 MUSIC FREEZE  
 MUSIC OFF  
 MUSIC ON  
 MUSIC n  
 NOISE v,p  
 PLAY [voice,]pitch,duration  
 =PVOICE(v)  
 SHOOT  
 TEMPO s  
 TRANSPOSE df  
 VOICE [v]  
 VOLUME [v,]intensity

Simple bell sound  
 Generate an explosion sound  
 Turn off keyboard click  
 Activate an envelope  
 Temporarily stop a piece of music  
 Turn off music  
 Restart a piece of music  
 Play tune number n  
 Produce rushing wind effects  
 Play a note  
 Return position in music  
 Create a noise like a gun firing  
 Change the speed of a sample of music  
 Change the pitch of a piece of music  
 Turn on/off a voice  
 Change the sound volume

### Menu commands

MENU FREEZE  
 MENU OFF  
 MENU ON [border][,mode]  
 =MENU\$(title,option)OFF  
 =MENU\$(title,option)ON  
 MENU\$(x,y)=title\$ [,paper,pen]  
 MENU\$(x,y)=option\$ [,paper,pen]  
 =MNBAR  
 =MNSELECT  
 ON MENU GOTO lin1,line2,...  
 ON MENU OFF  
 ON MENU ON

Freeze the actions of the menu  
 Switch off the entire menu  
 Turn on the menu  
 Turn off an option  
 Turn back on an option  
 Set a menu title for a drop down menu  
 Set up an option in a menu  
 Holds the menu title that has been selected  
 Holds the option number which has been selected  
 Go to a line dictated by the option selected  
 Turn off on menu option  
 Start on menu system

### Arithmetic operations (Listed in order of priority)

^ Power  
 / and \*  
 MOD  
 + and -  
 AND  
 OR  
 XOR

Power  
 Divide and multiply  
 Modulo operator (Produces remainder of a division)  
 Plus and minus  
 Logical AND  
 Logical OR  
 Logical eXclusive OR

### Mathematical functions

=ABS(y)  
 DEF FN name[(variable list)]=expression  
 =EXP(y#)  
 =FN name[(variable)]  
 =INT(y#)  
 =LN(y#)  
 =LOG(y#)  
 =MAX(x,y)  
 =MIN(x,y)  
 =SGN(y)  
 =SOR(y)  
 =SWAP(x,y)

Absolute value  
 Create a user-defined function  
 Exponential function  
 Call a user-defined function  
 Convert floating point to an integer  
 Natural logarithm  
 Logarithm  
 Get the maximum of two values  
 Return the minimum of two values  
 Find the sign of a number  
 Square root  
 Swap the contents of two variables

### Trigonometric functions

=ACOS(number)  
 =ASIN(number)  
 =ATAN(number)  
 =COS(angle)  
 =DEG(angle)  
 =HCOS(angle)  
 =HSIN(angle)  
 =HTAN(angle)  
 =PI π  
 =RAD(angle)

Arc cosine  
 Arc sine  
 Arc tangent  
 Cosine  
 Convert an angle expressed in radians to degrees  
 Hyperbolic cosine  
 Hyperbolic sine  
 Hyperbolic tangent  
 A constant  
 Convert an angle expressed in degrees to radians

=SIN(angle)  
=TAN(angle)

Sine  
Tangent

### The printer

HARDCOPY  
LDIR [path\$]  
LISTBANK  
LLIST [linestart[-line end]]  
LPRINT variable list  
WINDCOPY

Screen dump  
List a directory to the printer  
Prints a list of the banks used by your program on the printer  
Print part or all of a program on a printer  
Output a list of variables to the printer  
Window dump

### Other commands

=ACCNB  
ACCNEW  
=AREG(n)=  
BCHG(x,y)  
BCLR(x,y)  
=BIN\$(x[,y])  
BSET(x,y)  
BTST(x,y)  
CALL address  
CLEAR KEY  
COPY start,finish TO destination  
DATA variable list  
=DATE\$=  
=DEEK(address)  
DOKE address,value  
=DREG(n)=  
=FALSE  
FILL start TO finish,longword  
FIX(n)  
=FKEY  
=HEX\$(x[,n])  
=HUNT(start TO end,a\$)  
=INKEY\$  
KEY SPEED repeatspeed,delay  
=LEEK(address)  
LINE INPUT variable list  
LOKE address,number  
LOWERS\$(n\$)  
=NOT(x)  
=PEEK(address)  
POKE address,x  
PRINT or ? variable list  
=PSG(n)=  
PUT KEY  
READ variable list  
REM or '  
RESTORE line / expression  
=RND(y)  
ROL x,y  
ROR x,y  
=SCANCODE  
=TIME\$=  
=TIMER=  
TRAP n[ parameters]  
=TRUE  
USING format\$,variable list

Get accessory number  
Remove all currently installed accessories  
Variable used to pass information to the 68000's address registers  
Change a bit  
Clear a bit  
Convert number to binary string  
Set a bit to 1  
Test a bit  
Call a machine-code program - stored at address  
Initialise the keyboard buffer  
Copy a memory block  
Place a list of data items in a program  
Get date  
Get word at address  
Change word at address  
Variable used to pass information to the 68000's data registers  
Logical false 0  
Fill memory block with a longword  
Set precision  
Read the function keys directly  
Convert number to hexadecimal  
Find a string in memory  
Function to get a keypress  
Change key repeat speed and initial delay response  
Get long word at address  
Input a list of variables separated by a return  
Change long word at address  
Convert n\$ to lower case  
Logical NOT operation  
Get byte at address  
Change byte at address  
Print a list of variables to the screen  
Access the programmable sound generator  
Put a string into the keyboard buffer  
Read some data from a DATA statement into a variable  
Remark  
Set the READ pointer to a new line  
Random number generator  
Rotate left  
Rotate right  
The Scan Code of the last key input with INKEY\$  
Get time  
Count in 50ths of a second  
Call a 68000 trap function  
Logical true -1  
Formatted output  
~ Replacement character  
# Number of digits of a variable  
+ Adds a plus to a positive number or a minus sign to a negative number  
- Produces only a minus sign if a value is negative  
. Places a decimal point in a number  
, Centres a number but does not output a decimal point  
^ Prints a number in exponential format  
Get address of a variable  
Wait for a keypress  
Wait in 50ths of a second

=VARPTR(variable)  
WAIT KEY  
WAIT x

(c) Mandarin/Jawx 1988

**MANDARIN**  
**SOFTWARE**

Europa House,  
Adlington Park,  
Adlington,  
Macclesfield SK10 4NP.