

The Cracker

user's manual

Software Technology Limited

PO Box 724, BIRMINGHAM, B15 3HQ

USING THIS MANUAL WITH THE TATUNG EINSTEIN

Your version of THE CRACKER has already been prepared to take full advantage of the features of the EINSTEIN. Please ignore all references to the INSTALL program.

The screen display in this manual is based on 60 columns screen width and is laid out in a form suitable for printing. You will notice that the your own screen display is laid out to take advantage of the colour capabilities of your computer. With your 40 column display the prompting is much the same but there are few slight rearrangements to make sure that there is room for the prompts. The primary command list 'ABCDEFGHIJLMNOP.....' is replaced by 'Command'.

Your Einstein computer has minor differences with the character set used in the preparation of this manual. The following characters have been replaced by alternatives on the EINSTEIN keyboard.

Character in manual	replacement character	meaning
]	←	less than or equal
[→	greater than or equal
:		not equal
^	↑	to the power of
\	-	switch vertically between windows

References to CP/M should be read as references to Xtal-DOS. The two operating systems behave in exactly the same way while running THE CRACKER.

THE CRACKER
CONTENTS

OVERVIEW.....	1
TUTORIAL I	
GETTING STARTED.....	3
ENTERING A SIMPLE EXPRESSION.....	7
ENTERING TEXT.....	9
THE KEYBOARD.....	13
MOVING THE CURSOR.....	14
WHAT TO DO WHEN YOU MAKE A MISTAKE.....	16
TUTORIAL II, BASIC TECHNIQUES	
MORE ABOUT ENTERING TEXT.....	17
EDITING.....	22
MORE ABOUT EXPRESSIONS.....	28
CHANGING THE LAYOUT.....	35
CHANGING COLUMN WIDTHS AND DEFAULT FORMAT.....	40
INSERTING AND DELETING (ZAP).....	41
USING SECTIONS OF THE WORKSHEET.....	44
BLANKING.....	49
USING THE MOVE COMMAND.....	51
A FURTHER LOOK AT THE COPY COMMAND.....	53
THE ORDER OF CALCULATION.....	61
PARTITIONING THE SCREEN.....	62
USING MACRO COMMAND GROUPS.....	70
TUTORIAL III, USE OF FUNCTIONS	
MORE ON FUNCTIONS.....	74
THE IF, THEN, ELSE FUNCTIONS.....	75
TABLE HANDLING FUNCTIONS.....	78
A NOTE ON LISTS.....	82
MATHEMATICAL FUNCTIONS.....	83
UTILITY FUNCTIONS.....	85
STATISTICAL FUNCTIONS.....	87
DEFINED FUNCTIONS.....	89
TUTORIAL IV, DATABASE HANDLING TECHNIQUES	
SEARCHING.....	92
SORTING THE LINES.....	94
PREPARATION OF MAIL LABELS.....	97
TUTORIAL V, ADVANCED TECHNIQUES	
FUNCTIONS THAT ALLOW LOOPING.....	102
SUBROUTINES USING THE DO FUNCTION.....	106
LOOPING USING THE FUNCTIONS DO AND WHILE.....	110
TABLE FILLING USING THE DO FUNCTION.....	115

THE CRACKER
CONTENTS

COMMAND REFERENCE		
DEFINITION OF TERMS.....	126	
AUTOMATIC CALCULATION COMMAND.....	128	
BLANK COMMAND.....	129	
CONTROL CODE COMMANDS.....	131	
COPY COMMAND.....	132	
CURSOR MOVEMENT.....	135	
EDIT COMMAND.....	136	
EXCHANGE COMMAND.....	138	
EXPRESSION ENTRY.....	139	
ARITHMETIC OPERATORS.....	141	
LOGICAL OPERATORS.....	141	
BUILT IN FUNCTIONS.....	142	
TRIGONOMETRIC FUNCTIONS.....	142	
LOGICAL FUNCTIONS.....	143	
UTILITY FUNCTIONS.....	144	
FORMAT COMMAND.....	148	
FORMAT TYPES.....	149	
OVERFLOW.....	151	
GET COMMAND.....	152	
INSERT COMMAND.....	153	
JUMP COMMAND.....	155	
TEXT ENTRY.....	156	
MOVE COMMAND.....	157	
NEW COMMAND, OUT COMMAND.....	158	
PARTITION COMMAND.....	159	
QUIT COMMAND.....	160	
SORT COMMAND.....	161	
VERIFY AND CALCULATE COMMAND.....	162	
WHICH FILES COMMAND.....	163	
ZAP (DELETE) COMMAND.....	164	
* MACRO COMMAND.....	165	
MORE DETAIL ON ERROR MESSAGES.....		166
MORE DETAIL ON PROMPTS.....		171
.....		173
INDEX.....		179

THE CRACKER OVERVIEW

- The computer and this program are here to make the tasks you find difficult, repetitive or boring - enjoyable.
- THE CRACKER is very powerful but is just as happy multiplying 2 by 2 as solving engineering calculations.
- You can easily get started but with familiarity you will be able to handle problems that will improve your job performance.

You are probably wondering just what the THE CRACKER is. It is a spreadsheet program. This name has arisen because of the similarity with a balance sheet or calculation sheet which is laid out as rows or columns. Within the computer this sheet can be very large and so you are able to use the screen to scan the sheet to view the area in which you are interested.

This sheet is really two sheets lying on top of one another. The top one shows all the results neatly presented. The underlying one holds the rules and data used in the calculation of those results.

The aim of THE CRACKER is to allow you to use the computer as an everyday tool to take care of your data handling and calculation problems. THE CRACKER is so flexible that you will find that most of the time you can use it instead of buying dedicated programs.

Once you have recorded some information on the sheet you can manipulate it using CRACKER commands. Text or value entries may

be edited, copied, moved, deleted, sorted, printed, filed and may be output as mail labels. Value entries may also be calculated using formulae or rules that you build in. These rules are called expressions. For example a rule might say 'add up this column' and the display will show the result of the addition.

No program language needs to be learnt to use THE CRACKER and yet with it you can do most day to day calculations and book-keeping type operations. Files can be created stored and merged. Small sections of your worksheet can be filed and consolidated into other worksheets.

You refer to every cell on the sheet by a column letter and row number. Rules and formulae can use the values stored in any cell by quoting its letter and number. For example you may wish to put the value 2 into the top left cell which is numbered A1 and then find out what the value is when added to 3. To do this you set up a formula A1+3 which says add 3 to the value stored in A1. The result 5 is displayed. If you now change the 2 at A1 to 4 then the result immediately changes to 7.

This facility to do immediate recalculations is very powerful and accounts for the popularity of this type of program. If you have a complex set of figures and want to see the effect of changing one of the values this is very easy. It is so easy that you can do 'what if' assessments by just trying the values to get a feel of the situation.

Using this manual

This manual is divided into two main sections. The first is the tutorial section which you can use to get familiar with THE CRACKER. The other is a reference section which details all the options and commands open to you.

The tutorial section is designed so that you build on and use only the features to which you have been introduced. Do not worry about the reference section until you are familiar with the tutorial section. Not all the options open to you are explained in the tutorial and so as you get more confident study the reference section.

THE CRACKER TUTORIAL I

GETTING STARTED

If THE CRACKER has not been installed on your machine please start by reading the guide to installation and following the procedure outlined. Then bring up CP/M according to your system instructions. Make the disk with THE CRACKER the current disk and type CRACKER. You will see a screen something like this:

Contents line
Prompt line
Edit line

* C,I,W
Next: _

9780
Auto.

THE CRACKER

copyright (C)
1984 I.W.Searle
80-100000 V1.22

THE CRACKER TUTORIAL I
GETTING STARTED

The flashing character cursor (the cue, shown here as an underline) will be on the entry line. Above the cue is the prompt C,I,W which is the selection of options that you can use next. The main worksheet will be formed below the entry line.

Initially the sheet does not exist, you will be building it up as required. The first command you will be using is I for Insert. Type the following sequence IC20<R>G<R> and as you type note that each letter is shorthand for a full word. The <R> is only one keystroke and is shorthand for the RETURN or ENTER key. You will see the screen much as this:

*
Next: Insert column , width 20_

9780
Auto.

The prompt line gives you the keystroke options that are available at each stage. This dynamic prompting means that you always know which are the valid commands. The # symbol means that a numeral may be entered.

*
Next: Default Format to be General_

9780
Auto.

Do not worry yet about what it all means. Now type I10L<R><R>. The screens you will see are:

A1
G *
Next: Insert 10 lines_
=====A=====

9780
Auto.


```

A1
G *      U,D,L,R,arrows,J,<R>,E      9780
Next: Destination_                    Auto.
-----A-----
1(                                     )

```

```

A1
G * ABCDEFGI JLMNPQRSUVWXZ./+>.* arrows      9780
Next: _                                         Auto.
-----A-----
1(                                     )
2
3
4
5
6
7
8
9
10
11

```

You now have created enough worksheet to begin experimenting. If you make a mistake just press the ESC key and start the sequence again. No harm will be done. The numbers running down the screen are the worksheet line numbers and the A above them is the worksheet column letter. The cell cursor is shown as a pair of brackets. On your screen it may be different. This cell cursor marks the active cell. A cell is a single unit of available space on the worksheet.

Each has a name, or coordinate reference, which you get by taking the column letter and the line number. At the moment the cursor is in column A and line 1 so the coordinate reference is A1. As this is the active cell you will see this coordinate at the top left of the screen.

You will now get a short description of the other items at the top of the screen. The G beneath the cell coordinate tells you the cell will use the general format. The asterisk * next to the G says the storage area for the commands is empty. Over on the right of the screen there is the number 9780 which is the number of bytes memory you have left. Note that this number has not changed since you created the sheet. THE CRACKER will keep track

of this and will not let you lose your work if all the memory is used up. Beneath the memory is the word Auto. which tells you the calculation will be automatic. You have now been introduced to most of the control panel.

Try moving the cursor by typing D for Down. Note that active cell coordinate at the top changes as the cursor moves down. Move the cursor back up again with U for Up.

ENTERING A SIMPLE EXPRESSION

Type '.' and the screen will change to:

```

A1
G * Enter number or expression          9780
>> _                                     Auto.
-----A-----
1(                                     )
2

```

THE CRACKER is now ready to accept some form of number or expression. So why not try it, enter a number 2.3 and press <R> (RETURN or ENTER). If you make a mistake press the DEL key and the cue will backspace removing the last character. You can always press the ESC key and go back to the primary command situation. Your screen will look like this:

```

A1      (2.3)
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\>.* arrows  9759
Next: _                                     Auto.
-----A-----
1(                                     2.3)
2

```

Note that the contents line at the top now has your entry

enclosed in (). This is an indication that the entry is a number or expression. Note also that the memory has gone down a little.

You are next going to see the power of THE CRACKER. Move the cell cursor down with the D key and press '.' again. This time enter 2+A1<R> which is an expression meaning 2 plus the current value at coordinate A1. You will now see:

```

-----
A2      (2+A1)
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\>.* arrows  (1)
Next: _                                     9737
-----A-----                               Auto.
1                                     2.3
2(                                     4.3)
3

```

Go back to A1 and change the value and you will note that A2 changes at the same time. You have managed to enter an expression that uses a value from another cell and this expression works however you change that cell. You will note that there is a number (1) in the top right hand corner of the screen. This is the order of calculation number for which you will have an explanation later.

Now look at this list to remind yourself of some of the features you have seen.

I	Is used to insert columns and lines
U	moves the cell cursor up.
D	moves the cell cursor down.
.	allows you to enter a number or expression.
<R>	stands for the RETURN or ENTER key.
ESC	press this key to abandon the current operation, without harm.
DEL	use this key to backspace while entering an expression.
A1,A2	are cell coordinates and may be used as if they are numbers in an expression.

It will now be coming apparent to you that you will be able to build complex expressions change the data and always come out with a valid answer. You can test the effect of different values on an answer and so do 'what if?' assessments of a situation.

ENTERING TEXT

Now you are going to create another column and insert some text.
Type IC20<R><R>TL<R>. You will see screens like these:

```

A2      (2+A1)
G      *
Next: Insert column , width 20_
-----A-----
1              2.3
2(              4.3)
3
    
```

(1)
9748
Auto.

```

A2      (2+A1)
G      *
Next: Destination _
-----A-----
1              2.3
2(              4.3)
3
    
```

(1)
9748
Auto.

```

A2      (2+A1)
G      *
Next: Default Format to be Text right justified_
-----A-----
1              2.3
2(              4.3)
3
    
```

(1)
9748
Auto.

```

A2      * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Tr
Next: _
-----A-----B-----
1              2.3
2(              4.3)
3
4
5
6
7
8
9
10
11
    
```

9748
Auto.

Column A is now set up for you to put in text. The original column has been shifted to the right and labelled B. Before you go any further move the cursor to location B2 which has your expression. Use the R key to move the cursor right. If you look at the top line of the screen you will see that your A1 has now changed to B1 to keep track of the effect of entering another column. So your expression will still work.


```

B2      (2+B1)
G * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----B-----
1
2          (          2.3
3          4.3)
    
```

(1)
9748
Auto.

Go back to A1 using the L key for left and the U key for up.
Type '.' and you will see this:

```

A1
Tr * Enter characters
>> _
-----A-----B-----
1(          )          2.3
2          4.3
3
    
```

9748
Auto.

Now enter some text, type 'First number'.

```

A1
Tr *
>> First number_
-----A-----B-----
1(          )          2.3
2          4.3
3
    
```

9748
Auto.

```

A1      First number
Tr * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----B-----
1(          First number) 2.3
2          4.3
3
    
```

9728
Auto.

You have now tried both expression entry and text entry.

H moves the cell cursor right
L moves the cell cursor left

Now for more detail on the keyboard and cursor movement.

THE KEYBOARD

You probably know about upper and lower case characters. You hold down a shift key while typing a letter to get the upper case capital form. You may not know that there is a second form of shift key which is marked CTRL for control. This is held down in the same way as the shift while a letter is typed. CTRL-H means hold down the CTRL and type the letter following the dash. The ESC or escape key is well named and very important with THE CRACKER. It is used as an exit without harm back to the state of waiting for a primary command.

It may be that you may be able to enter commands faster than the cracker can process them. THE CRACKER has a built in 'type ahead facility' which stores these commands until they can be processed. If the commands can not be accepted you will hear a beep and whatever you typed will be ignored. The asterisk * on the left near the top of the screen means that there are no commands waiting to be done.

MOVING THE CURSOR

U, up arrow, CTRL-W
D, down arrow, CTRL-Z
L, left arrow, CTRL-A
R, right arrow, CTRL-D

Any of these move the cell cursor up one cell.
These move the cell cursor down.
You can move the cell cursor left with these.
Right movement of the cell cursor.
Moves the cell cursor down one page.
Moves the cell cursor up one page.

You can use any of the above commands to move the cursor. You probably find that you will be using one set in certain circumstances and another one at other times. CTRL-W means hold down the CTRL key while typing W.

If you have to move the cursor some distance you will find it easier to use the Jump command. This command goes directly to the coordinate that you specify. Type JB2<R> and you will see:

```

A1 First number
Tr * crd <R>
Next: Jump to _
=====A=====B=====
1( First number) 2.3
2 4.3
3

```

9728
Auto.

A1	First number	
Tr	#,<R>	
Next:	Jump to B2_	
	-----A-----B-----	
1(First number)	2.3
2		4.3
3		

9728
Auto.

WHAT TO DO WHEN YOU MAKE A MISTAKE

You will always know when you have made a mistake because THE CRACKER will give you a beep and an explanation of your error. THE CRACKER is set up so that whatever the mistake you make you will do no harm to your work. When you are entering an expression just press the DEL key to back up the cue to the place before your mistake. THE CRACKER usually checks for a mistake as each key is pressed and so you will rarely have to make long corrections. If you change your mind about an entry and want to abandon what are doing just press the ESC key. This will stop the current entry and return the worksheet to the exact state before you started the entry.

When you get an error message and are not entering an expression then just type ESC or <R>. This will clear the message and reset the worksheet ready for a primary command.

THE CRACKER TUTORIAL II

BASIC TECHNIQUES

MORE ABOUT ENTERING TEXT

You are now going to get more detail about entering text into the worksheet. Text data can not be processed to a numerical value. This negative form of definition is necessary because you could enter a mathematical expression into a text cell but you would not get a value. Text is used for headings, explanatory labels or for information in a database such as names and addresses.

There are two normal types of text entry. Justified in the column width is the first type. The second is heading which will overwrite the next column if there is not enough space.

Try an experiment with the various text types. Move the cursor back to A1. Type FTL<R> and you will see:

```

A1  First number
Tr  *           #,C,F,G,H,I,T,P
Next: Format to be _
=====A=====B=====
1(      First number)          2.3
2                                           4.3
3
  
```

9728
Auto.

THE CRACKER TUTORIAL II

MORE ABOUT ENTERING TEXT

```

A1  First number
Tr  *           <R>
Next: Format to be Text left justified_
=====A=====B=====
1(      First number)          2.3
2                                           4.3
3
  
```

9728
Auto.

Type a few characters such as 'Test text' and you will have:

```

A1  First number
Tr  * Enter characters
Next: Text test_
=====A=====B=====
1(      First number)          2.3
2                                           4.3
3
  
```

9728
Auto.

Type <R> and your entry will be transferred to the worksheet and replace the previous entry. Up to this point you could have pressed ESC and got 'First number' back. Note that you increased your remaining memory by 3 bytes because the new entry is 3 characters shorter. The entry is displayed on the left of the cell:

```

A1  Text test
T1  * ABCDEFGIJKLMNPQRSUVWXYZ+-\/>.* arrows
Next: _
=====A=====B=====
1(Text test )          2.3
2                                           4.3
3
  
```

9731
Auto.

Type NFTR<R> to see the effect of changing the format to right


```

A1      Text test
T1      *          #,C,F,G,H,I,T,P          9731
Next:   New Format to be _                  Auto.
        =====A=====B=====
1(Text test          )                      2.3
2                                                  4.3
3

```

```

A1      Text test
T1      *
Next:   New Format to be Text right justified_
        -----A-----B-----
1(Text test           )                2.3
2                                           4.3
3

```

```

-----
A1      Text test
Tr *    ABCDEFGI JLMNPQRSUVWXZ+-\|/>>.* arrows
Next: _
-----A-----B-----
1(      Text test)      2.3
2
3      4.3

```

THE CRACKER TUTORIAL II

MORE ABOUT ENTERING TEXT

```

A1      Text test
Tr      #, <R>
Next:   New width of column 4_
=====A=====B=====
1(      Text test)          2.3
2                                     4.3
3

```

```

Al      Text test
Tr *    ABCDEFGIJKLMNOPQRSTUVWXYZ+~\|>.* arrows
Next:
--A= =====B=====
1(Text)                                2.3
2                                         4.3
3

```

If you type NFH<R> you will change the cell format to heading:

[illegible]


```
A1  Text test
H  *  ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
      ==A= -----B=====
1(Text test)          2.3
2                      4.3
3
```

9731
Auto.

The whole of the 'Text test' is again displayed and part of it overwrites the start of cell B2.

While entering text you can backspace using the DEL key or press ESC to abandon the whole process. This does not destroy what you had before you started the entry.

EDITING

If you decide to change an entry you can either re-enter it or use the built in cell editor. This editor allows you to add, delete or change characters in an entry. The editor may be used for both text and expression entries.

As an example prepare a line of characters that can be edited. In this case you have just a line of letters between A and P. Type 1C20<R>T<R> then key '.' followed by 'ABCDEFGHIJKLMNPO'.

```
A1
Tr  * Enter characters
    >> ABCDEFGHIJKLMNPO_
      =====A=====
1(                                     )
```

7968
Auto.

```
A1  ABCDEFGHIJKLMNPO
Tr  *  ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
      =====A=====
1(      ABCDEFGHIJKLMNPO)
```

7944
Auto.

Now type E<R> to get into the edit mode:

```
A1  ABCDEFGHIJKLMNOP
Tr  *   <R>
Next: Edit_
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

```
A1  ABCDEFGHIJKLMNOP
Tr  *  I,X,Z,space,<DEL>,<R>
Edit: ABCDEFGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

As you can see the list of options is then displayed. It reads Insert, eXchange, Zap (delete), space to move the cue right, to move left, and <R> to leave the edit mode. Press the space bar 5 times and then type I for insert:

```
A1  ABCDEFGHIJKLMNOP
Tr  *   "Insert" char,<DEL>,<R>
Edit: ABCDE_FGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

A space is opened up for you to insert your new character. Type " and you will see it inserted and a new space opened up for you.

```
A1  ABCDEFGHIJKLMNOP
Tr  *   "Insert" char,<DEL>,<R>
Edit: ABCDE_FGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Type <R> to leave the insert mode and this returns you to the main edit prompt:

```
A1  ABCDEFGHIJKLMNOP
Tr  *   I,X,Z,space,<DEL>,<R>
Edit: ABCDE_FGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Press the key to go back one space:

```
A1  ABCDEFGHIJKLMNOP
Tr  *   I,X,Z,space,<DEL>,<R>
Edit: ABCDE_FGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Press X and you enter eXchange mode:

```
A1  ABCDEFGHIJKLMNOP
Tr  *    "Xchange" char,<R>
Edit: ABCDE_FGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Type / and you will see it replace the -.

```
A1  ABCDEFGHIJKLMNOP
Tr  *    "Xchange" char,<R>
Edit: ABCDE/FGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Now type and the cue will back one space again:

```
A1  ABCDEFGHIJKLMNOP
Tr  *    "Xchange" char,<R>
Edit: ABCDE/FGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

You can now use the <R> key to leave the eXchange mode and return to main edit prompt:

```
A1  ABCDEFGHIJKLMNOP
Tr  *    l,X,Z,space,<DEL>,<R>
Edit: ABCDE/FGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

The other option you have while editing is character deletion.
Type Z and you enter zap mode:

```
A1  ABCDEFGHIJKLMNOP
Tr  *    "zap" space,<R>
Edit: ABCDE/FGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

To delete just press the space bar once for each deletion. The / will immediately disappear.

```
A1  ABCDEFGHIJKLMNOP
Tr  *    "zap" space,<R>
Edit: ABCDEFGHIJKLMNOP
      -----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

You leave the zap mode by pressing <R> and you return to the main edit prompt:

```
A1  ABCDEFGHIJKLMNOP
Tr  *  I,X,Z,space,<DEL>,<R>
Edit: ABCDEFGHIJKLMNOP
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

Now to leave the editor type <R> for a second time:

```
A1  ABCDEFGHIJKLMNOP
Tr  *  ABCDEFGIJKLMNOPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----
1(  ABCDEFGHIJKLMNOP)
```

7944
Auto.

At any stage up to this point you could have pressed the ESC key and retrieved the entry in the form it was before you entered the edit mode.

MORE ABOUT EXPRESSIONS

Expressions are at the heart of THE CRACKER for with them you can do business calculations as well as scientific and engineering mathematics. An expression is any entry that can be used to calculate a value. This includes a simple number.

What are the features of an expression? You may think of it as a mathematical formula. 2+2 is an expression which should equal 4. 2+A1 is also an expression which as you have already seen is equal to '2 plus whatever is the value of A1'. You can use any of the mathematical operators:

+	plus
-	minus
*	multiply
/	divide
^	to the power of
%	percent

The + and - you will of course be familiar with but if you are new to computing the symbols *, / and ^ may be unfamiliar. You always need to put in the ^ symbol when you want get 'to the power of' because superscripts are not available. The % symbol is not often used in calculations outside of spreadsheets. It means percent such that 5%20 is 5 percent of 20.

You will now see how to enter a formula into a cell. You may have realised by now that there is the problem of how to enter the formula on one line when usually they take up two or more. This is simply done by enclosing the parts in () and putting / between them, so:

$\frac{2+A1}{5+A2}$ becomes (2+A1)/(5+A2)

If necessary you can have several layers of () to avoid ambiguities. If THE CRACKER finds a reference does not have a value it will use 0.

THE CRACKER works out expressions in normal algebra. This is the algebra you use in hand calculations. The back to front entry method (reverse polish notation) is not used. There is an order of precedence of the operators in calculation. In general make up formulae as you would for pencil and paper, convert them to a one line form and enter them.

On many spreadsheets $1 + 2/5$ is worked out to be 0.6 which is not the correct result of 1.4. On those sheets you have to take positive action to force a correct result by entering $1 + (2/5)$. They often get unexpected results, but you have no such problems. The order of precedence of calculation is:

%
^
* and /
+ and -
>, < and = these are logical operators, more of them later.
], [and !

THE CRACKER starts with the innermost brackets. Within the brackets it works out the part of the expression starting with the highest precedence operator. Where there are two operators of equal precedence the one on the left is done first. The next bracket is then done and so on. Do not worry about the details just make up your expressions in your usual way.

The other item you can use in an expression is a function. This is a device for getting a value of a built in formula. The simplest function is PI which when used in calculations gives the value 3.141592653589793 without you having to enter it.

```
A1
G * Enter number or expression
>> 2*PI_
=====A=====
1(
2
3
```

9745
Auto.

```
A1 (2*PI)
G * ABCDEFGIJKLMNOPRSUVWXZ+-\/>.* arrows
Next: _
=====A=====
1( 6.28318530717959)
2
3
```

9723
Auto.

The next type of function is the one in which you put a value such as SIN(30) which means the Sine of 30 degrees. Note the value is enclosed in brackets:

```
A1
G * Enter number or expression
>> SIN(30)_
=====A=====
1( )
2
3
```

9745
Auto.

```
A1 (SIN(30))
G * ABCDEFGIJKLMNOPRSUVWXZ+-\/>.* arrows
Next: _
=====A=====
1( 0.5)
2
3
```

9720
Auto.

One feature you will find especially useful is that you could have put $2*15$ or $4*A1$ or any other expression in the brackets part. This bit is calculated first and then fed to the function. You can even have functions of functions such as $SIN(SIN(30))$ and so on. There are no real restrictions except the number of levels of brackets.


```

A1
G * Enter number or expression          9745
>> SIN(SIN(30))_                        Auto.
=====A=====
1(                                     )
2
3

```

```

A1 (SIN(SIN(30)))
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 9715
Next: _                               Auto.
=====A=====
1( 0.00872653549837393)
2
3

```

The last type of function is the one which has two or more values enclosed in (). These enclosed values are called arguments. SUM(2,3) adds all the values separated by commas in the (). This function has a value of 2+3 or 5. Try these functions for yourself:

```

A1
G * Enter number or expression          9745
>> SUM(2,3)_                            Auto.
=====A=====
1(                                     )
2
3

```

```

A1 (SUM(2,3))
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 9719
Next: _                               Auto.
=====A=====
1(                                     5)
2
3

```

You can have as many parts to the inside as you like. Here is an example with three arguments:

```

A1
G * Enter number or expression          9745
>> SUM(2,3,4)_                          Auto.
=====A=====
1(                                     )
2
3

```

```

A1 (SUM(2,3,4))
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 9717
Next: _                               Auto.
=====A=====
1(                                     9)
2
3

```

The values could also be expressions:


```

A1
G * Enter number or expression
>> SUM(2+3,4,5)_
=====A=====
1(
2
3

```

9745
Auto.

```

A1 (SUM(2+3,4,5))
G * ABCDEFGHILMNOPQRSUVWXYZ+ - / > . * arrows
Next: _
=====A=====
1(
2 15)
3

```

9715
Auto.

These then are what make expressions. There are just two more facilities you might find useful. The first is the # symbol which when used after a coordinate puts its value into the worksheet. A1# will actually put the value of A1 into the expression. If you type 2+A1# and A1 is 3 you will find your expression become 2+3.

```

A2
G * Enter number or expression
>> 2+A1#
=====A=====
1(
2 3)
3

```

9726
Auto.

Instantly becomes:

```

A2
G *
>> 2+3_
=====A=====
1(
2 3)
3

```

9726
Auto.

The other facility is the ! symbol which forces the expression you are entering to be replaced by its value up to that point. So if you type 2+3! the display will immediately convert to 5 and you can continue the entry from that point:

```

A1
G * Enter number or expression
>> 2+3!
=====A=====
1(
2 5)
3

```

9745
Auto.

Instantly becomes:

```

A1
G *
>> 5_
=====A=====
1(
2 5)
3

```

9745
Auto.

CHANGING THE LAYOUT

The layout of the cells may be changed using the various format commands. So far you have only used the G for 'General' format in expressions. You may even have been wondering what is meant by format. If you have a number such as 2 there are several ways you could write it down such as 2, 2.0, 2.000, 0.2E1. All are perfectly valid but not all are as you would want for presentation in a report. If it was referring to a number of items you would want 2. If it was the amount of money you would want 2.00. If it was a laboratory test result you may want to infer a precision to the nearest 0.1 by using 2.0.

You can get these layouts using the various formats available with THE CRACKER. The full list of format types is:

- ```
G General, rather like a scientific calculator.
#F Finance, balance sheet format (# is no. places shifted)
#D Decimal places specified (# is no. places).
#E Exponent scientific notation (# is no. sig. figs.).
I Integer, nearest whole number.
P Plot format, horizontal bar graph.
```

You can now experiment with changing the format. Take a cell and type FG<R>:

```

A1
G * <R>
Next: Format to be General_
 =====A=====
1(
2
3

```

**Type 2<R>:**

```

A1
G * Enter number or expression
>> 2...
=====A=====
1()
2
3

```

9745  
Auto.

```

A1 (2)
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+-\|>.* arrows
Next: _
 =====A=====
 1(2)
 2
 3

```

9726  
Auto.

Now you can change the format of an existing entry. Type  
NF4D<R>:

```

| A1 (2)
| G *
| Next: New Format to be 4 places Decimal_
| =====A=====
| 1(
| 2
| 3

```

9726  
Auto.



```

A1 (2)
4D * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: _
-----A-----
1(2.0000)
2
3

```

9726  
Auto.

Type NFOF<R>:

```

A1 (2)
4D * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: New Format to be 0 places shifted Finance
-----A-----
1(2.0000)
2
3

```

9726  
Auto.

```

A1 (2)
OF * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: _
-----A-----
1(2.00)
2
3

```

9726  
Auto.

This format will put commas into the numbers as they get larger.  
Negatives will be enclosed in brackets. Type NFI<R>:

```

A1 (2)
F * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: New Format to be Integer_
-----A-----
1(2.00)
2
3

```

9726  
Auto.

```

A1 (2)
I * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: _
-----A-----
1(2)
2
3

```

9726  
Auto.

Type NF6E<R>:

```

A1 (2)
I * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: New Format to be 6 sig. figs. Exponent_
-----A-----
1(2)
2
3

```

9726  
Auto.

```

A1 (2)
6E * ABCDEFGIJKLMNPQRSUVWXZ+~\|/>.* arrows
Next: _
-----A-----
1(0.200000E+01)
2
3

```

9726  
Auto.



However you display a value there is no change in the way the number is stored by THE CRACKER. Formatting does not change the value just how it is laid out.

---

CHANGING COLUMN WIDTHS AND DEFAULT VALUES

You have already used the New command to change the column width and to change the format. It can also be used to change the default format. This is the format that will be used if you do not declare a particular one before each entry. The complete set of New commands is:

|     |                                          |
|-----|------------------------------------------|
| NW# | New column width (# is no. spaces width) |
| NF  | New format to be ...                     |
| ND  | New default format to be ...             |



INSERTING AND DELETING (ZAP)

The insert and delete (Zap) commands act on rows or columns either one at a time or in groups. Type ZA<R>Y<R> to clear the screen with confirmation:

```
A1 (2)
6E *
Next: Zap (delete) all <R>
-----A-----
1(0.200000E+01)
2
3
```

9726  
Auto.

```
A1 (2)
6E *
Next: Zap (delete) all - confirm Y,N
-----A-----
1(0.200000E+01)
2
3
```

9726  
Auto.

Type IC10<R>:

```
*
Next: Insert column , width 10_
```

9745  
Auto.

Each column will have its own default format so type G<R>:

```
*
Next: Default Format to be General_ <R>
```

9745  
Auto.

```
A1
G * ABCDEFGIJKLMNPQRSUVWXYZ+-|/>.* arrows
Next: _
-----A-----
1()
```

9745  
Auto.

Next try entering some extra lines with I5L<R>:

```
A1
G
Next: Insert 5 lines_ <R>
-----A-----
1
```

9745  
Auto.

If you type <R> for destination they will be put in front of the cursor. Your alternative at this stage is E<R> for 'End' in which case it will be put after the cursor.



```
A1
G * U,D,L,R,arrows,J,<R>,E
Next: Destination _
 -----A-----
 1()
```

9745  
Auto.

```
A1
G * ABCDEFGIJKLMNOPQRSUVWXYZ+~\/>.* arrows
Next: _
 -----A-----
 1()
 2
 3
 4
 5
 6
```

9745  
Auto.

By inserting columns and rows in this way you can build the worksheet to up to 52 columns and 255 lines. The limitation on size will be useable memory. THE CRACKER is designed to use absolutely the minimum memory and so only those cells which are occupied use any memory at all. You can therefore start with as big a sheet as you like and fill in the detail later. If you have used other spreadsheets you will appreciate the difference.

You can use the Zap command to reduce the sheet by column, line or as a whole with the 'All' option. You will get more information about available options in the next section.

# USING SECTIONS OF THE WORKSHEET

You know about columns, lines and cell entries. THE CRACKER will also manipulate blocks of the worksheet. Consider a block as a rectangular area of the sheet which you call by giving the coordinate of the top left and then the bottom right corners. You will find that this feature is very useful and makes THE CRACKER especially powerful. You will see how it will allow you to collect your information together and then rearrange it in any way you like. The full list of the parts of the worksheet is:

|        |                                    |
|--------|------------------------------------|
| Entry  | a single cell.                     |
| Block  | a rectangle from within the sheet. |
| Column | a vertical band.                   |
| Line   | a horizontal row.                  |
| All    | the whole of the sheet.            |

Only the Copy and Blank commands uses all five options. You will always know which are available from the prompt list. To give you a feel for the parts of the worksheet here are a few examples for Copy: Type I3C15<R>TR<R> then I3L<R><R> followed by '.' and 'abcdefghijklmno'.

```
A1 abcdefghijklmno
Tr * <R>
Next: Copy entry _
 -----A-----B-----C-----
 1(abcdefghijklmno)
 2
 3
 4
```

9722  
Auto.



```
A1 abcdefghijklmno
Tr * U,D,L,R,arrows,J,<R>,F,P,M
Next: Destination _
-----A-----B-----C-----
1(abcdefghijklmno)
2
3
4
```

9722  
Auto.

Move the cursor right:

```
B1 abcdefghijklmno
Tr * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----B-----C-----
1 abcdefghijklmno(abcdefghijklmno)
2
3
4
```

9699  
Auto.

```
B1 abcdefghijklmno
Tr * #,<R>
Next: Copy line _
-----A-----B-----C-----
1 abcdefghijklmno(abcdefghijklmno)
2
3
4
```

9699  
Auto.

```
B1
Tr * U,D,L,R,arrows,J,<R>,F,P,M
Next: Destination _
-----A-----B-----C-----
1 abcdefghijklmno(abcdefghijklmno)
2
3
4
```

9699  
Auto.

Move the cursor down and left:

```
A2 abcdefghijklmno
Tr * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno
2(abcdefghijklmno)abcdefghijklmno
3
4
```

9653  
Auto.

```
B2 abcdefghijklmno
Tr * #,<R>
Next: Copy column _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno(abcdefghijklmno)
3
4
```

9653  
Auto.

```
B2 abcdefghijklmno
Tr * U,D,L,R,arrows,J,<R>,F,P,M
Next: Destination _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno(abcdefghijklmno)
3
4
```

9653  
Auto.

Move the cursor right twice.



```
C1 abcdefghijklmno
Tr * ABCDEFGI JLMNPQRSUVWXZ+-\|/>.* arrows
Next: _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno(abcdefghijklmno)
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3
4
```

9607  
Auto.

```
A1 abcdefghijklmno
Tr * crd...crd <R>
Next: Copy block A1...B2_
-----A-----B-----C-----
1(abcdefghijklmno)abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3
4
```

9607  
Auto.

```
A1 abcdefghijklmno
Tr * U,D,L,R,arrows,J,<R>,F,P,M
Next: Destination_
-----A-----B-----C-----
1(abcdefghijklmno)abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3
4
```

9607  
Auto.

Move the cursor down and across:

```
B3 abcdefghijklmno
Tr * ABCDEFGI JLMNPQRSUVWXZ+-\|/>.* arrows
Next: _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3 (abcdefghijklmno)abcdefghijklmno
4 abcdefghijklmno abcdefghijklmno
```

9515  
Auto.

```
B3 abcdefghijklmno
Tr * <R>
Next: Copy all _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3 (abcdefghijklmno)abcdefghijklmno
4 abcdefghijklmno abcdefghijklmno
```

9607  
Auto.

```
B3 abcdefghijklmno
Tr * F,P,M
Next: Destination _
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3 (abcdefghijklmno)abcdefghijklmno
4 abcdefghijklmno abcdefghijklmno
```

9607  
Auto.

```
B3 abcdefghijklmno
Tr *
Next: Destination Printer_
-----A-----B-----C-----
1 abcdefghijklmno abcdefghijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3 (abcdefghijklmno)abcdefghijklmno
4 abcdefghijklmno abcdefghijklmno
```

9607  
Auto.

All the entries will be copied to the printer. The command area and axes will be excluded.



BLANKING

You can use the Blank command to clear all or parts of the worksheet. You will not change its size or default settings with this command. Individual set formats will revert to the default ones. You may use all five of the options from the previous section.

As an example you will see how to use the block option in more detail. Type BBA1.B2<R>. Note that when you type the single period THE CRACKER expands this to three for clarity.

```
A1 abcdefghijklmno
Tr * crd...crd <R>
Next: Blank block A1...B2_
=====A=====B=====C=====
1(abcdefgijklmno)abcdefgijklmno abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno abcdefghijklmno
3 abcdefghijklmno abcdefghijklmno
4
```

9561  
Auto.

```
A1
Tr * ABCDEFGIJKLMNPQRSUVWXZ+-\/>!* arrows
Next: _
=====A=====B=====C=====
1() abcdefghijklmno
2 abcdefghijklmno
3 abcdefghijklmno abcdefghijklmno
4
```

9653  
Auto.

Now remove the rest of the worksheet with BE<R> on B3 and B4 and with BC<R> on column C.

You will never destroy the sense of the sheet with the Blank command. If you have an entry like 2\*B3 and try to blank B3 THE CRACKER will check and warn you.



USING THE MOVE COMMAND

You use the Move command to transfer parts of the sheet to another part of the work area. After the previous section you just have a blank worksheet. Fill up the four top left entries again using the '.' command. Now type MBA1.B2<R>:

```
A1 abcdefghijklmno
Tr * crd...crd <R>
Next: Move block A1...B2_
-----A-----B-----C-----
1(abcdefghijklmno)abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno
3
4
```

9647  
Auto.

Type JB3<R>:

```
A1 abcdefghijklmno
Tr * U,D,L,R,arrows,J,<R>
Next: Destination _
-----A-----B-----C-----
1(abcdefghijklmno)abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno
3
4
```

9647  
Auto.

```
A1 abcdefghijklmno
Tr * #,<R>
Next: Jump to B3_
-----A-----B-----C-----
1(abcdefghijklmno)abcdefghijklmno
2 abcdefghijklmno abcdefghijklmno
3
4
```

9647  
Auto.

```
B4 abcdefghijklmno
Tr * ABCDEFGIJKLMNPQRSUVWXYZ+-\/>.* arrows
Next: _
-----A-----B-----C-----
1()
2
3 abcdefghijklmno abcdefghijklmno
4 abcdefghijklmno abcdefghijklmno
```

9647  
Auto.

Try the Move command with some of the other options. The command will overwrite the destination area. As there is a danger of you destroying the sense of any calculations on the sheet THE CRACKER will make a check and prevent you so doing.

If there are any coordinate references in expressions being moved you will find these adjusted to the new locations.



A FURTHER LOOK AT THE COPY COMMAND

You have looked at many of the options of the Copy command while investigating the parts of the worksheet. As you have seen any part of the worksheet can be copied to any other part or to the printer. In some ways the Copy command is the same as the Move command only the contents of the original location are not deleted. There is however a major difference if you have coordinates in your expressions being copied. You will be asked whether you want the references adjusted. If you answer yes and it is a row copy every reference to other locations on that row will be changed for the new location.

You can try an example of an adjusted copy to get a feel for what is involved. Type ZA<R>Y<R> to clear the sheet and then I2C15<R>G<R> to put in two general format columns followed by IL<R><R> to add an extra line.

Type .2<R> to put 2 into location A1, move the cursor to B2 with R, then enter 2\*A1<R>:

```

B1
G * Enter number or expression
>> 2*A1_
-----A-----B-----
1
2

```

9720  
Auto.

Then type CL<R> for copy line:

```

B1 (2*A1)
G * #,<R>
Next: Copy line _
-----A-----B-----
1
2

```

(1)  
9698  
Auto.

Followed by D to move the cursor to the destination line:

```

B1 (2*A1)
G * U,D,L,R,arrows,J,<R>,F,P,M
Next: Destination _
-----A-----B-----
1
2

```

(1)  
9698  
Auto.

You will then be queried about adjusting references so type Y<R>:

```

B2
G * <R>
Next: Adjust references - Yes_
-----A-----B-----
1
2

```

9698  
Auto.

```

B2 (2*A2)
G * ABCDEFGIJKLMNOPQRSUVWXYZ+~\/>.* arrows
Next: _
-----A-----B-----
1
2

```

(1)  
9657  
Auto.



You can see that what was 2\*A1 in B1 has become 2\*A2 in B2. If you think about it the adjustment option has kept the meaning of the first line, two times the value on the left, in the second line. You will find it is very much the same for columns.

As well as copying within the worksheet you can copy to the printer or to disk files. Similarly you can copy from disk files.

Copying to the printer is the same as copying within the worksheet but for the destination give P for printer. For disk files give the destination as F for file. You will then be asked for a CP/M filename. You may not be fully familiar with the filename convention so here is a summary.

The filename may be up to 8 characters long and may include any printable character except !\*?=/.,: or space. It should be in upper case. The filename is followed by a '.' and then an extension which defines the type of file. There are only 3 extensions that THE CRACKER recognises .MEM, .TXT and .DAT. By giving the relevant extension you will inform THE CRACKER of the type of file you want to read or write.

Now for more information on types of file. If you want to write out all or a part of your worksheet to a file that can be used by an editor or word processing program give it a .TXT for text extension. This is very useful if you want to incorporate the results calculated by THE CRACKER into a full written report.

If you wish to save all or part of the worksheet in such a way that THE CRACKER can read it and display it in the original form then use the .MEM for memory extension.

The .DAT extension is used when you wish to read a file of data numbers into your sheet. The file should be in character form such as you would get from an editor or word processor, or from formatted output from FORTRAN or BASIC. The file will be read in much the same way as prepared but be careful that if more than one column is being read that zero readings are shown by a 0 and not just by blanks. It will do no harm but if this was a blank in the first column the then second column will be read as if it was the first.

If you use the editor to make a file such as EXAMPLE.DAT:

2.34, 23.7  
456.73, 84  
27, 3  
63.3, 1234.5

Then you load THE CRACKER and create a worksheet of sufficient size to hold the entries. Type I2C15<R>G<R> then I3L<R><R> to do

this:

```

A1 FILENAME.EXT <R> 9733
G Next: Copy filename EXAMPLE.DAT_ Auto.
-----A-----B-----
1(
2
3
4

```

```

A1 U,D,L,R,arrows,J,<R> 9733
G Next: Destination _ Auto.
-----A-----B-----
1(
2
3
4

```

Type <R> to indicate you want A1 to be the start:

```

A1 (2.34) 9560
G ABCDEFGIJKLMNOPQRSTUVWXYZ+~\>.* arrows Auto.
Next: _
-----A-----B-----
1(2.34) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5

```

The entries will be in the default format of the columns. If this is Text then the General format will be used. Finance format entries will be converted to the G format. As with the .TXT extension only the body of the worksheet will be in the file



not the control area column letters or line numbers.

If you want to output a file of numbers you can also use the .DAT extension. When you do this any text will be ignored and only the numerical values will be sent to the file. To give you a demonstration change A1 to a text entry by typing FTL<R> followed by Text<R>.

```
A1 (2.34)
G *
Next: Format to be Text left justified_
-----A-----B-----
1(2.34) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

```
A1 Text
T1 * ABCDEFGHIJKLMNOPQRSTUVWXYZ+~|/>.* arrows
Next: _
-----A-----B-----
1(Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

Now type CA<R>F and EXAMPLE2.DAT<R>:

```
A1 Text
T1 * <R>
Next: Copy all _
-----A-----B-----
1(Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

```
A1 Text
T1 * FILENAME.EXT <R>
Next: Destination Filename EXAMPLE2.DAT_
-----A-----B-----
1(Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

If you now leave THE CRACKER by typing Q<R>:

```
A1 Text
T1 * <R>
Next: Quit
-----A-----B-----
1 Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

```
Worksheet filed to SECURITY.MEM
* Bye *
```

Note your work is automatically saved in case you later regret leaving at that point.

You can use the system command TYPE EXAMPLE2.DAT<R> and you will get:

```
0.0, 23.7
456.73, 84
27, 3
63.3, 1234.5
```

As you can see the Text entry was not included.



To see how the .TXT extension works you will first need to load THE CRACKER and get back your previous sheet. Type CRACKER to load it then type CF and SECURITY.MEM<R>:

```
* FILENAME.EXT <R>
Next: Copy filename SECURITY.MEM_ 9707
 Auto.
```

Type CA<R>F and EXAMPLE3.TXT:

```
A1 Text
T1 * <R>
Next: Copy all _ 9544
-----A-----B-----
1(Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

```
A1 Text
T1 * FILENAME.EXT <R>
Next: Destination Filename EXAMPLE3.TXT_ 9544
-----A-----B-----
1(Text) 23.7
2 456.73 84
3 27 3
4 63.3 1234.5
```

Now quit THE CRACKER again as you did above Q<R> and enter TYPE EXAMPLE3.TXT<R>:

```
Text 23.7
 84
 3
456.73
27
63.3 1234.5
```

This time all your entries including the text have been displayed.



# THE ORDER OF CALCULATION

A unique feature of THE CRACKER is that you will find it does not matter where in the spreadsheet you enter your data or expressions. If you have used other spreadsheets before you will appreciate the difficulties of getting all your calculations in the right order and avoiding circular references.

As you enter an expression THE CRACKER gives it an 'order of calculation number' which is displayed at the top right of the screen. This is not affected by where you enter the expression just by the references in the formula. If you enter 2\*B1 into A3, for example, then you know that you want the 2\*B1 to be calculated after B1 otherwise it does not make sense. Well THE CRACKER goes through all the expressions as you enter them working out which is the correct order for calculation. If the entry is a constant then it does not have an 'order of calculation number'.

This feature allows you to rearrange the sheet as you like without affecting your calculated values. This can not be done on other spreadsheets. The operation is entirely automatic so you do not have to worry about it.

You will see the calculation number at the top right hand corner of the screen.

## PARTITIONING THE SCREEN

As your worksheet gets larger you sometimes want to see what is going on in more than one position at a time. You can do this with the partition commands. You can split the screen horizontally or vertically or both. You also have the option of locking the windows so that as you move in one part the other will move in a synchronised way. It is easy to move around the screens with single key commands.

Type ZA<R>Y<R> to clear the screen and CF<R> ADLIST.MEM to read in this example.

| ----- |                                        |                  |          |       |
|-------|----------------------------------------|------------------|----------|-------|
| A1    | Init                                   |                  |          |       |
| T1    | * ABCDEFGI JLMNPQRSUVWXZ+-\/>.* arrows |                  |          | 7401  |
| Next: |                                        |                  |          | Auto. |
|       | ==A==                                  | ==B==            | ==C==    | ==D== |
|       | 1(Init)Name                            | No Road          | Town     | £     |
| 2     |                                        |                  |          |       |
| 3     | R.T. Cowan                             | 16 Jeremy Grove  | Hampton  | 17    |
| 4     | G.L. Blake                             | 37 Osmaston Road | Knowle   | 56    |
| 5     | A.J. Hunt                              | 25 Latimer Road  | Wythall  | 43    |
| 6     | H.A. Fisher                            | 32 Florence Road | Henley   | 28    |
| 7     | C.D. Beard                             | 86 Valley Road   | Redditch | 25    |

Here you are going to see the screen partitioned horizontally. As you type PH for partition horizontally you will see that a grid is put up on the screen. You use this to judge where you want the split to take place. Whatever number you choose that will be the first location of the second window.



```

A1 Init
T1 *
Next: Partition screen horizontally 4_
==A= ==B== =C= =====D===== =E==== =F=
1(Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston2Road Knowle 56
5 A.J. Hunt 25 Latimer 3oad Wythall 43
6 H.A. Fisher 32 Florence4Road Henley 28
7 C.D. Beard 86 Valley R5ad Redditch 25
6
7
8
9
10
1
2
3
4

```

```

A1 Init
T1 * ABCDEFGI JLMNPQRSUVWXZ+-\|/>.* arrows
Next: _
==A= ==B== =C= =====D===== =E==== =F=
1(Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56

1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

You now have two sets of row numbers starting from 1. The next command to investigate is the backslash \. This is used to jump between horizontal windows sections.

```

A1 Init
T1 * ABCDEFGI JLMNPQRSUVWXZ+-\|/>.* arrows
Next: _
==A= ==B== =C= =====D===== =E==== =F=
1(Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56

1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

The effect as you can see is to jump to the same cell but in the other window. Now let us see what happens when we move the cursor in the lower part and then transfer back to the upper part:



| Al                           | Init             | V,H |             | 7401        |
|------------------------------|------------------|-----|-------------|-------------|
| Tl                           | *                |     |             | Auto.       |
| Next: Partition synchronised |                  |     |             |             |
| --A=                         | ====B==          | =C= | =====D===== | =====E===== |
| 1 (Init)Name                 | No Road          |     | Town        | £           |
| 2                            |                  |     |             |             |
| 3 R.T. Cowan                 | 16 Jeremy Grove  |     | Hampton     | 17          |
| 4 G.L. Blake                 | 37 Osmaston Road |     | Knowle      | 56          |
| 1 Init Name                  | No Road          |     | Town        | £           |
| 2                            |                  |     |             |             |
| 3 R.T. Cowan                 | 16 Jeremy Grove  |     | Hampton     | 17          |
| 4 G.L. Blake                 | 37 Osmaston Road |     | Knowle      | 56          |
| 5 A.J. Hunt                  | 25 Latimer Road  |     | Wythall     | 43          |
| 6 H.A. Fisher                | 32 Florence Road |     | Henley      | 28          |
| 7 C.D. Beard                 | 86 Valley Road   |     | Redditch    | 25          |

To test it you can move the cursor down and then swap parts:

```

A3 R.T.
T1 * ABCDEFGIJKLMNOPQRSUVWXZ+~\|/>.* arrows
Next: D_
==A= ====B=== =C= =====D===== =E===== =F=
1 Init Name No Road Town £
2
3(R.T.)Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56

1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

Now the swap:



```

A7 C.D.
T1 * ABCDEFGIJKLMNPQRSUVWXYZ+~\|/.* arrows
Next: \
 ==A= ==B== =C= =====D===== =====E===== =F=
1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5
6
7(C.D.)Beard 86 Valley Road Redditch 25

```

Yes the lower cursor has also move down two places. Exactly the same operations can be used to divide the screen vertically:

```

A7 C.D.
T1 *
Next: Partition screen vertically 20_
 ==A= ----B--- -C= -----D----- -E----- -F=
1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56

1 Init Name No Road Town £
2 1 2 3 4
3 R.T. Cow12345678901234567890123456789012345678901
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7(C.D.)Beard 86 Valley Road Redditch 25

```

As before a grid will be superimposed:

| A7 C.D.       |          | T1 * ABCDEFGIJKLMNOPQRSUVWXYZ+~\>.* arrows |                | 7401  |
|---------------|----------|--------------------------------------------|----------------|-------|
| Next: --      |          |                                            |                | Auto. |
| --A=          | ====B=== | =C=                                        | ==D=-          |       |
| 1 Init Name   | No Road- | Init Name                                  | No Road        | -     |
| 2             | -        |                                            |                | -     |
| 3 R.T. Cowan  | 16 Jere- | R.T. Cowan                                 | 16 Jeremy Gro- |       |
| 4 G.L. Blake  | 37 Osma- | G.L. Blake                                 | 37 Osmaston R- |       |
| 1 Init Name   | No Road- | Init Name                                  | No Road        | -     |
| 2             | -        |                                            |                | -     |
| 3 R.T. Cowan  | 16 Jere- | R.T. Cowan                                 | 16 Jeremy Gro- |       |
| 4 G.L. Blake  | 37 Osma- | G.L. Blake                                 | 37 Osmaston R- |       |
| 5 A.J. Hunt   | 25 Lati- | A.J. Hunt                                  | 25 Latimer Ro- |       |
| 6 H.A. Fisher | 32 Flor- | H.A. Fisher                                | 32 Florence R- |       |
| 7(C.D.)Beard  | 86 Vall- | C.D. Beard                                 | 86 Valley Roa- |       |

There are now four parts to the screen. To move across the parts you use the forward slash /:

```

A7 C.D.
T1 * ABCDEFGI JLMNPQRSUVWXZ+-\|/>.* arrows
Next: /_
==A= ====B=== =C= =D== ==A= ====B=== =C= =====D=====
1 Init Name No Road- Init Name No Road -
2 -
3 R.T. Cowan 16 Jere- R.T. Cowan 16 Jeremy Gro-
4 G.L. Blake 37 Osma- G.L. Blake 37 Osmaston R-

1 Init Name No Road- Init Name No Road -
2 -
3 R.T. Cowan 16 Jere- R.T. Cowan 16 Jeremy Gro-
4 G.L. Blake 37 Osma- G.L. Blake 37 Osmaston R-
5 A.J. Hunt 25 Lati- A.J. Hunt 25 Latimer Ro-
6 H.A. Fisher 32 Flor- H.A. Fisher 32 Florence R-
7 C.D. Beard 86 Vall- (C.D.)Beard 86 Valley Roa-

```

Now track right so that column D is brought into view.



|       |                                           |             |               |             |                |
|-------|-------------------------------------------|-------------|---------------|-------------|----------------|
| D7    | Valley Road                               |             |               |             | 7401           |
| T1    | * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows |             |               |             | Auto.          |
| Next: | -----                                     |             |               |             |                |
|       | --A--                                     | -----B----- | -C- --D--     | -----D----- | -----E----- F- |
| 1     | Init Name                                 | No Road-    | Road          | Town        | -              |
| 2     |                                           |             |               |             | -              |
| 3     | R.T. Cowan                                | 16 Jere-    | Jeremy Grove  | Hampton     | -              |
| 4     | G.L. Blake                                | 37 Osma-    | Osmaston Road | Knowle      | -              |
| 1     | Init Name                                 | No Road-    | Road          | Town        | -              |
| 2     |                                           |             |               |             | -              |
| 3     | R.T. Cowan                                | 16 Jere-    | Jeremy Grove  | Hampton     | -              |
| 4     | G.L. Blake                                | 37 Osma-    | Osmaston Road | Knowle      | -              |
| 5     | A.J. Hunt                                 | 25 Lati-    | Latimer Road  | Wythall     | -              |
| 6     | H.A. Fisher                               | 32 Flor-    | Florence Road | Henley      | -              |
| 7     | C.D. Beard                                | 86 Vall-    | (Valley Road  | )Redditch   | -              |

You can synchronise the parts vertically and stop it at any time. To get rid of the partitioning just type PE for partition end. If you save the worksheet on file, it will be in the unpartitioned form.

# USING MACRO COMMAND GROUPS

Often you will want to go through the same set of commands repeatedly. A typical example is the changing of the format of all the cells in one column. This can be both time consuming and tedious. To get round this situation use the \* MACRO command. A macro is defined as a linked set of commands. To use the facility you have to put your set of commands into one of the first 9 cells of column A. If necessary you will have to create a new column A. You can later remove it.

Just enter the command letters as if you were actually using the commands. Where you would want to use a <R> carriage return use the q symbol.

To call the macro you only have to type \* followed by the line number in which the macro is stored. So \*1 calls the macro in cell A1. If you want your macro to loop and be carried out repeatedly then finish it with a reference to itself. For example with cell A1 this would mean finishing with \*1.

As an illustration the following example takes a column of numbers in general format and changes them all to '1 place decimal' format.

|       |                                           |             |
|-------|-------------------------------------------|-------------|
| B1    | (2.345)                                   | 7828        |
| G     | * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows | Auto.       |
| Next: | -----                                     |             |
|       | -----A-----                               | -----B----- |
| 1     | (                                         | 2.345)      |
| 2     |                                           | 34.56       |
| 3     |                                           | 76.54       |
| 4     |                                           | 123.4       |
| 5     |                                           | 2345.6      |
| 6     |                                           | 7.2345      |



Start by moving the cursor to cell A1 where the macro is to be entered. Type '.' to get into entry mode and the type NF1D@D\*1 which says 'new format one place decimal, carriage return, down and finally do macro 1 again'.

|    |                    |   |        |
|----|--------------------|---|--------|
| A1 |                    |   |        |
| T1 | * Enter characters |   | 7828   |
|    | >> NF1D@D*1_       |   | Auto.  |
|    | -----A-----B-----  |   |        |
| 1  | (                  | ) | 2.345  |
| 2  |                    |   | 34.56  |
| 3  |                    |   | 76.54  |
| 4  |                    |   | 123.4  |
| 5  |                    |   | 2345.6 |
| 6  |                    |   | 7.2345 |

The @ symbol stands for the <R> carriage return.

|    |                                        |   |        |
|----|----------------------------------------|---|--------|
| A1 | NF1D@D*1                               |   |        |
| T1 | * ABCDEFGIJKLMNPQRSUVWXZ+~\/>.* arrows |   | 7812   |
|    | Next: _                                |   | Auto.  |
|    | -----A-----B-----                      |   |        |
| 1  | (NF1D@D*1                              | ) | 2.345  |
| 2  |                                        |   | 34.56  |
| 3  |                                        |   | 76.54  |
| 4  |                                        |   | 123.4  |
| 5  |                                        |   | 2345.6 |
| 6  |                                        |   | 7.2345 |

Next move the cursor to the location you want the macro to start its operations. In this case it is cell B1, and then type \*1 to get the macro going.

|    |                   |   |        |
|----|-------------------|---|--------|
| B1 | (2.345)           |   |        |
| G  | *                 |   | 7812   |
|    | Next: *1          |   | Auto.  |
|    | -----A-----B----- |   |        |
| 1  | NF1D@D*1          | ( | 2.345) |
| 2  |                   |   | 34.56  |
| 3  |                   |   | 76.54  |
| 4  |                   |   | 123.4  |
| 5  |                   |   | 2345.6 |
| 6  |                   |   | 7.2345 |

The commands will then work their way down the column.

|    |                                          |   |         |
|----|------------------------------------------|---|---------|
| B5 | (2345.6)                                 |   |         |
| G  | *                                        |   | 7812    |
|    | Next: New Format to be 1 places Decimal_ |   | Auto.   |
|    | -----A-----B-----                        |   |         |
| 1  | NF1D@D*1                                 |   | 2.3     |
| 2  |                                          |   | 34.6    |
| 3  |                                          |   | 76.5    |
| 4  |                                          |   | 123.4   |
| 5  |                                          | ( | 2345.6) |
| 6  |                                          |   | 7.2345  |

In the screen above the macro has got as far as cell B5.



```

86 (7.2345)
1D * outside worksheet
Next: D_
=====A=====B=====
1 NF1D@D*1 2.3
2 34.6
3 76.5
4 123.4
5 2345.6
6 (7.2)

```

7812  
Auto.

The macro is brought to a stop by any error message. In the screen above you can see that the D command for down can not be carried out because that is the end of the worksheet. Just type ESC and your looping macro operations will be completed and you will be in a position to go onto your next command.

```

86 (7.2345)
1D * ABCDEFGIJKLMNOPQRSTUVWXYZ+~|/>.* arrows
Next: _
=====A=====B=====
1 NF1D@D*1 2.3
2 34.6
3 76.5
4 123.4
5 2345.6
6 (7.2)

```

7812  
Auto.

## THE CRACKER TUTORIAL III USE OF FUNCTIONS

### MORE ON FUNCTIONS

You can see the full list of functions in the EXPRESSION ENTRY section of the command reference further on in the manual.

If you want to enter a function such as SUM(A1,A2,A3,A4) to add the cells from A1 to A4 you can use the shorthand SUM(A1...A4). A1...A4 is a range meaning use all the values between the first coordinate and the second. You only have to enter the first period and THE CRACKER will add the other two for clarity; you type A1.A4. You can even use the function in the form SUM(A1...A4,B7). Use a range anywhere you would otherwise put a list of adjacent entries. For most functions the range will still work if one or more of the entries is a blank. This is because the functions only work on the non blank cells.



THE IF,THEN,ELSE FUNCTIONS

There is a special function group which is known as the conditional. This is of the form IF(-),THEN(-),ELSE(-). The - stands for an expression. The first expression must be logical, which just means it must have an answer of TRUE or FALSE. An example of a logical expression is IF(B3=4) which has a value of TRUE if B3 does equal 4 or FALSE if it does not.

If indeed the IF(-) is TRUE the THEN(-) becomes operative and the cell takes the value given by the expression after the THEN. When the IF(-) is FALSE the ELSE(-) is used. With these functions you can build decision making into your calculations.

The full list of special operators you can use to give you an answer of TRUE or FALSE are:

```
= equal
! not equal
> greater than
] greater than or equal
< less than
[less than or equal
```

You can also use the functions TRUE or FALSE instead of an expression. They do not have arguments. If an expression is TRUE it is given a value of -1 and if it is FALSE it is set to 0. If you put a logical expression into a cell these will be the displayed values.

Alternatively you can set another cell, say B3 to TRUE or FALSE and then use the conditional in the form IF(B3),THEN(-),ELSE(-). In place of the expressions after THEN and ELSE you can use the special function ERROR. If this is encountered during a calculation then the calculation is stopped and a message is put up on the prompt line. You can treat this as if it is a normal error message. No harm is done using this function.

You can now try an example which includes some of these functions

and features. Clear the worksheet and type !C15<R>G<R> and !L<R><R> to set up your work area. Now type .TRUE<R> into A1:

```
A1
G * Enter number or expression 9493
>> TRUE_ Auto.
-----A-----
1()
2
```

Note that A1 takes on the value -1. Type D to move to A2 and type .IF(A1),THEN(5),ELSE(ERROR)<R> which means if A1 is true then give A2 the value 5 otherwise indicate an error:

```
A2
G * Enter number or expression 9471
>> IF(A1),THEN(5),ELSE(ERROR)_ Auto.
-----A-----
1 -1
2()
```

```
A2 (IF(A1),THEN(5),ELSE(ERROR)) (1)
G * ABCDEFGI JLMNPQRSUVWXZ+-\/>.* arrows 9427
Next: _ Auto.
-----A-----
1 -1
2(5)
```

As A1 was TRUE then A2 has become 5. You can now change A1 to see the effect on A2. Type U.FALSE<R>:



```

A1 (TRUE)
G * Enter number or expression
 >> FALSE_
 -----A-----
 1(-1)
 2 5

```

9427  
Auto.

```

A1 (TRUE)
G * ERROR called from (A2)_
 >> FALSE
 -----A-----
 1(0)
 2

```

9426  
Auto.

A1 has taken the value 0 for FALSE. The error message has come up saying where it was found. To get out of the situation you only have to press the ESC key. You find the current cell is now the one with the ERROR function in it so that you can do something about it.

TABLE HANDLING FUNCTIONS

Several functions extract values by analysing lists of values. You are probably not familiar with these functions and so you are going to have a more detailed description.

As a first example you are going to see the LOOKUP function. This function when given a value looks along a list to see where the value lies and then takes a value from the adjacent row or column. Confused? Well consider it as being the same as looking up a value in table where you look for your value in the first column to get your answer in the second. A typical example may be finding a commission percentage given sales income. These rates tend to jump from band to band.

```

B11 (LOOKUP(B10,B3...B8))
G * ABCDEFGIJKLMNOPQRSUVWXYZ+-\/>.* arrows
Next: _
 -----A----- B----- C-----
 1 Sales Commission
 2
 3 0 0
 4 1000 2.5
 5 2000 5
 6 4000 7.5
 7 10000 10
 8 20000 20
 9
 10 Sales achieved 15000
 11 Commission (%) (10)

```

In this case the salesperson brought in £15000 worth of business



and so he managed to get into the band between £10000 and £20000 for which he gets 10% commission. The form of the function is LOOKUP(value,list), the final value of the function is taken from the adjacent list. There is a similar function which you can use in the same way called INTERP which will interpolate a value from a list.

|                                             |                |             |             |
|---------------------------------------------|----------------|-------------|-------------|
| B11 (INTERP(B10,B3...B8))                   |                |             | (1)         |
| G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows |                |             | 8969        |
| Next: _                                     |                |             | Auto.       |
|                                             | -----A-----    | -----B----- | -----C----- |
| 1                                           |                | Sales       | Commission  |
| 2                                           |                |             |             |
| 3                                           |                | 0           | 0           |
| 4                                           |                | 1000        | 2.5         |
| 5                                           |                | 2000        | 5           |
| 6                                           |                | 4000        | 7.5         |
| 7                                           |                | 10000       | 10          |
| 8                                           |                | 20000       | 20          |
| 9                                           |                |             |             |
| 10                                          | Sales achieved | 15000       |             |
| 11                                          | Commission (%) | (           | 15)         |

Here the salesperson has been told that the commission will be calculated on a sliding scale based on the sales and commission table. As £15000 worth was sold this is midway between £10000 and £20000 and so he can expect a commission midway between 10% and 20%. The INTERP function does this calculation for you and in this case comes up with the answer 15%.

The CHOOSE function will look at a list and return the value given by the first argument. The form of this argument is CHOOSE(value,list). The value will be rounded to nearest whole number.

|                                             |             |       |
|---------------------------------------------|-------------|-------|
| A8 (CHOOSE(4,A1...A6))                      |             | (1)   |
| G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows |             | 9259  |
| Next: _                                     |             | Auto. |
|                                             | -----A----- |       |
| 1                                           |             | 0     |
| 2                                           |             | 1000  |
| 3                                           |             | 2000  |
| 4                                           |             | 4000  |
| 5                                           |             | 10000 |
| 6                                           |             | 20000 |
| 7                                           |             |       |
| 8                                           |             | 4000) |

The CHOOSE function at A8 has looked through A1...A6 to find the 4th value and returned the value 4000.

NPV stands for net present value and is a discounted cash flow function that calculates the effect of a discount rate on a set of cash flow figures. The form of the function is NPV(value,list) where value is the discount rate in percent and the list is a list of cash flows.

|                                             |                |             |             |             |             |             |       |
|---------------------------------------------|----------------|-------------|-------------|-------------|-------------|-------------|-------|
| B5 (NPV(B4,B2...F2))                        |                |             |             |             |             |             | (1)   |
| G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows |                |             |             |             |             |             | 9081  |
| Next: _                                     |                |             |             |             |             |             | Auto. |
|                                             | -----A-----    | -----B----- | -----C----- | -----D----- | -----E----- | -----F----- |       |
| 1                                           | Year           | 1984        | 1985        | 1986        | 1987        | 1988        |       |
| 2                                           | Cash flow      | 1000        | 1200        | 1500        | 2000        | 1000        |       |
| 3                                           |                |             |             |             |             |             |       |
| 4                                           | Discount rate  | 15          |             |             |             |             |       |
| 5                                           | Present value( | 4404)       |             |             |             |             |       |



In this example you can assume that in 1983 some money is to be invested and the figures above represent the expected yearly returns on that investment. To find out how the investment will perform, the yearly values each need to be converted to 'present values' and summed. In this instance they are all converted to 1983 values.

The first return in 1984 will be calculated as  $1000/(1+dr/100)$ . The 1000 is effectively worth less because of the one year taken to get it. The next year 1200 is obtained but this is worth less still because it is discounted once for 1984 and again in 1985 so its present value is calculated as  $1200/(1+dr/100)/(1+dr/100)$  and so on. The value of the return in n years is  $return/((1+dr)^n)$ .

The 'Internal rate of return' is the discount rate necessary to make the present value equal to the initial investment. It can be found by trial and error, changing the value of discount rate until you get the right answer.

If you want to integrate the area numerically then you would probably use Simpsons rule. You can use the function SIMPRULE to do this directly.

```

B8 (SIMPRULE(PI/8,B1...B5))
4D * ABCDEFGIJKLMNPQRSUVWXYZ+<->.* arrows
Next: -
=====A===== B=====
1 SINR(0) .0000
2 SINR(PI/8) .3827
3 SINR(PI/4) .7071
4 SINR(3*PI/8) .9239
5 SINR(PI/2) 1.0000
6
7 Integral of SINR(x)
8 between 0 and PI/2(1.0001)

```

In this example 5 values of SINR(X) have been calculated at intervals of PI/8. The SIMPRULE function has been used to obtain an approximate value of the integral. The exact value is 1.

The form of the function is SIMPRULE(step,range), the range must have an odd number of values.

# A NOTE ON LISTS

In most functions a list can be specified using a range such as B1...B5. You can however have blank entries at the end of your range and the function will still be worked out correctly. This feature allows you to set up a template worksheet and enter your particular data later. It will also cater for the situation where the number of items will be variable.



MATHEMATICAL FUNCTIONS

The usual trigonometric functions are available with versions for degrees and radians. The functions SIN, COS, TAN, ASIN, ACOS and ATAN refer to degrees. The functions starting with A in front represent the inverse values. SINR, COSR, TANR, ASINR, ACOSR and ATANR are the equivalent functions using radians. Each function takes just one argument.

```

B1 (SIN(30))
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows
Next: -
-----A-----B-----
1 SIN(30)(0.5)
2 COS(45) 0.7071068
3 TAN(60) 1.7320508
4 ASIN(0.2) 11.536959
5 ACOS(0.3) 72.542397
6 ATAN(0.5) 26.565051
7
8 SINR(PI/6) 0.5
9 COSR(PI/4) 0.7071068
10 TANR(PI/3) 1.7320508
11 ASINR(0.2) 0.2013579
12 ACOSR(0.3) 1.2661037
13 ATANR(0.5) 0.4636476

```

8779  
Auto.

So that you can see what is going on with just one screen example, the expressions in column B have been copied into column A in text form.

The natural logarithm is referred to as LN and the base 10

logarithm is LOG10. The natural antilogarithm is eAX and is referred to as EXP. The equivalent base 10 antilogarithm must be obtained by using 10AX where X is the value for which you want the antilogarithm.

The exponential constant e is available as function without an argument e. Similarly PI is available.

The square root is called with SQRT.

```

B1 (LN(3.5))
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows
Next: -
-----A-----B-----
1 LN(3.5)(1.252763)
2 EXP(B1) 3.5
3 LOG10(23.5) 1.3710679
4 10.B3 23.5
5
6 e 2.7182818
7 PI 3.1415927
8
9 SQRT(2) 1.4142136

```

9043  
Auto.



# UTILITY FUNCTIONS

Sum, minimum and maximum as SUM, MIN and MAX will scan a list and return the relevant value. COUNT will find the number of non-blank entries. All these functions are of the form FUNC(list).

GROW takes two arguments, a value and a percentage. The effect is to cause a percentage increase to the value. The form is GROW(value,percent).

The remaining functions in this section all act on a single argument and are of the form FUNC(value). The ABS absolute value of an argument is the value ignoring the sign. POS returns the value of the argument if it is positive but zero if it negative. The integer or non-decimal part of a number can be obtained with the function INT or if you want the the nearest whole number to a value use NINT. The decimal part of a number is given by the function DPART.

ROUND is a useful function that will round a value to 2 decimal places. It puts this rounded value back into memory and the original value is lost. You may need this useful function in complex financial calculations which may not balance because fractions of a penny (cent etc.) are having an effect.

RND returns a true integer random number between 0 and 127. This is taken from the refresh counter of the processor. RND is only available with the 280 processor.

```

B1 (SUM(3,2,5))
G * ABCDEFGI JLMNPQRSUVWXZ+~\/>.* arrows
Next: _
-----A-----B-----
1 SUM(3,2,5)(10)
2 MIN(3,2,5) 2
3 MAX(3,2,5) 5
4 COUNT(3,2,5) 3
5
6 GROW(50,5) 52.5
7 ABS(-12) 12
8 POS(12) 12
9 POS(-12) 0
10 INT(23.55) 23
11 NINT(23.55) 24
12 DPART(23.55) 0.55
13
14 ROUND(23.248) 23.25
15 RND 32
16

```

8719  
Auto.



STATISTICAL FUNCTIONS

Permutation and combination can be calculated with PERM(n,r) and COMB(n,r). For factorial use FACT(value).

|       |                                           |          |
|-------|-------------------------------------------|----------|
| B1    | (PERM(52,4))                              |          |
| G     | * ABCDEFGIJKLMNOPQRSTUVWXYZ+-\/>.* arrows |          |
| Next: | _____                                     |          |
|       | -----A-----B-----                         |          |
| 1     | PERM(52,4)(                               | 6497400) |
| 2     | COMB(52,4)                                | 270725   |
| 3     | FACT(9)                                   | 362880   |

Standard deviation and variance and average all act on a list to complete your set of powerful statistical tools. These functions are called STDEV, VAR and AVERAGE.

|       |                                           |           |
|-------|-------------------------------------------|-----------|
| B14   | (VAR(B1...B10))                           |           |
| G     | * ABCDEFGIJKLMNOPQRSTUVWXYZ+-\/>.* arrows |           |
| Next: | _____                                     |           |
|       | -----A-----B-----                         |           |
| 1     |                                           | 12.02     |
| 2     |                                           | 11.78     |
| 3     |                                           | 12.15     |
| 4     |                                           | 12.05     |
| 5     |                                           | 11.93     |
| 6     |                                           | 11.99     |
| 7     |                                           | 12.12     |
| 8     |                                           | 12.23     |
| 9     |                                           | 11.89     |
| 10    |                                           | 12.07     |
| 11    |                                           |           |
| 12    | AVERAGE(B1...B10)                         | 12.023    |
| 13    | STDEV(B1...B10)                           | 0.1324177 |
| 14    | VAR(B1...B10)                             | 0.0175344 |

(2)  
8901  
Auto.



DEFINED FUNCTIONS

A defined function is a single cell formula that can be used by other cells. After the first time you do not have to write the formula out in full. Consider the following formula which uses the value in A1. If you had to write it out 50 or 100 times it would get very time consuming and fast use up your available memory:

```

B1 (0.789*(A1+3)) (1)
OF * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 8347
Next: _ Auto.
-----A-----B-----
1 23.40 (20.83)
2
3
4

```

By using a defined function you only have to give the coordinate of the cell with the formula and follow it with a reference to the cell value on which you want it to operate. In this case the formula is in B1 and the value is A3. To call the function write the formula cell reference and follow it by a bracket just as if you were writing a function. Inside the brackets put the value cell reference:

```

B3
OF * Enter number or expression 8325
>> B1(A3)_ Auto.
-----A-----B-----
1 23.40 20.83
2
3 18.90 ()
4

```

In the defined function above you can read it as: Take the formula in B1 and replace the first cell reference found by A3. This is what THE CRACKER does. If there is more than one cell referred to in the original formula then you must have extra arguments in the defined function. Note that if the same cell is referred to more than once in the formula then it must also be repeated in the argument list. The number of arguments can be less than those in the formula. If this is the case the later cell references will remain unchanged. This can be useful.

```

B3 (B1(A3)) (2)
OF * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 8301
Next: _ Auto.
-----A-----B-----
1 23.40 20.83
2
3 18.90 (17.28)
4

```

The formula has acted on A3 to give the answer 17.28. Be warned that the defined function original formula must be kept separately in a work area. This is because when the formula is



reused the cell values referred to are changed every time the defined function is called.

|    |                              |             |                    |       |
|----|------------------------------|-------------|--------------------|-------|
| C3 |                              |             |                    | 8293  |
| OF | * Enter number or expression |             |                    | Auto. |
|    | >> C1(A3,B3)                 |             |                    |       |
|    | -----A-----                  | -----B----- | -----C-----        |       |
| 1  | (2)                          | (3)         | (1) ((A1+B1)*56.7) |       |
| 2  |                              |             |                    |       |
| 3  | (5)                          | (6)(        |                    |       |
| 4  |                              |             | )                  |       |

This is an example where there are two cell references in C1. The defined function call will be made in C3:

|    |                                          |             |             |       |
|----|------------------------------------------|-------------|-------------|-------|
| C3 | (C1(A3,B3))                              |             |             | (2)   |
| OF | * ABCDEFGHIJLMNPQRSUVWXYZ+-\ />.* arrows |             |             | 8266  |
|    | Next: -                                  |             |             | Auto. |
|    | -----A-----                              | -----B----- | -----C----- |       |
| 1  | 5.00                                     | 6.00        | 623.70      |       |
| 2  |                                          |             |             |       |
| 3  | 5.00                                     | 6.00 (      | 623.70 )    |       |
| 4  |                                          |             |             |       |

Note that the original values in A1 and B1 have been changed by the call.

# SEARCHING

You may want to find a particular location within a large worksheet without having to go through it systematically. To help you there is the GET command which goes through the columns and lines starting at the current cursor location looking for any string that you care to enter. Your string must be enclosed in single character delineators. Valid delineators are any characters that are printable but not letters or numerals. The string will be assumed to have been completed when the second matching delineator is found.

If you want to find the second occurrence of the same string then you only need to type a valid delineator twice and the string you last used will be automatically inserted between the characters.



```

A1 Init
T1 * /string/
Next: Get /43/
==A== ==B== ==C== ==D== ==E== ==F==
1(Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

In this example you are searching for the number 43 which is to be found at location F5:

```

F5 (43)
I * ABCDEFGHIJKLMNOPQRSTUVWXYZ+~\/>.* arrows
Next: _
==A== ==B== ==C== ==D== ==E== ==F==
1 Init Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall (43)
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

The cursor ends up at location of the string. You should remember that only the formulae and text entries are searched. If for example the last column had be in financial format and you had tried to search for 43.00 you would not find it even though it was displayed. As you can see from the contents line only 43 is actually stored in the memory. In practice this means you can only search for things that can be displayed on the contents line.

# SORTING THE LINES

THE CRACKER can selectively sort lines. You can specify the part or whole of a column you want to be used as the basis of the sort. Both text and values can be sorted, either increasing or decreasing. With this facility you can handle address lists and client lists. By using only part of columns in the sorts you carry out most of the activities that you would otherwise use a database management program for.

```

A1 Init
T1 * crd...crd <R>
Next: Sort lines using range: B3...B7_
==A== ==B== ==C== ==D== ==E== ==F==
1(Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

In this example the lines are going to be sorted using the name as the basis.



```

A1 Init
T1 *
Next: Increasing or decreasing I_
==A== ==B== ==C== ==D== ==E== ==F=
1 (Init)Name No Road Town £
2
3 R.T. Cowan 16 Jeremy Grove Hampton 17
4 G.L. Blake 37 Osmaston Road Knowle 56
5 A.J. Hunt 25 Latimer Road Wythall 43
6 H.A. Fisher 32 Florence Road Henley 28
7 C.D. Beard 86 Valley Road Redditch 25

```

```

B3 Beard
T1 * ABCDEFGIJKLMNOPQRSUVWXYZ+~\/>.* arrows
Next: _
==A== ==B== ==C== ==D== ==E== ==F=
1 Init Name No Road Town £
2
3 C.D.(Beard) 86 Valley Road Redditch 25
4 G.L. Blake 37 Osmaston Road Knowle 56
5 R.T. Cowan 16 Jeremy Grove Hampton 17
6 H.A. Fisher 32 Florence Road Henley 28
7 A.J. Hunt 25 Latimer Road Wythall 43

```

Column B is now in alphabetical order. Next you will see a numerical sort in decending order. Note that the sort is carried out on the internal value of the number and not on the numerals as displayed.

```

B3 Beard
T1 * crd...crd <R>
Next: Sort lines using range: F3...F7_
==A== ==B== ==C== ==D== ==E== ==F=
1 Init Name No Road Town £
2
3 C.D.(Beard) 86 Valley Road Redditch 25
4 G.L. Blake 37 Osmaston Road Knowle 56
5 R.T. Cowan 16 Jeremy Grove Hampton 17
6 H.A. Fisher 32 Florence Road Henley 28
7 A.J. Hunt 25 Latimer Road Wythall 43

```

```

B3 Beard
T1 * <R>
Next: Increasing or decreasing D_
==A== ==B== ==C== ==D== ==E== ==F=
1 Init Name No Road Town £
2
3 C.D.(Beard) 86 Valley Road Redditch 25
4 G.L. Blake 37 Osmaston Road Knowle 56
5 R.T. Cowan 16 Jeremy Grove Hampton 17
6 H.A. Fisher 32 Florence Road Henley 28
7 A.J. Hunt 25 Latimer Road Wythall 43

```

```

F3 (56)
I * ABCDEFGIJKLMNOPQRSUVWXYZ+~\/>.* arrows
Next: _
==A== ==B== ==C== ==D== ==E== ==F=
1 Init Name No Road Town £
2
3 G.L. Blake 37 Osmaston Road Knowle (56)
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17

```

The lines have now been sorted to make the numbers in column F decending.



PREPARATION OF MAIL LABELS

If you want to keep address lists then you probably also will want to prepare mailing labels. Blank labels are available on listing paper suitable for most printers. THE CRACKER will prepare these labels for you. Your mailing list will however need a little preparation. The first stage is to indicate the ends of the lines. To do this you must insert columns at the appropriate points, each with a Carriage return default format. It is very important that you put at least one such column at the end of the block you want printed. It is easy to forget this and the results will be unpredictable.

In this example you will not want to print the last column as it only contains a reference to the amount of money paid. You must start by putting a carriage return column in front of column C:

```
C1 No
T1 *
Next: Insert column , width 1_
==A= ==B== =C= =====D===== E===== F=
1 Init Name (No)Road Town £
2
3 G.L. Blake 37 Osmaston Road Knowle 56
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17
```

```
C1 No
T1 *
Next: Default Format to be Carriage return_
==A= ==B== =C= =====D===== E===== F=
1 Init Name (No)Road Town £
2
3 G.L. Blake 37 Osmaston Road Knowle 56
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17
```

Now two further such columns are needed one in front of the town column and one at the end in front of the £ column:

```
H1
C * ABCDEFGIJKLMNOPQRSUVWXZ+~\/>.* arrows
Next:
==A= ==B== C =D= =====E===== F =====G===== H =I=
1 Init Name No Road Town () £
2
3 G.L. Blake 37 Osmaston Road Knowle 56
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17
```

Now you are in a position to print out you first trial set of labels. You only want to print a portion and so you use the Copy block option:



```

H1
C * crd...crd <R> 7401
Next: Copy block A3...H7_ Auto.
 --A= ----B--- C =D= -----E----- F -----G----- H =I=
1 Init Name No Road Town () £
2
3 G.L. Blake 37 Osmaston Road Knowle 56
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17

```

By typing M2 after the request for destination you will be set up for having mail labels printed two abreast:

```

H1
C * #,<R> 7401
Next: Destination Mail Labels, groups of 2_ Auto.
 --A= ----B--- C =D= -----E----- F -----G----- H =I=
1 Init Name No Road Town () £
2
3 G.L. Blake 37 Osmaston Road Knowle 56
4 A.J. Hunt 25 Latimer Road Wythall 43
5 H.A. Fisher 32 Florence Road Henley 28
6 C.D. Beard 86 Valley Road Redditch 25
7 R.T. Cowan 16 Jeremy Grove Hampton 17

```

And here is what they will look like. Not very organised, but your next task is to change the width of the columns so that various parts line up under one another:

```

G.L. Blake A.J. Hunt
37 Osmaston Road 25 Latimer Road
Knowle Wythall

```

```

H.A. Fisher C.D. Beard
32 Florence Road 86 Valley Road
Henley Redditch

```

```

R.T. Cowan
16 Jeremy Grove
Hampton

```

After some adjustment here is what you can achieve. In this case the labels are going to be printed three abreast:

```

A1 Init
T1 * crd...crd <R> 7401
Next: Copy block A3...H7_ Auto.
 --A= ----B--- C =D= -----E----- F -----G-----
1(Init)Name No Road Town -
2
3 G.L. Blake 37 Osmaston Road Knowle -
4 A.J. Hunt 25 Latimer Road Wythall -
5 H.A. Fisher 32 Florence Road Henley -
6 C.D. Beard 86 Valley Road Redditch -
7 R.T. Cowan 16 Jeremy Grove Hampton -

```

```

A1 Init
T1 * #,<R> 7401
Next: Destination Mail Labels, groups of 3_ Auto.
 --A= ----B--- C =D= -----E----- F -----G-----
1(Init)Name No Road Town -
2
3 G.L. Blake 37 Osmaston Road Knowle -
4 A.J. Hunt 25 Latimer Road Wythall -
5 H.A. Fisher 32 Florence Road Henley -
6 C.D. Beard 86 Valley Road Redditch -
7 R.T. Cowan 16 Jeremy Grove Hampton -

```



G.L. Blake  
37 Osmaston Road  
Knowle

A.J. Hunt  
25 Latimer Road  
Wythall

H.A. Fisher  
32 Florence Road  
Henley

C.D. Beard  
86 Valley Road  
Redditch

R.T. Cowan  
16 Jeremy Grove  
Hampton

Futher adjusment may be needed with extra lines at the end or changes in spacing to suit the particular labels. You will find that this is quick and easy by trial and error.

## THE CRACKER TUTORIAL V ADVANCED TECHNIQUES

### FUNCTIONS THAT ALLOW LOOPING

Sometimes it would be very useful if you could use a few formulae repetitively to work towards an answer. With the 'internal rate of return' you have to keep trying values until you get close enough to the answer. In fact there is a way in which you could have set up a short entry that would have tried a range of possibilities and stopped at the nearest.

The functions that allow you to do this are DO and WHILE. The DO function specifies the section over which you want to loop. Note that you must specify a range so your working must be in one line or one column and not in a block. The function looks at the range and finds the highest and lowest recalculation number and then all those entries between the numbers are recalculated. It does not matter if some of your intermediate calculation numbers are not in the specified range as they will still be correctly calculated.

After the DO function put in a comma then any expression or function that you wish. Usually this function will change a value somewhere and act a loop counter. The function that make this easy for you are INIT, SET, INC and DEC. Which respectively initialise a cell entry, set a value to it, increment it and decrement it.

After the loop counter section put in a further comma and then use the WHILE function. This function has a logical argument which might be a logical expression finding whether the loop counter has reached a certain value. If the WHILE function is



TRUE then the cell formula is started again at the DO and repeated to the WHILE until the WHILE becomes FALSE.

That is a basic description and it will seem fairly complicated at this stage. Do not worry about it yet, follow through the examples and then go back and look at this section again.

First you are going to see how the SET function works. It is of the form SET(coord,value). The value can also be an expression and can include the referenced coordinate. SET(A2,A2+1) is valid and works in exactly the same way as INC(A2).

|    |                              |       |
|----|------------------------------|-------|
| A1 |                              | 9188  |
| G  | * Enter number or expression | Auto. |
|    | >> SET(A2,5)                 |       |
|    | -----A-----                  |       |
| 1  | (                            | )     |
| 2  |                              | 5     |
| 3  |                              |       |
| 4  |                              |       |
| 5  |                              |       |

You may note that the destination cell is set even before the expression is fully entered.

|       |                                          |       |
|-------|------------------------------------------|-------|
| A1    | (SET(A2,5))                              | (1)   |
| G     | * ABCDEFGIJKLMNOPQRSUVWXYZ+-\/>.* arrows | 9140  |
| Next: | -----A-----                              | Auto. |
| 1     | (                                        | 5)    |
| 2     |                                          | 5     |
| 3     |                                          |       |
| 4     |                                          |       |
| 5     |                                          |       |

Next see how the INC increment function works. This is not a realistic expression by itself but you will see later how it can form part of a DO WHILE loop. Once again the function works even before the expression is completed, so as you enter such a formula you will see A2 first have the value 5 and then 6 as it is incremented.

|    |                              |       |
|----|------------------------------|-------|
| A1 |                              | 9188  |
| G  | * Enter number or expression | Auto. |
|    | >> SET(A2,5),INC(A2)         |       |
|    | -----A-----                  |       |
| 1  | (                            | )     |
| 2  |                              | 6     |
| 3  |                              |       |
| 4  |                              |       |
| 5  |                              |       |

|       |                                          |       |
|-------|------------------------------------------|-------|
| A1    | (SET(A2,5),INC(A2))                      | (1)   |
| G     | * ABCDEFGIJKLMNOPQRSUVWXYZ+-\/>.* arrows | 9132  |
| Next: | -----A-----                              | Auto. |
| 1     | (                                        | 6)    |
| 2     |                                          | 6     |
| 3     |                                          |       |
| 4     |                                          |       |
| 5     |                                          |       |

The value that ends up in A1 is that of the expression or function after the last dividing comma. It is valid to put a further comma in and then an expression or value. The effect of dividing commas is to cause an effective restart as if what follows was at the beginning of the line. You can understand that only functions like SET, INC and DEC have any actual effect on the worksheet if there is a later dividing comma:



|    |                              |       |
|----|------------------------------|-------|
| A1 |                              |       |
| G  | * Enter number or expression | 9188  |
|    | >> SET(A2,5),INC(A2),2*A2_   | Auto. |
|    | -----A-----                  |       |
| 1  | (                            | )     |
| 2  |                              | 6     |
| 3  |                              |       |
| 4  |                              |       |
| 5  |                              |       |

|       |                                          |       |
|-------|------------------------------------------|-------|
| A1    | (SET(A2,5),INC(A2),2*A2)                 | (1)   |
| G     | * ABCDEFGIJKLMNOPQRSTUVWXYZ+</>.* arrows | 9127  |
| Next: | -----A-----                              | Auto. |
| 1     | (                                        | 12)   |
| 2     |                                          | 6     |
| 3     |                                          |       |
| 4     |                                          |       |
| 5     |                                          |       |

In this case the entry at A1 has ended up with the value 2\*A2 which is 12. The cell entries as a whole have become a series of instructions executed in turn.

SUBROUTINES USING THE DO FUNCTION

Now you are going to see how to use the DO function as a subroutine call. You may be familiar with subroutines but if not here is an explanation.

A subroutine is a group of formulae which you may want to use repeatedly. Rather than enter the formulae many times over you access them with the DO function. Every time the function is found in a cell during recalculation the whole of the group is calculated again. Now have a look at it in practice:

|       |                                          |       |
|-------|------------------------------------------|-------|
| A3    | (2*A4)                                   | (1)   |
| G     | * ABCDEFGIJKLMNOPQRSTUVWXYZ+</>.* arrows | 9147  |
| Next: | -----A-----                              | Auto. |
| 1     |                                          |       |
| 2     |                                          |       |
| 3     | (                                        | 6)    |
| 4     |                                          | 3     |

Here is a small group, in this case just 2 cells. The formula is also very simple. You will see how the DO function is used to find out what 2 times any number is, not just the 3 above:



```

A1
G * Enter number or expression
 >> SET(A4,5)_
 -----A-----
1(
2
3 6
4 5

```

9147  
Auto.

First you use the SET function to place the number you want doubled into A4 so that A3 can access it. Note it has already changed:

```

A1
G *
 >> SET(A4,5),DO(A3...A4)_
 -----A-----
1(
2
3 6
4 5

```

9147  
Auto.

Then you put in a comma and the DO function. The argument of the DO function must be a range it cannot be individual entries or a block.

```

A1 (SET(A4,5),DO(A3...A4))
G * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
 -----A-----
1(
2 5)
3 10
4 5

```

(2)  
9106  
Auto.

When you complete the expression the calculation is carried out.

But you do not have the answer you want in cell A1. This can be got by following the expression by a reference to the cell with the answer:

```

A1
G * Enter number or expression
 >> SET(A4,5),DO(A3...A4),A3_
 -----A-----
1(
2
3 10
4 5

```

9145  
Auto.

This time you have the answer you want. Now you can use the group to multiply 7 by 2. You can do this in cell A2:

```

A2
G * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows
Next: _
 -----A-----
1 10
2(
3 10
4 7

```

9103  
Auto.

```

A2
G * Enter number or expression
 >> SET(A4,7),DO(A3...A4),A3_
 -----A-----
1 10
2(
3 10
4 7

```

9103  
Auto.



|             |                                            |       |
|-------------|--------------------------------------------|-------|
| A2          | (SET(A4,7),DO(A3...A4),A3)                 | (2)   |
| G           | * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\ />.* arrows | 9061  |
| Next:       | _                                          | Auto. |
| -----A----- |                                            |       |
| 1           | 10                                         |       |
| 2           | 14)                                        |       |
| 3           | 10                                         |       |
| 4           | 5                                          |       |

Thus you end up with 14 in cell A2 as well as the 10 in A1. This is a trivial example but the same technique can be used for much larger sections of the worksheet. The group used for subroutine calculation should not be used in the main body of the worksheet calculation as it will give nonsense values. You are after all changing the constants in it several times during the calculation.

You can change several values with SET commands before each DO function.

LOOPING USING THE FUNCTIONS DO AND WHILE

Your first entry when setting up a loop is to establish a counter. Do this with the INIT function. This function is almost identical to SET but THE CRACKER will make sure it gets a very low calculation number and hence will not be affected by later workings:

|             |                              |       |
|-------------|------------------------------|-------|
| A1          |                              |       |
| G           | * Enter number or expression | 9188  |
|             | >> INIT(A2,0)_               | Auto. |
| -----A----- |                              |       |
| 1           | (                            | )     |
| 2           |                              | 0     |
| 3           |                              |       |
| 4           |                              |       |
| 5           |                              |       |

|             |                                            |       |
|-------------|--------------------------------------------|-------|
| A1          | (INIT(A2,0))                               | (1)   |
| G           | * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\ />.* arrows | 9140  |
| Next:       | _                                          | Auto. |
| -----A----- |                                            |       |
| 1           | (                                          | 0)    |
| 2           |                                            | 0     |
| 3           |                                            |       |
| 4           |                                            |       |
| 5           |                                            |       |

The counter is A2. The object of this example is to multiply the



value of the counter by 2 and then increment the value and do it again. A criterion for stopping will be specified.

|    |                              |   |       |
|----|------------------------------|---|-------|
| A5 |                              |   |       |
| G  | * Enter number or expression |   | 9140  |
|    | >> A2_                       |   | Auto. |
|    | =====A=====                  |   |       |
| 1  |                              | 0 |       |
| 2  |                              | 0 |       |
| 3  |                              |   |       |
| 4  |                              |   |       |
| 5( |                              | ) |       |

Here a reference to the counter is placed in A5:

|    |                                           |    |       |
|----|-------------------------------------------|----|-------|
| A5 | (A2)                                      |    | (2)   |
| G  | * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows |    | 9120  |
|    | Next: _                                   |    | Auto. |
|    | =====A=====                               |    |       |
| 1  |                                           | 0  |       |
| 2  |                                           | 0  |       |
| 3  |                                           |    |       |
| 4  |                                           |    |       |
| 5( |                                           | 0) |       |

At this stage it has the value 0:

|    |                              |   |       |
|----|------------------------------|---|-------|
| A4 |                              |   |       |
| G  | * Enter number or expression |   | 9120  |
|    | >> 2*A5_                     |   | Auto. |
|    | =====A=====                  |   |       |
| 1  |                              | 0 |       |
| 2  |                              | 0 |       |
| 3  |                              |   |       |
| 4( |                              | ) |       |
| 5  |                              | 0 |       |

Now the cursor moves to A4 and the formula for multiplying by 2 is entered. The counter and formula are now set up so the loop can be established:

|    |                              |   |       |
|----|------------------------------|---|-------|
| A3 |                              |   |       |
| G  | * Enter number or expression |   | 9098  |
|    | >> DO(A4...A5),INC(A2)_      |   | Auto. |
|    | =====A=====                  |   |       |
| 1  |                              | 0 |       |
| 2  |                              | 1 |       |
| 3( |                              | ) |       |
| 4  |                              | 0 |       |
| 5  |                              | 0 |       |

So far it is much as with the subroutine examples:

|    |                                     |   |       |
|----|-------------------------------------|---|-------|
| A3 |                                     |   |       |
| G  | * DO(A4...A5),INC(A2),WHILE(A2<10)_ |   | 9098  |
|    | =====A=====                         |   | Auto. |
| 1  |                                     | 0 |       |
| 2  |                                     | 1 |       |
| 3( |                                     | ) |       |
| 4  |                                     | 0 |       |
| 5  |                                     | 0 |       |

The WHILE function has a special property that if the argument is TRUE then the expression is wound back to the DO and repeated. It remains TRUE while A2 is less than 10. But note that after the DO function the counter A2 is incremented. The multiply by 2 formula therefore has a different start value. Overall the effect is that on the screen you see A5 going from 0 to 9 while A4 goes from 0 to 18. Not a lot of use in this case but it shows the loop working 10 times. Later you will see the looping used to fill a table.

To make it easier to see the rules of the spreadsheet you can convert the display to show just the formulae. First the column width needs to be increased:



|       |                                    |       |
|-------|------------------------------------|-------|
| A3    | (DO(A4...A5),INC(A2),WHILE(A2<10)) | (4)   |
| G     | * #,<R>                            | 9048  |
| Next: | New width of column 45_            | Auto. |
|       | -----A-----                        |       |
| 1     |                                    | 0     |
| 2     |                                    | 10    |
| 3(    |                                    | 0)    |
| 4     |                                    | 18    |
| 5     |                                    | 9     |

Then the X for eXchange command is used:

|       |                                    |       |
|-------|------------------------------------|-------|
| A3    | (DO(A4...A5),INC(A2),WHILE(A2<10)) | (4)   |
| G     | * <R>                              | 9048  |
| Next: | eXchange rules/results_            | Auto. |
|       | -----A-----                        |       |
| 1     |                                    | 0     |
| 2     |                                    | 10    |
| 3(    |                                    | 0)    |
| 4     |                                    | 18    |
| 5     |                                    | 9     |

|       |                                        |       |
|-------|----------------------------------------|-------|
| A3    | (DO(A4...A5),INC(A2),WHILE(A2<10))     | (4)   |
| G     | * ABCDEFGIJLMNPQRSUVWXYZ+~ />.* arrows | 9048  |
| Next: | _____                                  | Auto. |
|       | -----A-----                            |       |
| 1     | (1) (INIT(A2,0))                       |       |
| 2     | (set)                                  |       |
| 3(    | (4) (DO(A4...A5),INC(A2),WHILE(A2<10)) | )     |
| 4     | (3) (2*A5)                             |       |
| 5     | (2) (A2)                               |       |

You can now see all the formulae. The number in front of the formula is the recalculation order number. This is the same as the one you see in the top right hand corner.

The way the looping is organised means that the DO formulae will always be calculated once before the WHILE is tested. This ties in with the way DO and WHILE work in the main computer languages.



TABLE FILLING USING THE DO FUNCTION

You are going to see how to fill a table with the values of Sine(x) between 10 and 90 degrees. As before you start by initialising a counter in this case cell A2 is set up with 1:

|    |                              |             |       |
|----|------------------------------|-------------|-------|
| A1 |                              |             |       |
| G  | * Enter number or expression |             | 8399  |
|    | >> INIT(A2,1)_               |             | Auto. |
|    | -----A-----                  | -----B----- |       |
| 1  | (                            | )           |       |
| 2  |                              | 1           |       |
| 3  |                              |             |       |
| 4  |                              |             |       |

As you want the Sine values every 10 degrees you can use the counter multiplied by 10. This formula is inserted in cell A3:

|    |                              |             |       |
|----|------------------------------|-------------|-------|
| A3 |                              |             |       |
| G  | * Enter number or expression |             | 8350  |
|    | >> 10*A2_                    |             | Auto. |
|    | -----A-----                  | -----B----- |       |
| 1  |                              | 1           |       |
| 2  |                              | 1           |       |
| 3  | (                            | )           |       |
| 4  |                              |             |       |

Now you have value for degrees you can refer to it with a SIN function in cell A4:

|    |                              |             |       |
|----|------------------------------|-------------|-------|
| A4 |                              |             |       |
| G  | * Enter number or expression |             | 8327  |
|    | >> SIN(A3)_                  |             | Auto. |
|    | -----A-----                  | -----B----- |       |
| 1  |                              | 1           |       |
| 2  |                              | 1           |       |
| 3  |                              | 10          |       |
| 4  | (                            | )           |       |

The object is to fill a table in column B with the Sine values at every 10 degrees. Somehow you must transmit the calculated value to the required slot. Once more you can use the SET function but together with the CRD function. This special function can be used anywhere you would use a coordinate reference. The two arguments are calculated and depending on their values the function is actually set to be a cell reference.

|    |                              |                  |             |
|----|------------------------------|------------------|-------------|
| A5 |                              |                  |             |
| G  | * Enter number or expression |                  | 8302        |
|    | >> SET(CRD(2,A2),A4)_        |                  | Auto.       |
|    | -----A-----                  | -----B-----      |             |
| 1  |                              | 1                | 0.173648178 |
| 2  |                              | 1                |             |
| 3  |                              | 10               |             |
| 4  |                              | 0.17364817766693 |             |
| 5  | (                            | )                |             |
| 6  |                              |                  |             |

The CRD function is of the form CRD(column,row). In the case above the column number is 2 which is column B. The row number is the value of A2 which is 1. The CRD function will therefore become the cell reference B1 and will behave exactly as if the entry read SET(B1,A4). As you can see the cell B1 has already been set to the value of A4. Now A2 is a counter and can be made to change in value by using a DO function. As A2 changes so will the CRD function become different cell references. In this way we can fill a column of cells with values and form a table.



The next step is for you to enter the DO - WHILE loop:

```

A6
G * Enter number or expression 8246
 >> DO(A3...A5),INC(A2),WHILE(A2[9])_ Auto.
 =====A=====B=====
 1 1 0.173648178
 2 2
 3 10
 4 0.17364817766693
 5 0.17364817766693
 6()

```

This says calculate the Sine values as set out in cells A3...A5 then up the counter by 1 and do it again as long as the counter is less than or equal to 9:

```

A6
G * Calculating 8197
 >> DO(A3...A5),INC(A2),WHILE(A2[9])_ Auto.
 =====A=====B=====
 1 1 0.173648178
 2 6 0.342020143
 3 50 0.5
 4 0.766044443118978 0.64278761
 5 0.766044443118978 0.766044443
 6()
 7

```

You can see in the screen above the state of the display after the first 5 Sine values have been calculated.

```

A6 (DO(A3...A5),INC(A2),WHILE(A2[9])) (5)
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 8197
Next: _ Auto.
=====A=====B=====
1 1 0.173648178
2 10 0.342020143
3 90 0.5
4 1 0.64278761
5 1 0.766044443
6(0) 0.866025404
7 0.939692621
8 0.984807753
9 1
10
11

```

The calculation has now been completed and below you can see the full set of formulae used. To get this display you use the X command:

```

A6 (DO(A3...A5),INC(A2),WHILE(A2[9])) (5)
G * ABCDEFGIJKLMNOPQRSTUVWXYZ+~\/>.* arrows 8197
Next: _ Auto.
=====A=====B=====
1 (1) (INIT(A2,1)) (set)
2 (set) (set)
3 (2) (10*A2) (set)
4 (3) (SIN(A3)) (set)
5 (4) (SET(CRD(2,A2),A4)) (set)
6(5) (DO(A3...A5),INC(A2),WHILE(A2[9])) (set)
7 (set)
8 (set)
9 (set)
10
11

```



You can be more sophisticated by actually having the degrees presented as well. To do this you must first insert another column and a line at row 6.

```

A6 (SET(CRD(2,A2),A4)) (4)
G * ABCDEFGIJKLMNOPQRSUVWXZ+~|/>.* arrows 8239
Next: _ Auto.
=====A===== --B= ----C==
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (3) (SIN(A3))
5 (4) (SET(CRD(2,A2),A4))
6(
7 (5) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9
10
11

```

With the new column B inserted you can see that the CRD function in cell A5 needs adjusting to refer to column C. This is not done automatically for you and so you will have to re-enter it or use the Edit command to change the first 2 to a 3.

```

A5 (SET(CRD(2,A2),A4)) (4)
G * <R> 8029
Next: Edit_ Auto.
=====A===== --B= ----C==
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (3) (SIN(A3))
5((4) (SET(CRD(2,A2),A4)))
6
7 (5) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9
10
11

```

```

A5 (SET(CRD(2,A2),A4)) (4)
G * 1,X,Z,space,,<R> 8029
Edit: SET(CRD(2,A2),A4) Auto.
=====A===== --B= ----C==
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (3) (SIN(A3))
5((4) (SET(CRD(2,A2),A4)))
6
7 (5) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9
10
11

```

You move to the relevant position using the space bar and then type an X for eXchange. Type the 3 and the press the <R> twice, once to get out of the exchange mode and the second to get out of the edit mode.

```

A5 (SET(CRD(2,A2),A4)) (4)
G * "Xchange" char,<R> 8029
Edit: SET(CRD(3,A2),A4) Auto.
=====A===== --B= ----C==
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (3) (SIN(A3))
5((4) (SET(CRD(2,A2),A4)))
6
7 (5) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9
10
11

```

This editing will force a recalculation:



|       |                                         |         |
|-------|-----------------------------------------|---------|
| A5    | (SET(CRD(3,A2),A4))                     | (4)     |
| G     | * ABCDEFGIJKLMNOPQRSUVWXYZ+</>.* arrows | 3029    |
| Next: | Auto.                                   |         |
|       | =====A=====                             | ==B==   |
| 1     | 1                                       | 0.17365 |
| 2     | 10                                      | 0.34202 |
| 3     | 90                                      | 0.5     |
| 4     | 1                                       | 0.64279 |
| 5     | 1)                                      | 0.76604 |
| 6     |                                         | 0.86603 |
| 7     | 0                                       | 0.93969 |
| 8     |                                         | 0.98481 |
| 9     |                                         | 1       |
| 10    |                                         |         |
| 11    |                                         |         |

Now you can enter the extra SET function to fill up column B with the degree values which are calculated in cell A3:

|    |                                        |       |
|----|----------------------------------------|-------|
| A6 | (SET(CRD(2,A2),A3))                    | (3)   |
| G  | * Enter number or expression           | 7973  |
| >> | SET(CRD(2,A2),A3)                      | Auto. |
|    | =====A=====                            | ==B== |
| 1  | (1) (INIT(A2,1))                       |       |
| 2  | (set)                                  |       |
| 3  | (2) (10*A2)                            |       |
| 4  | (3) (SIN(A3))                          |       |
| 5  | (4) (SET(CRD(3,A2),A4))                |       |
| 6  |                                        |       |
| 7  | (5) (DO(A3...A5),INC(A2),WHILE(A2[9])) |       |
| 8  |                                        |       |
| 9  |                                        |       |
| 10 |                                        |       |
| 11 |                                        |       |

|       |                                         |       |
|-------|-----------------------------------------|-------|
| A6    | (SET(CRD(2,A2),A3))                     | (3)   |
| G     | * ABCDEFGIJKLMNOPQRSUVWXYZ+</>.* arrows | 7973  |
| Next: | Auto.                                   |       |
|       | =====A=====                             | ==B== |
| 1     | (1) (INIT(A2,1))                        |       |
| 2     | (set)                                   |       |
| 3     | (2) (10*A2)                             |       |
| 4     | (4) (SIN(A3))                           |       |
| 5     | (5) (SET(CRD(3,A2),A4))                 |       |
| 6     | (3) (SET(CRD(2,A2),A3))                 |       |
| 7     | (6) (DO(A3...A5),INC(A2),WHILE(A2[9]))  |       |
| 8     |                                         |       |
| 9     |                                         |       |
| 10    |                                         |       |
| 11    |                                         |       |

This will also force a recalculation to give the tables you want:

|       |                                         |            |
|-------|-----------------------------------------|------------|
| A6    | (SET(CRD(2,A2),A3))                     | (3)        |
| G     | * ABCDEFGIJKLMNOPQRSUVWXYZ+</>.* arrows | 7973       |
| Next: | Auto.                                   |            |
|       | =====A=====                             | ==B==      |
| 1     | 1                                       | 10 0.17365 |
| 2     | 10                                      | 20 0.34202 |
| 3     | 90                                      | 30 0.5     |
| 4     | 1                                       | 40 0.64279 |
| 5     | 1                                       | 50 0.76604 |
| 6     | 90)                                     | 60 0.86603 |
| 7     | 0                                       | 70 0.93969 |
| 8     |                                         | 80 0.98481 |
| 9     |                                         | 90 1       |
| 10    |                                         |            |
| 11    |                                         |            |



Next you are going to use the tables in columns B and C to interpolate a value for 25 degrees. You can enter the INTERP function in the following form to get the value you want:

```

A9
G * Enter number or expression 7973
 >> INTERP(25,B1...B9) Auto.
 -----A-----B-----C-----
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (4) (SIN(A3))
5 (5) (SET(CRD(3,A2),A4))
6 (3) (SET(CRD(2,A2),A3))
7 (6) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9(
10
11

```

But this does not work as an error message about an ambiguity comes up. This is because you are trying to operate on a calculated table and THE CRACKER can not resolve just exactly what calculation number to give your new entry:

```

A9
G * Ambiguity (see manual) 7973
 >> INTERP(25,B1...B9) Auto.
 -----A-----B-----C-----
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (4) (SIN(A3))
5 (5) (SET(CRD(3,A2),A4))
6 (3) (SET(CRD(2,A2),A3))
7 (6) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9(
10
11

```

This is easily solved by putting a reference to the cell that created the table. In this case it was the DO function cell, so you have to put A7, before your INTERP function. This forces your new function to have a calculation number greater than that of the DO function.

```

A9
G * Enter number or expression 7805
 >> A7,INTERP(25,B1...B9) Auto.
 -----A-----B-----C-----
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (4) (SIN(A3))
5 (5) (SET(CRD(3,A2),A4))
6 (3) (SET(CRD(2,A2),A3))
7 (6) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9(
10
11

```

Note the DO function has a calculation number of 6 and the INTERP follows it with 7.

```

A9 (A7,INTERP(25,B1...B9)) (7)
G * ABCDEFGIJKLMNPQRSUVWXYZ+~\/>.* arrows 7766
Next: _ Auto.
 -----A-----B-----C-----
1 (1) (INIT(A2,1))
2 (set)
3 (2) (10*A2)
4 (4) (SIN(A3))
5 (5) (SET(CRD(3,A2),A4))
6 (3) (SET(CRD(2,A2),A3))
7 (6) (DO(A3...A5),INC(A2),WHILE(A2[9]))
8
9((7) (A7,INTERP(25,B1...B9)))
10
11

```



If you use the X command you can see the result of the interpolation which gives value exactly half way between the Sine of 20 and the Sine of 30 as you would expect:

|       |                                      |                   |
|-------|--------------------------------------|-------------------|
| A9    | (A7,INTERP(25,B1...89))              | (7)               |
| G     | * ABCDEFGIJLMNPQRSUVWXZ+</>.* arrows | 7766              |
| Next: | Auto.                                |                   |
| 1     | -----A-----                          | -----B-----C----- |
| 2     |                                      | 1 10 0.17365      |
| 3     |                                      | 10 20 0.34202     |
| 4     |                                      | 90 30 0.5         |
| 5     |                                      | 1 40 0.64279      |
| 6     |                                      | 1 50 0.76604      |
| 7     |                                      | 90 60 0.86603     |
| 8     |                                      | 0 70 0.93969      |
| 9(    | 0.421010071662834)                   | 80 0.98481        |
| 10    |                                      | 90 1              |
| 11    |                                      |                   |

The CRD function should only be used with INIT, SET, INC and DEC to specify the cell to be acted upon. It will not give the value of the cell if used in any other expressions. The function that will do this for you is the VAL or value function which returns the current value of the cell to which it refers. The arguments are formed in the same way as for the CRD function. Here is an example of how the VAL function performs as you would want it to and the CRD function just returns a value of zero:

|       |                                      |             |
|-------|--------------------------------------|-------------|
| A3    | VAL(1,1)                             |             |
| Tr    | * ABCDEFGIJLMNPQRSUVWXZ+</>.* arrows | 8287        |
| Next: | Auto.                                |             |
| 1     | -----A-----                          | -----B----- |
| 2     |                                      | 3           |
| 3(    | CRD(1,1)                             | 0           |
|       | VAL(1,1))                            | 3           |

## THE CRACKER COMMAND REFERENCE

### DEFINITION OF TERMS

|             |                                                                                                                                                                                                                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adjust      | Adjust references to new location. Any coordinate references in that section will, if desired, be adjusted to suit the new location. If a line is copied and has internal references to the copied line itself these will be adjusted to the line number of the destination. Similarly when a column is copied. |
| All         | refers to the whole of the worksheet.                                                                                                                                                                                                                                                                           |
| Block       | refers to a rectangular portion of the worksheet. This is defined by the coordinates of the cells at the top left and bottom right of the rectangle.                                                                                                                                                            |
| Column      | is defined as all the cells in a vertical line of the worksheet.                                                                                                                                                                                                                                                |
| Destination | refers to the cell coordinate, filename, printer or mail labels. This defines where the results of an operation are to be placed. The cell coordinate is indicated by moving the cursor using the arrows or U,D,L,R or alternatively with the JUMP command.                                                     |
| Entry       | is the cell at which the cursor currently points.                                                                                                                                                                                                                                                               |
| End         | refers to the location of the end of the current worksheet                                                                                                                                                                                                                                                      |
| Filename    | Two types of file can be copied into memory defined by their                                                                                                                                                                                                                                                    |



extensions, .MEM and .DAT. Three types of file can be copied from memory, and these are .MEM, .DAT, .TXT. The definition of a filename follows the standard CP/M form, A:FILENAME.EXT. The disk unit definition A: is optional, the FILENAME can be up to 8 alphanumeric characters as a maximum and the .EXT is as explained above.

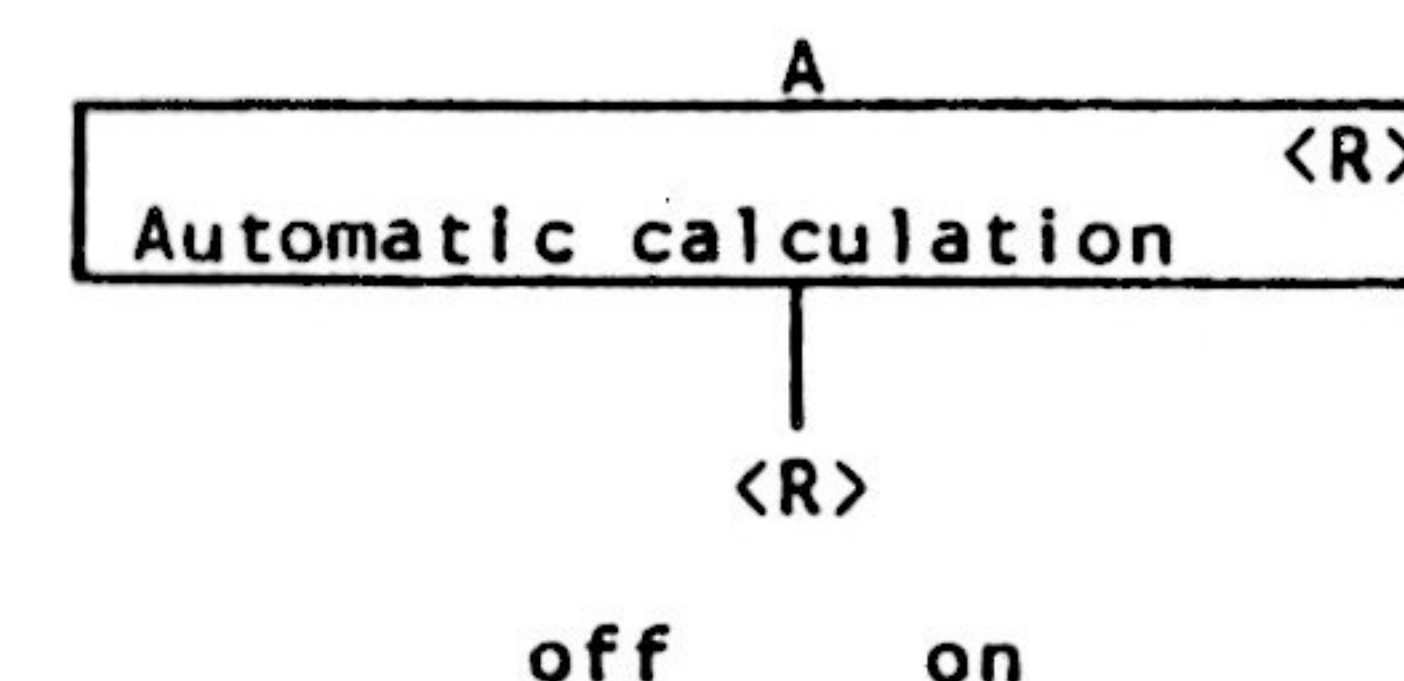
Line is defined as all the cells in a horizontal row of the worksheet.

List is a series of values which may be in the form of numerals, expressions, coordinates or ranges.

Width is the default displayed width of a cell.

AUTOMATIC CALCULATION COMMAND

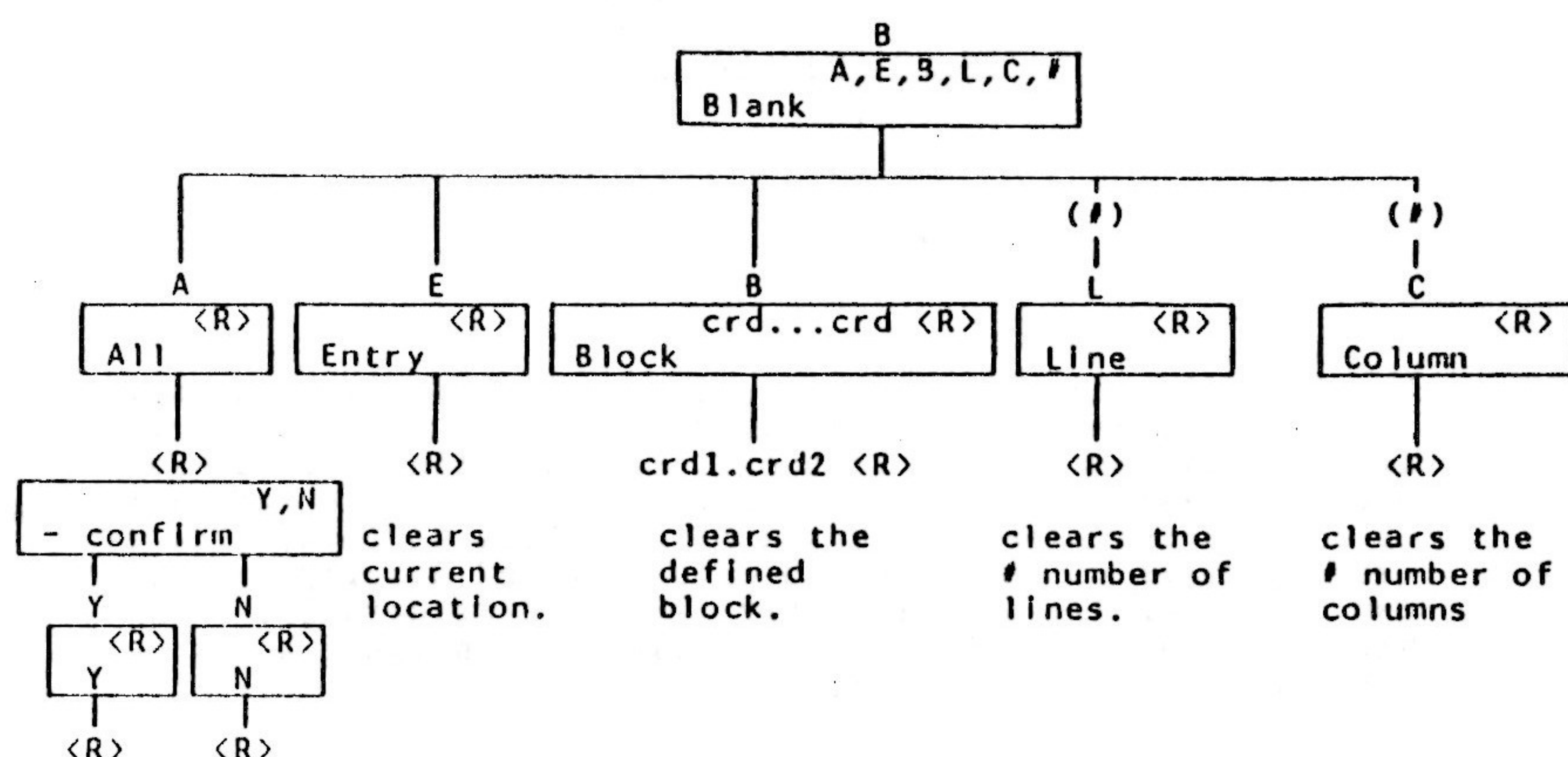
---



You initiate this command by typing an A followed by a carriage return. Similarly a second entry of A followed by carriage return will remove this setting. You will find this command is useful to switch off the calculation process while entering long or complex formulae or perhaps large tables of data. The few seconds taken up each time to calculate and display the results can slow down the overall entry process considerably. When you switch on the automatic calculation again a full recalculation will be carried out.



## BLANK COMMAND



BLANK removes the defined entries from memory but does not affect the structure of the worksheet. Blanked entries cannot be retrieved so if in doubt you should first copy the memory to a file before embarking on complex rearrangements.

Before the blanking is carried out the program will check whether any of the items to be blanked are referred to elsewhere in the worksheet. If cross-references are found the command will not be carried out.

## Cross-reference checking

A section of the sheet cannot be deleted or blanked if there are other parts of the sheet which depend upon values that are about to be deleted. This does not however apply if the cross-reference is within a range sum as SUM(A1...A9). Similarly if a section of memory is to be written to file it must not make references outside its own area. That section must be able to stand on its own so that filed sections to be merged in do not

interfere with the running of the worksheet. In this way the dynamic error checking is maintained at all times. If you have trouble in erasing by line or column go back to the Blank Entry command and remove the entries one by one in the reverse order of their calculation numbers.



## CONTROL CODE COMMANDS

CTRL-C Abort command, returns directly to CP/M.  
 CTRL-S Pause command, operations will cease until CTRL-S is typed again.  
 CTRL-F Snapshot taken of screen and put to file.  
 CTRL-P Snapshot taken of screen and printed.

These are four special commands all initiated by holding down the CTRL key and at the same time typing the character following the dash.

CTRL-C should only ever be used with extreme care as no copy of your work will be saved. As a general rule always use the QUIT command instead.

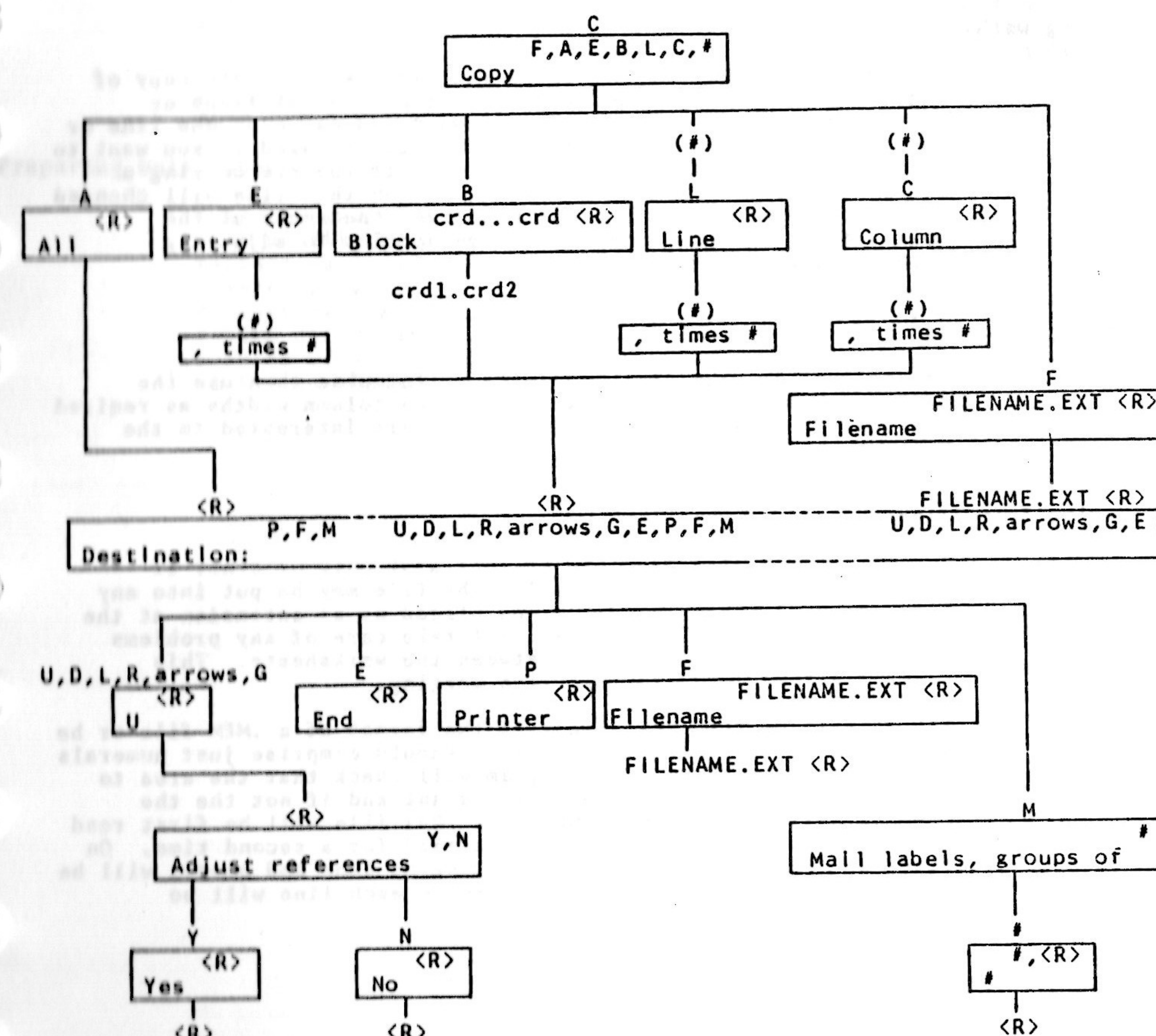
CTRL-S is a pause command that will suspend operations until a second CTRL-S is typed.

CTRL-F stops operations immediately and takes a copy of the screen and sends it to a file. This and the CTRL-P commands will only work if you have installed the remote page send screen commands when setting up THE CRACKER. The first file will be named DUMP-A and if the you take a second snapshot the file will be named DUMP-B and so on. These files are standard text files that can be handled with an editor or word processor. They can not be read by THE CRACKER.

CTRL-P stops operations and immediately sends a copy of the screen to your printer.

If your remaining memory is not enough then the CTRL-F and CTRL-P features will cease to work. This will free more space for your worksheet. The machine will beep to tell you this is the situation.

## COPY COMMAND





The COPY command is used to transfer copies of sections of the memory, displayed results or files to other locations, files or the printer. There are restrictions with just what can be transferred to where but even so this is the most useful command.

#### Copying within the worksheet

You have three options. Firstly you can make a single copy of line, column etc.. Secondly you can copy several lines or columns at once to a new area. Thirdly you can copy one line or column several times. Generally you will be asked if you want to adjust the references. This means that if you are copying a line, all references to other locations on that line will be changed to the destination line. This preserves the sense of the calculations along a line. The same applies to adjusting references down a column.

#### Copying formulae

If you want to print out the rules or formulae then use the EXCHANGE COMMAND first. Next adjust the column widths as required and finally copy the area in which you are interested to the printer.

#### Copying from files

You can copy a file from the current disk unit or other if specified, i.e. B:FILENAME.EXT. The file may be put into any blank area of the worksheet or be placed as an extension at the end of the sheet. The program will take care of any problems concerning different layouts between the worksheets. This command allows very flexible file merging.

The file to be read may either in the format of a .MEM file or be a .DAT file. In the latter case it should comprise just numerals in alphanumeric form. The program will check that the area to which the file is to be copied is vacant and if not the the command will not be implemented. A .DAT file will be first read to determine its structure and then read for a second time. On reading, values will be collected line by line and blanks will be taken to be spacers. Leading blanks on each line will be ignored.

.DAT file, contents ' 1.23'  
'2.34, 5.67'

will be read as: ---A--- ---B---  
1 1.23  
2 2.34 5.67

#### Copying to files

You can copy to files either in a way that a word processor can handle or in such a way that THE CRACKER can read it and reconstitute it exactly. There are two types of Text file the first with the extension .DAT will only copy numbers as they appear on the screen to the file. The second, with the extension .TXT will copy both the text and numbers as laid out on the screen. Both will copy any part of the worksheet. If you arrange your .DAT file correctly you should be able to read it from a BASIC or FORTRAN program if you want to do further processing. If you are copying to a file that already exists, then the original will be renamed with .BAK extension.

#### Preparing mail labels

This command will prepare printed mail labels. The addresses will usually be on single line and so you will have to insert markers to indicate where you require each next line to begin. Insert extra column at the end of each address line and give it a default carriage return format. Remember to put in a carriage return column at the end of the address. You will get confused results if you leave it out. You will need to adjust the column widths to align the addresses onto the tops of the labels.



## EDIT COMMAND

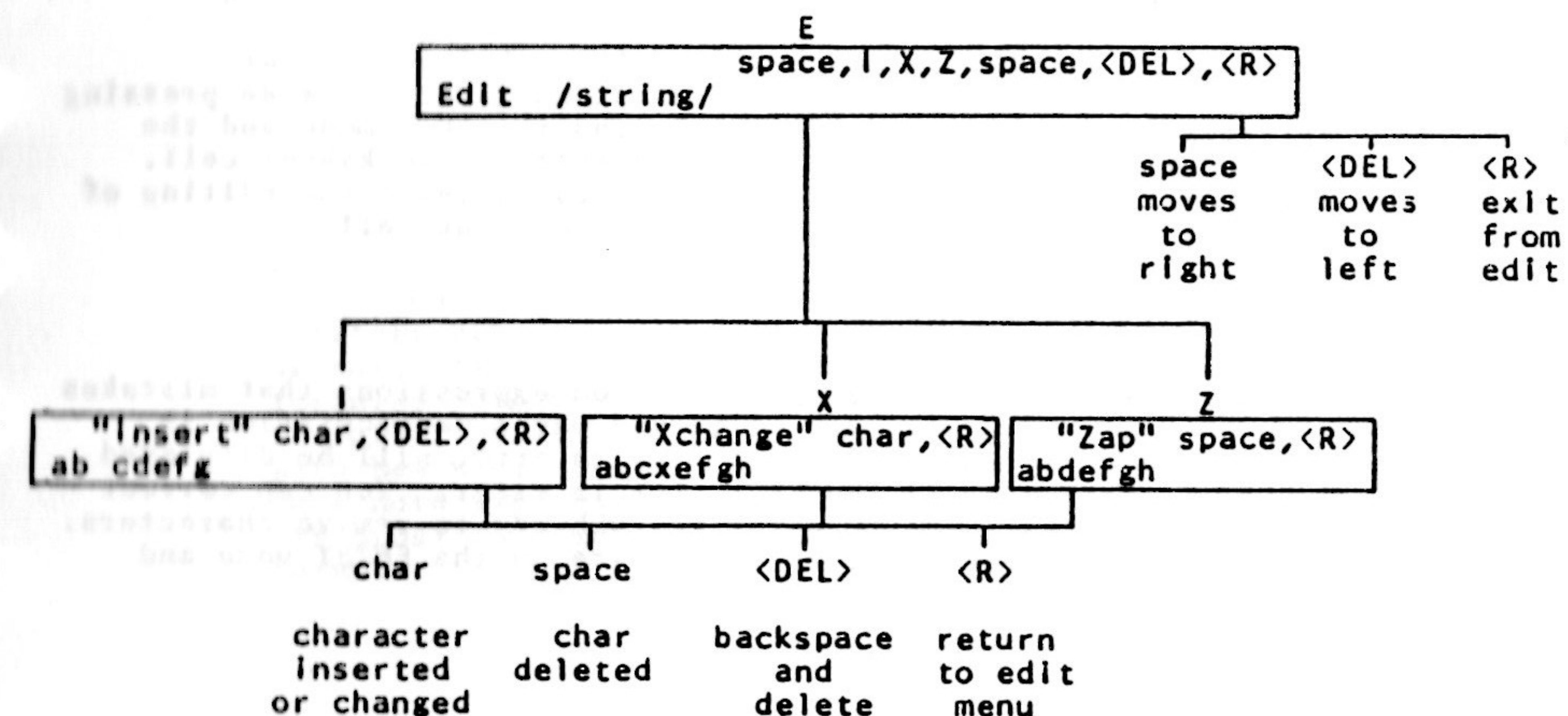
## CURSOR MOVEMENT

U, up arrow, CTRL-W upward movement.  
 D, down arrow, CTRL-J downward movement.  
 L, left arrow, CTRL-A left movement.  
 R, right arrow, CTRL-D right movement.

+ cursor moves down one page.  
 - cursor moves up one page.

A page is defined as the maximum number of worksheet lines that can be displayed on the screen.

If you use the arrows while entering text or expressions then it will finish off that entry and move the cursor to the cell in the direction you specified. This cell will be set up in entry mode. The use of arrows in this way can save much time with long lists as the one keystroke is equivalent to RETURN, cursor movement and . for entry. See also the JUMP COMMAND.



## Entering the EDIT mode

You can EDIT the contents of any cell. While editing the automatic calculation feature will be switched off for expressions. However on completion of the editing a recalculation will be carried out. You will not therefore know about any errors until you have left the EDIT mode. The EDIT command is entered by typing 'E' followed by a carriage return after which the relevant cell will be displayed on the edit line and the cue located at the first character.

## Moving the cue

To move the cue to the right just press the space bar once for each space. To move the cue to the left press the DEL key once for each movement.



**Insert:** If I is typed at any particular location, the entry will split about that point creating a blank space into which a new character can be inserted. A further space will then be created for the next character to be inserted. Pressing the return key will cause removal from the insertion mode.

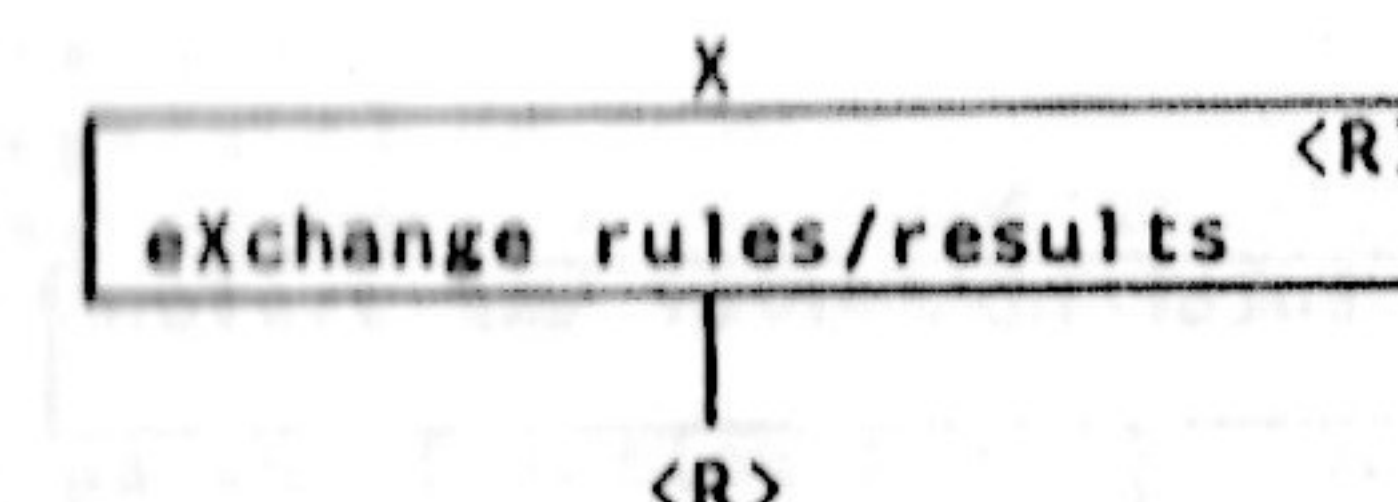
**eXchange:** Typing X will allow the character at the current cue location to be replaced by the next entered character. The RETURN key may be used to leave this mode and go back to the main edit menu.

**Zap:** Typing Z at any cue location will delete that particular character. While not in insertion, eXchange or zap mode pressing carriage return will cause removal from the EDIT mode and the amended line will be placed into the relevant worksheet cell. Alternatively the arrow keys may be used to leave the editing of the cell and go into entry mode on an adjacent cell.

**Caution**

Be wary when using the edit facility on expressions that mistakes made in your logic will not be picked until recalculation is carried out. At this point the line in error will be displayed on the edit line up to the character in error. You can correct the line at this stage by using the DEL key to remove characters. Then make insertions just as if you are in the ENTRY mode and finish off with a carriage return.

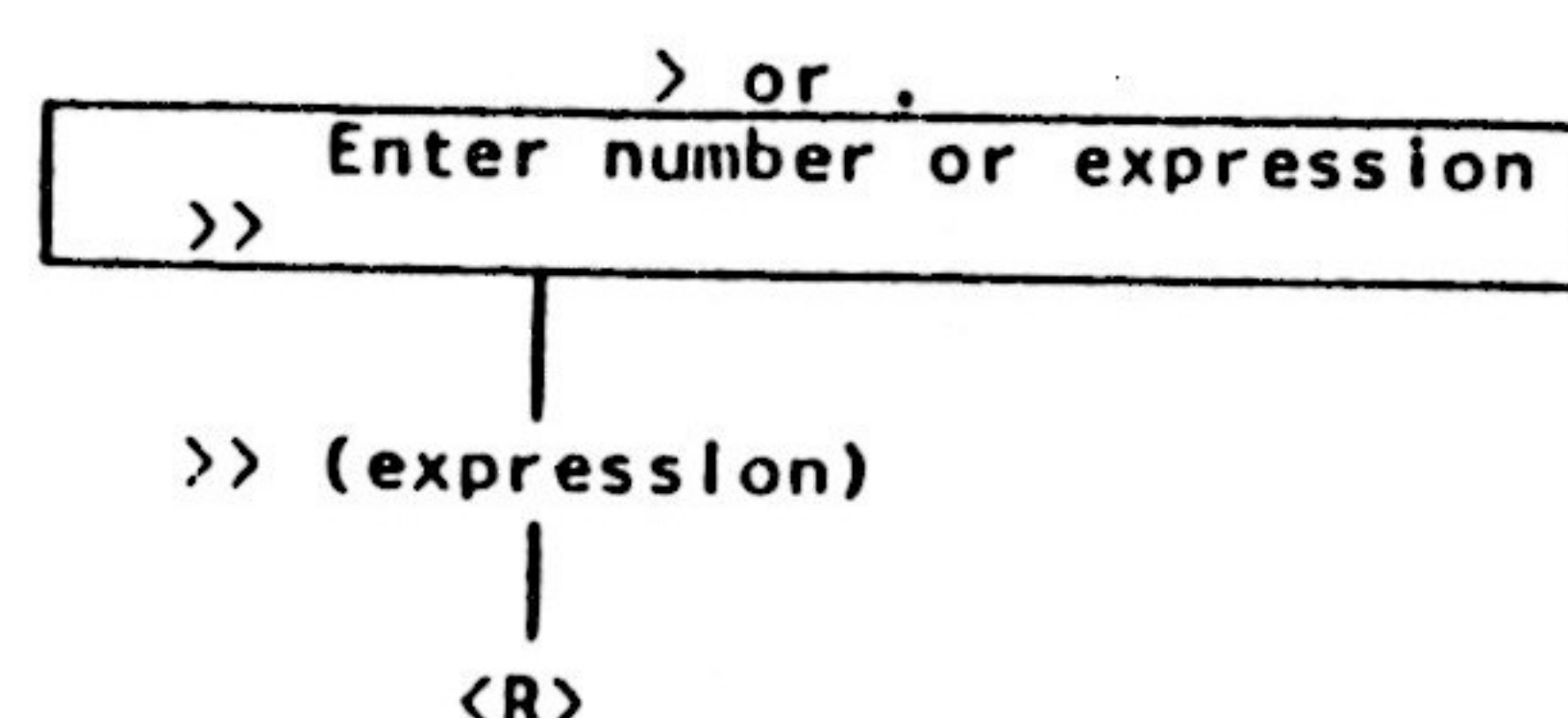
**EXCHANGE COMMAND**



The EXCHANGE command is initiated by typing in an X followed by a carriage return. The effect of this command is to change the default display in the cells from the numerical results to the formulae from which they are calculated. If they are to be seen in full it will usually be necessary to amend the column width with New Width command sequence. The command is cancelled by once more typing an X. Each expression cell is displayed with its order of calculation number in brackets followed by its formula.



## EXPRESSION ENTRY



The expression is collected on the edit line and on completion is inserted into the memory together with an order of calculation number. On completion the memory remaining is calculated and displayed and if automatic calculation is switched on the worksheet is recalculated. Expressions will be displayed inside brackets on the cell contents line.

## Evaluation of expressions

Expressions may contain functions, coordinate references, numbers, arithmetic operators, logical operators and special purpose operators. Expressions are evaluated according to the rules of normal algebra.

All operators are given an importance which determines the order in which the various parts of the expression are calculated. This order of importance is over-ridden by parentheses, each set of which is evaluated as if it is a self-contained expression. The innermost set of parentheses is calculated first and then the next innermost and so on working outwards.

A multiply symbol must be put between two sets of parentheses. You can not have (2)(3) for  $2 \times 3$  instead you must put (2)\*(3).

The operators within the expression or set of parentheses are calculated in the following order: 'plus and minus', 'percentage', 'exponentiation', 'multiplication and division', 'addition and subtraction', 'greater than' and 'less than' and 'equal', 'greater than or equal' and 'less than or equal' and 'not'.

If an exclamation mark is encountered then the value of the expression to that point is determined and replaces the whole of the expression on the edit line. It should be used with caution as indeterminate results may occur if for example it is used from within a function.

The valid parts of an expression are as follows: operators, coordinates, defined functions, numbers and conditionals. Functions comprise a named operation with its required parameters in brackets and where there are more than one these parameters must be separated by commas. There are three types of function, those whose arguments have multiple parameters, those which have single parameters and those which have no parameters at all.

When an apparent function name is encountered it is checked against the list of inbuilt functions. Functions may call other functions in their parameters. Indeed a function may even call itself amongst the parameters. As this is a stacked operation the number of levels of nesting is limited, however the program keeps a track of the state of the stack. If the nesting has become too great an error message will come up and it will only be necessary to rearrange the expression so that it is placed into two entries rather than one. The expression so far is not lost, it is merely necessary to use the DEL key to backspace.

Expressions are evaluated as you enter them, not on completion. This means that you have dynamic error checking character by character. If you enter a wrong character you will get an immediate error message. To correct an error, backspace with the DEL key and then enter a new character.

If you are entering a complex expression such as a DO - WHILE formula then the calculation character by character can get bogged down. In this case first switch off the automatic calculation. See the AUTOMATIC CALCULATION COMMAND.



# ARITHMETIC OPERATORS

+ Addition  
- Subtraction  
\* Multiplication  
/ Division  
^ Exponentiation (raising to a power)  
% Percentage

# LOGICAL OPERATORS

Logical operators act on arithmetic values or expressions (A and B below) and determine whether they comply with specified conditions.

|   |                       |              |
|---|-----------------------|--------------|
| = | Equal                 | e.g. (A = B) |
| > | Greater than          | e.g. (A > B) |
| < | Less than             | e.g. (A < B) |
| ] | Greater than or equal | e.g. (A ] B) |
| [ | Less than or equal    | e.g. (A [ B) |
| ! | Not equal             | e.g. (A ! B) |

If the expression in brackets is met then the result is TRUE and given the value -1, if it is not met then it is FALSE and given the value 0.

(3.4 ] 1.23) = TRUE  
(3.4 < 1.23) = FALSE

# BUILT IN FUNCTIONS

## TRIGONOMETRIC FUNCTIONS

SIN(X)  
COS(X)  
TAN(X)

Determine the sine, cosine and tangent respectively of X, where X is in degrees.

SINR(X)  
COSR(X)  
TANR(X)

Calculate the sine, cosine and tangent respectively of X, where X is in radians.

There is no restriction on the size of X as 2\*PI (or 360 degs) will be repeatedly subtracted until X is within range. X may be negative.

ASIN(Y)  
ACOS(Y)  
ATAN(Y)

These functions determine the angles whose sine, cosine or tangent respectively is given by Y. The result is given in degrees in the range 90 degs to -90 degs.

ASINR(Y)  
ACOSR(Y)  
ATANR(Y)

These functions determine the angles whose sine, cosine or tangent respectively is given by Y. The result is given in radians in the range PI/2 to -PI/2.

Y must be less than or equal to 1.



# LOGICAL FUNCTIONS

Logical functions analyse a list of logical values or expressions (X,Y,Z.... below) and return a value of TRUE or FALSE as defined above. In practice each expression or value is tested to see if it is -1 or 0. If it is -1 it is taken as TRUE if it is 0 it is taken as FALSE.

## AND(X,Y,Z....)

X,Y,Z.... are in turn checked for truth. If they are all TRUE then the function returns a value of TRUE (-1).

AND(TRUE,TRUE,TRUE) = TRUE  
AND(TRUE,FALSE,TRUE) = FALSE as list not all TRUE

## OR(X,Y,Z....)

X,Y,Z.... are in turn checked for truth. If any of them are TRUE then the function returns a value of TRUE (-1).

OR(TRUE,FALSE,FALSE) = TRUE as one item of list is TRUE  
OR(FALSE,FALSE,FALSE) = FALSE as none of list is TRUE

## NOT(X)

The truth of X is checked and the opposite is returned as the function value. If the value of X is TRUE then FALSE (0) is returned. If the value is anything other than TRUE then TRUE (-1) is returned.

NOT(TRUE) = FALSE  
NOT(FALSE) = TRUE

# UTILITY FUNCTIONS

## ABS(X)

The absolute value of X is returned. It is defined as the numerical value of X with a positive sign.

ABS(-2.345) = 2.345  
ABS(2.345) = 2.345

## AVERAGE(list)

The average of the non blank values in the list is calculated.

AVERAGE(3,4,5) = 4

## CHOOSE(N,list)

The nearest integer value to N is found and the Nth item in the list is returned.

CHOOSE(3,5,6,7,8,9) = 7

## COMB(N,R)

The number of ways of combining R items from a total of N is calculated.

## COUNT(list)

The list is checked for the number of values that are non blank.

COUNT(2,3,4,B5) = 3 ; cell B5 is blank

## CRD(X,Y)

Can be used in any location where a coordinate is expected. The two arguments are the column and line. The column must however be expressed in a numerical form (A=1, Z=26 etc.). This function is useful in filling up tables from DO iterations.

## DEC(X)

Returns the value of X minus 1. This is also intended for iterations.

DEC(2.345) = 1.345

## DO(range)

Facilitates subroutines, looping and iteration on the specified range. If iteration or looping is required the line will need to be terminated with a WHILE function.

## DPART(X)

Takes the value of the decimal part of X, that is the part after the decimal point. Be warned that using this function loses one significant figure for each figure that was previously before the decimal point. If you get unexpected results consider this.

DPART(5.78) = 0.78  
DPART(-3.45) = -0.45

## e

Gives the exponential constant 'e'

e = 2.718281828459045

## ERROR

A special function that if encountered in a calculation will bring up a message that an error has been called from that cell.



Usually used to check that values entered are in a permissible range.

EXP(X) Raises the constant 'e' to the power in the bracket following. A check is made to ensure that the exponent is not too great.

EXP(3.14159265358979) = 23.14069263277927 exp(pi)

FACT(X) Works out the factorial of X.

FACT(3) = 6

FALSE Takes a value of FALSE, that is 0.

GROW(%,N) Will increase a value N by a percentage. The percentage can be negative.

GROW(5,200) = 210

IF(logic expr),THEN(expr 1),ELSE(expr 2)  
This group is used to build decision making into the worksheet. The logical expression is analysed and if TRUE the THEN expression 1 is used to get the value for the cell. If the logical expression is FALSE then the ELSE expression 2 is used instead.

INC(X) Returns the value of X plus 1. It is general purpose but is included for convenience in iterations.

INC(2.345) = 3.345

INIT(crd,val)  
Will initialise a cell with a specified value. The cell entry with this function will have a low recalculation number. The function is used for setting up loops and iterations. It is very similar to the SET function.

INT(X) Takes the value of the integer part of X, that is the part before the decimal point, with the sign retained.

INT(5.78) = 5  
INT(-3.45) = -3

INTERP(N,range)  
Compares N with each value in the range to find the two values between which N lies. The two values from the adjacent line or column are then interpolated to give the final value. The interpolation is done by taking the proportionate distance than N lies between the first two values and applying it to the adjacent values.

LOG10(X) Determines the logarithm to the base 10 of the value in brackets.

LOG10(2.71828182845905) = 0.43429448190325 log10(e)

LN(X) Determines the natural logarithm to the base 'e' of the value in brackets. The routine will check for negative or zero arguments which are not allowable.

LN(10) = 2.30258509299405 loge(10)

LOOKUP(N,range)  
N is compared with each value in the range to find the first one it is greater than. The value from the adjacent line or column is then returned.

MAX(list) The list is scanned and the maximum non blank value returned.

MAX(2,7,3) = 7

MIN(list) The minimum non blank value in the list is returned.

MIN(2,7,3) = 2

NINT(X) The nearest integer value to X is found.

NINT(2.23) = 2  
NINT(5.67) = 6  
NINT(-3.45) = -3

NPV(I,list) Calculates the present value of the cash flow list using the discount % as specified.

PERM(N,R) Finds the number of ways of permuting R items from a total of N.

PI Returns the value of the constant PI

PI = 3.141592653589793

POS(X) Gives the value of the argument if it is positive. If it is negative then it returns a value of 0.

POS(2.345) = 2.345  
POS(-2.345) = 0

RND A true integer random number between 0 and 127 is found. This number is taken from the 280 refresh counter.

RND = 23  
RND = 120  
RND = 3

ROUND(X) Takes the value of X rounded to two decimal places. It is of use in financial calculations to avoid cumulative errors caused by including fractions of a penny (cent etc.). The rounded value is



not just displayed it is also the value stored in the internal memory.

ROUND(1357.5679) = 1357.57

SET(crd,value)

Will set a cell to a particular value. The cell must be either blank or a constant. This command is usually used with the DO function to fill up a table with values.

SIMPRULE(step,range)

Works out the numerical integration of the range by Simpsons rule. The first parameter is the step length. Range holds the values for integration. The range must be an odd number of values consecutively. The latter parts of the range can be blank and these will be ignored. This makes flexibility in setting up worksheets possible.

SQRT(X)

Determines the square root of X.

SQRT(16) = 4

STDEV(l1st)

Looks through the list and works out the standard deviation.

SUM(l1st)

Will add up the non blank values of the list and will return the total.

SUM(2,3,4) = 9  
SUM(-3,4,5) = 6

TRUE

Takes a value of TRUE, that is -1.

WHILE(expr)

Must be used together with a DO function. The expression must be logical and if TRUE the line will be recalculated from the start of the DO on that line. If FALSE the WHILE will do nothing.

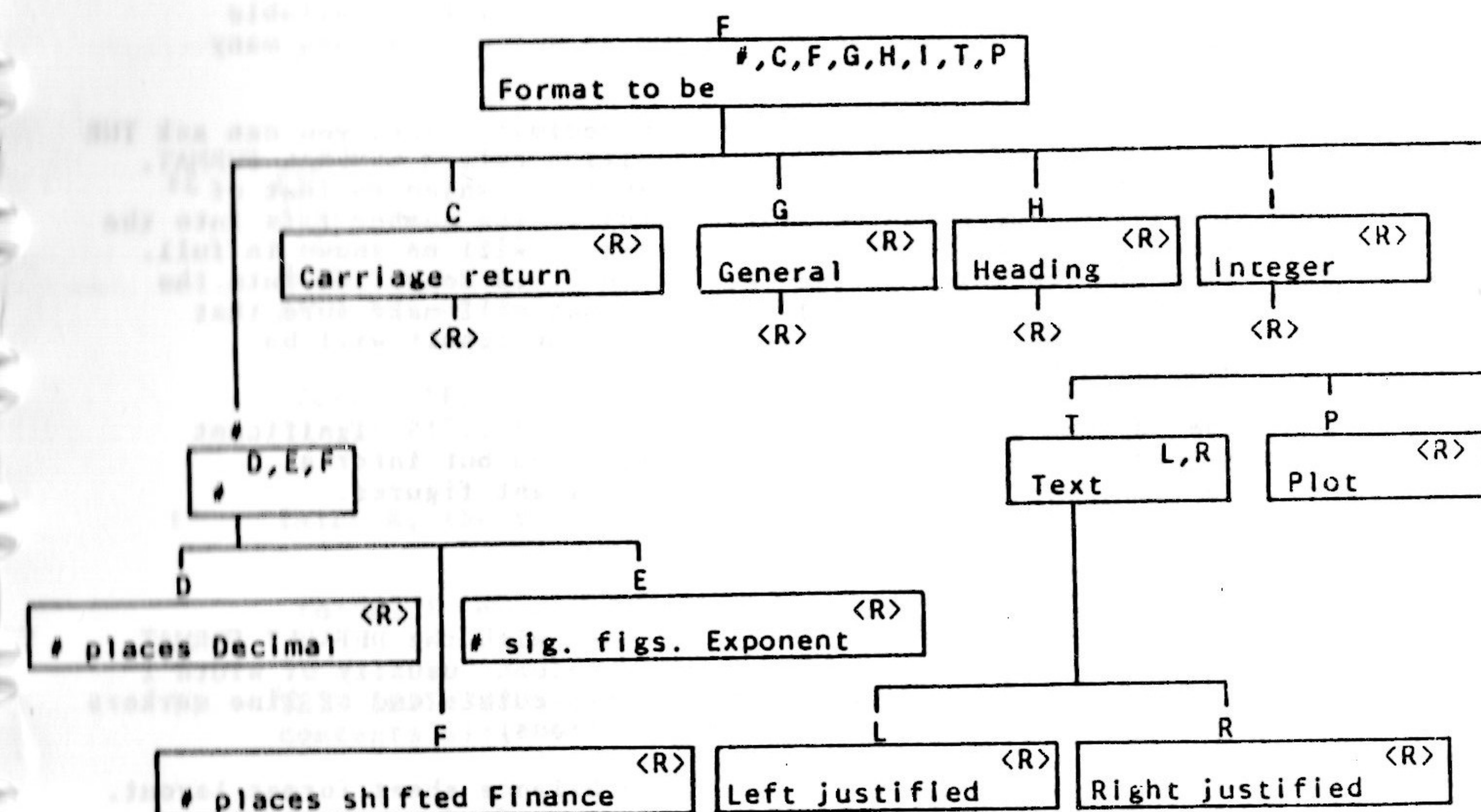
VAL(X,Y)

Gives the numerical value of the cell specified by the two parameters. The coordinates are defined in the same way as for the CRD function.

VAR(l1st)

Looks through the list and works out the variance. The variance is the standard deviation squared.

FORMAT COMMAND



The FORMAT command is used to specify the way in which a cell is to be displayed on the screen. There are two types of FORMAT command. The first is the DEFAULT FORMAT, which specifies the format in which an entire column will be displayed. The second defines the FORMAT for an individual cell. When the FORMAT for a cell has been defined it will have precedence over the DEFAULT FORMAT.

The command can have no effect on the way an entry is stored in memory. Once specified the format will stay with an entry even if moved to a new position. This is because it is an entry that has a format and not a cell location. If no format is specified



then an entry will automatically pick up the default format for that column.

Internally the numbers are stored in a binary floating point format. It is not important to understand the details of the internal floating point number merely to know that it will ensure accurate calculations with up to 15 significant figures. Generally you would not want to display all the available accuracy so the program allows you to define just how many decimal places you want to show.

If you do not really know how many decimal places you can ask THE CRACKER for the vague format definition called GENERAL FORMAT. It will display the number in a similar fashion to that of scientific pocket calculators. Thus if the number fits into the space available in its entirety then it will be shown in full. If it is too large or too small then it is converted into the EXPONENT form (see below). This format will make sure that whatever the result of a calculation the result will be displayable.

The maximum number of decimal places is 38. 15 significant figures is the most that can be displayed but internal calculations are done to 16-17 significant figures.

Format types

C CARRIAGE RETURN, this can only be used with the DEFAULT FORMAT. It should only be used with a blank column, usually of width 1 space. Its purpose is to put in intermediate end of line markers in mailing list and database applications.

#F FINANCE, this outputs all values in balance sheet format layout. The # represents the number of places shifted you want the results. This is the same as saying the number of times you want the value dividing by 10 before displaying it. With this facility you can display your answer in thousands or millions. Commas are added every third place in the usual way of presenting financial output. Negative values are enclosed in brackets.

```
format OF, memory 12345.678, display 12,345.68
" OF, " -12345.678, " (12,345.68)
" 3F, " 12345.678, " 12.3
```

G GENERAL, this is a general purpose format that will display the value as simply as possible and as nearly in the way that it might be displayed on pencil and paper calculations. Unnecessary zeroes will be removed and if the number is too large or small it will convert to exponent format. The format bears some resemblance to that used on scientific calculators.

```
format G, memory 123.0, display 123
" G, " 0.123, " 0.123
" G, " 0.000000123456, " 0.123456E-06
```

#D DECIMAL, the # defines the number of decimal places that will be displayed.

```
format 5D, memory 123.456789, display 123.45679
" 2D, " 123.456789, " 123.46
```

#E EXPONENT, the # defines the number of significant figures to be displayed in the decimal part. The exponent form is sometimes called scientific notation. It consists of a decimal number followed by the power of 10 by which it must be multiplied to give the actual number. This power is called the exponent.

```
format 5E, memory 123.4567, display 0.12346E+03
" 3E, " 0.001234, " 0.123E-02
```

I INTEGER, the value is displayed to the nearest whole number

```
format I, memory 567.89, display 568
```

Tl TEXT LEFT JUSTIFIED, is for alphanumeric entries. The memory contents will be displayed left justified in the cell.

```
format Tl, memory 'TEST', display TEST
```

Tr TEXT RIGHT JUSTIFIED, is for alphanumeric entries. The memory contents will be displayed right justified in the cell. If the cell is smaller than the entry the display will be filled as if it was going to be left justified.

```
format Tr, memory 'TEST', display TEST
" Tr, " 'THIS IS A TEST', " THIS IS A T
```

H HEADING, is for alphanumeric entries. the memory contents will be displayed in their entirety even if the column is not as wide as the entry. This format can therefore be used for titles when the column width is, at different times, liable to be varied.



Adjacent entries will not be displayed if the heading over-rides them.

format H, memory 'THIS IS A TITLE'

    ---A-- ---B-----  
display THIS IS A TITLE

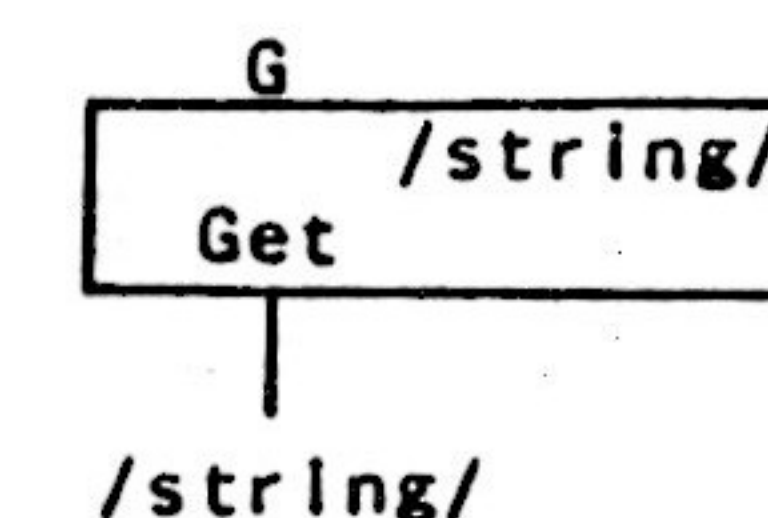
P PLOT, will fill up a cell with asterisks to the nearest integer value of the stored number.

format P, memory 5.556, display     ----A----  
                                                \*\*\*\*\*

Overflow: If there is insufficient space to display a numerical value then the cell will be filled with asterisks.

format 4D, memory 1234.56789, display     ---A---  
                                                \*\*\*\*\*

GET COMMAND



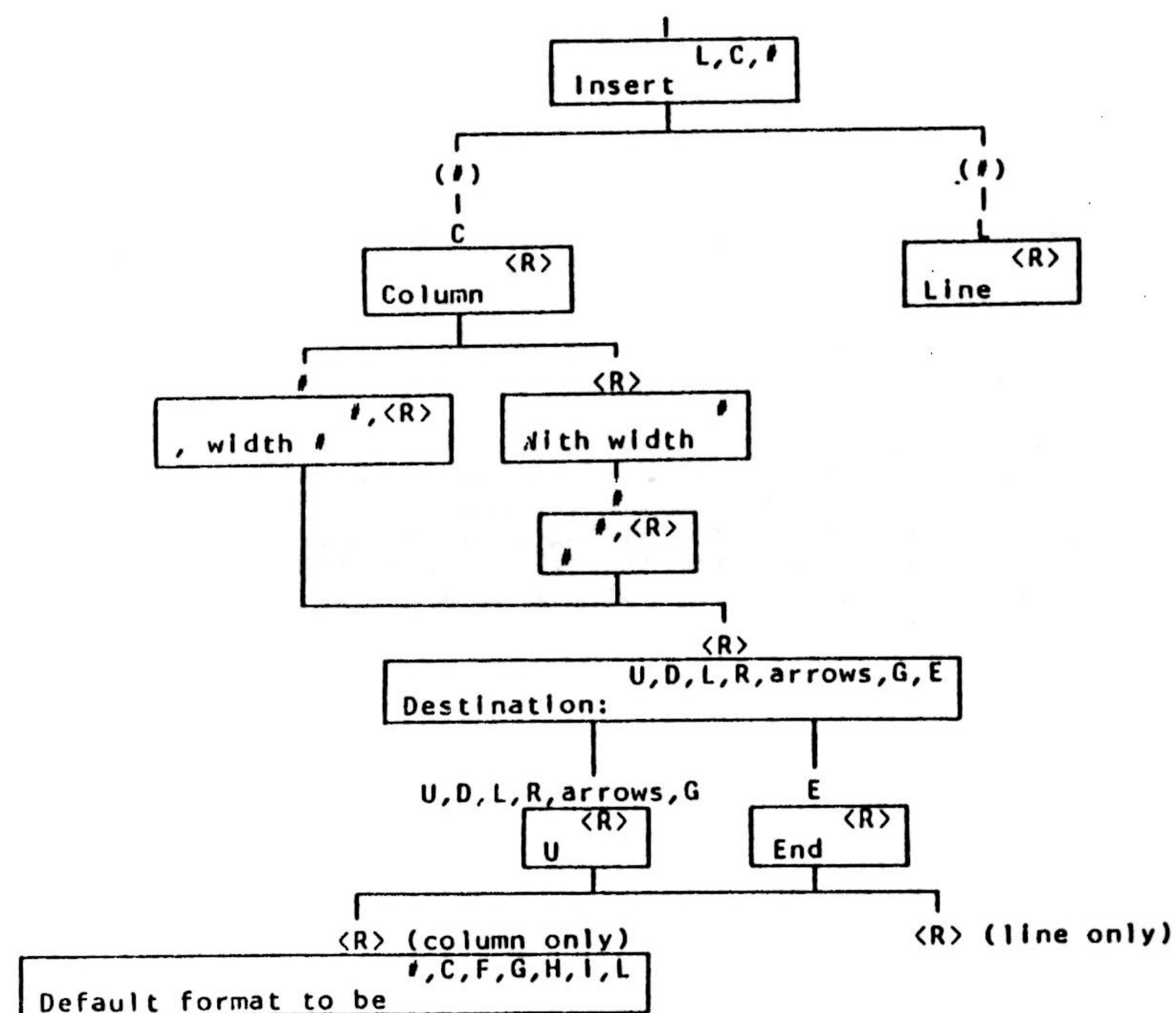
The GET command is used to search from the current cell location to the end of the worksheet for a particular string. The string is delineated by a single printable character which must not be a letter or numeral. When this delineator is next encountered the string is assumed to be complete. The string length may be up to 30 characters not including the delineators.

Only the text or expression as it appears on the contents line is searched. The search proceeds across each column and then down a line and so on. If repeated searching is to be used then after initially setting up the string it is merely necessary to put two delineators in one after the other. The string that was previously used will be redisplayed and form the basis of the following search.



## INSERT COMMAND

insert lines. You can insert several columns or lines at a time. Increasing the size of the sheet does not use up any of your available memory.

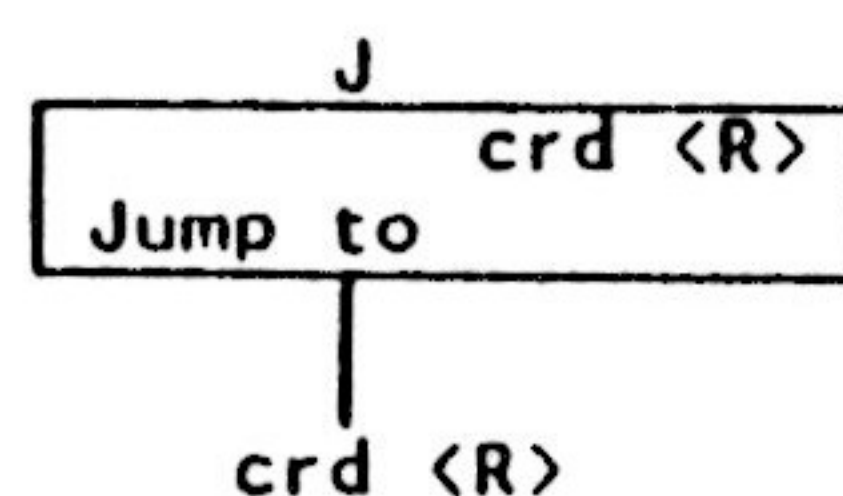


format definition in  
the same way as the  
FORMAT command.

The INSERT command is used to increase the size of the worksheet. The sheet does not initially exist and so you must start by defining at least one column. With a column inserted you can

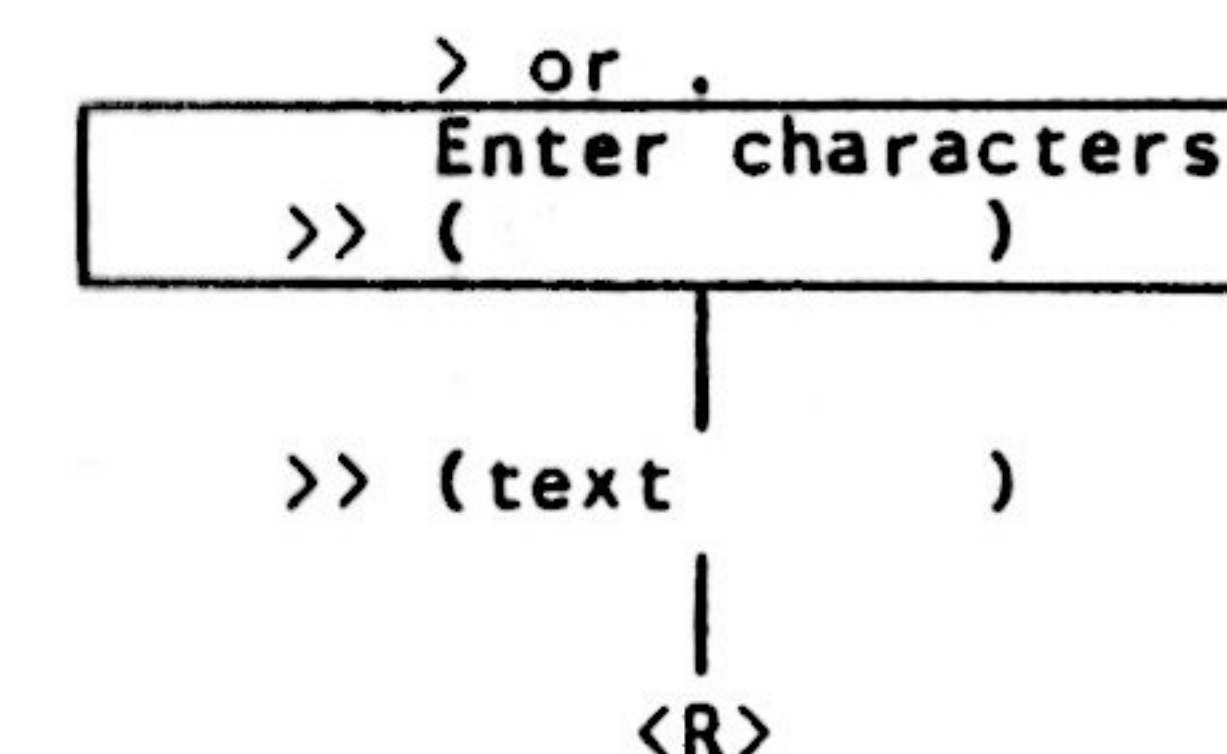


## JUMP COMMAND



The jump to command is used to move the cursor directly to the cell you specify. This command can be used both as a primary command and in answer to the request for destination.

## TEXT ENTRY



If the default format is for TEXT or if the format has been explicitly set as TEXT then on typing '.' or > new TEXT ENTRY mode can be entered. The edit line will be marked to show the width of the current column. It does not in fact restrict the number of characters that can be entered at that location.

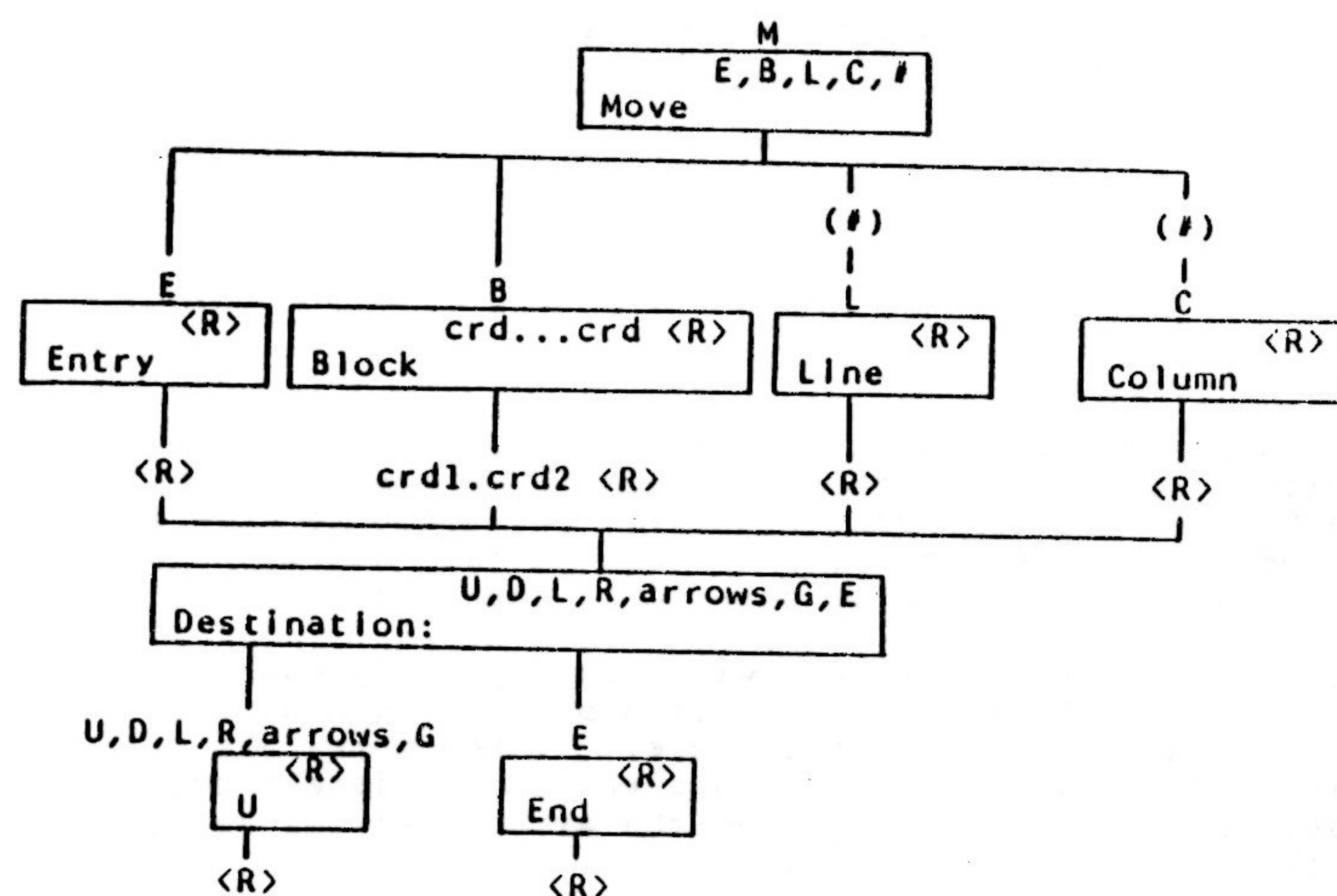
The prompt 'enter characters' will come up. Any number or character may be inserted and if a mistake is made the DEL (RUB OUT) key may be used to backspace. If the entry has been completed and it is desired to move to an adjacent cell and insert information there, then the arrow keys may be used instead of the carriage return key.

Pressing an arrow key once will move the cursor into the relevant adjacent cell and the program will be set up for entries at that location without any further key pressing. The one key movement therefore by-passes the carriage return, cursor movement and the > for ENTRY mode. It is for this reason that the backspace key cannot be used while entering text or expressions because the backspace key and left pointing arrow often give out the same command to the computer.

On completion of text entry it is inserted into the cell location. The memory remaining display is then adjusted.



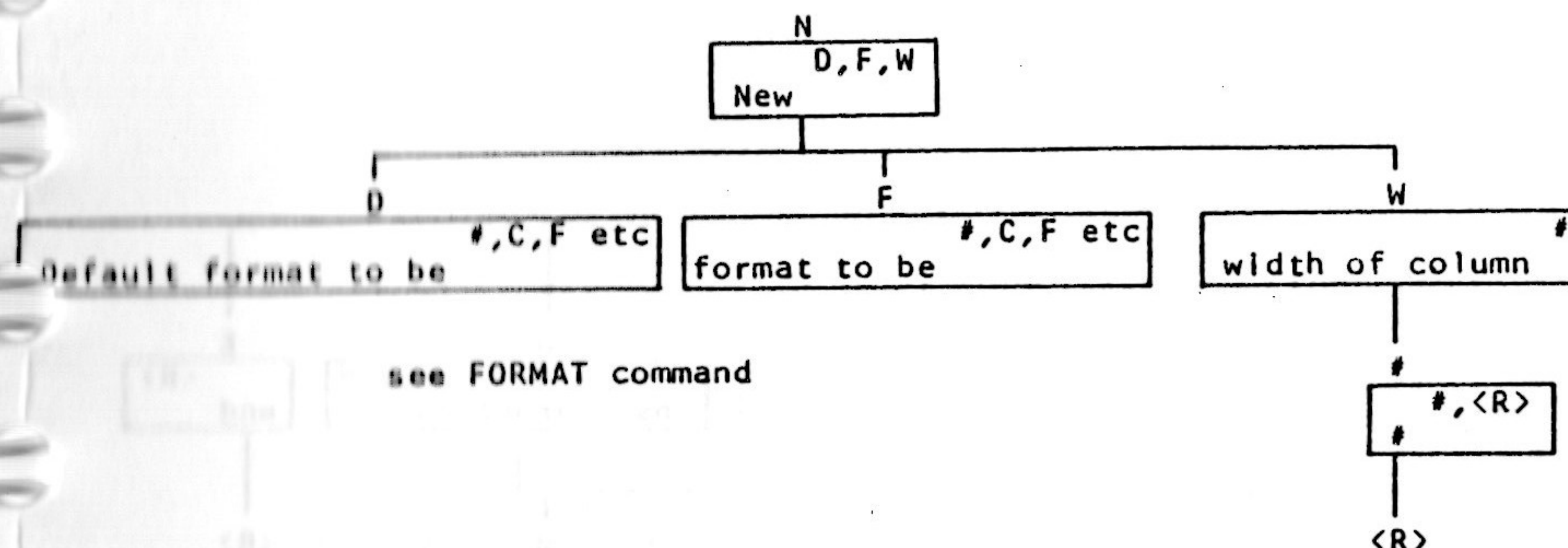
## MOVE COMMAND



The MOVE command transfers a section of memory to a new destination. Before the movement takes place a check is made to see that the destination area is vacant and if not the command is not carried out. All references to the transferred area will be automatically adjusted to the new location.

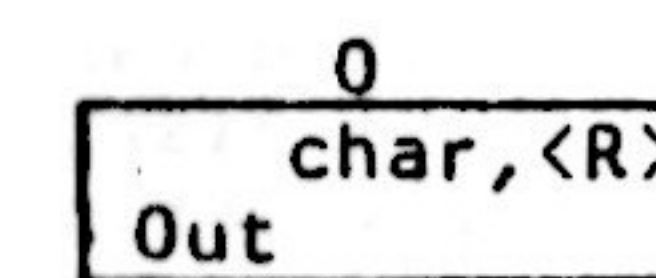
If several lines or columns are being moved an internal check is made that the destination specified will provide sufficient room.

## NEW COMMAND



The NEW command defines a change of either width of column, explicit format or default format. A new format can only be a change between compatible types for example a change cannot be made between a TEXT and a FINANCE format. This is because one is character information and the other is numerical information. A change in default format will only affect future entries that do not have their format explicitly defined.

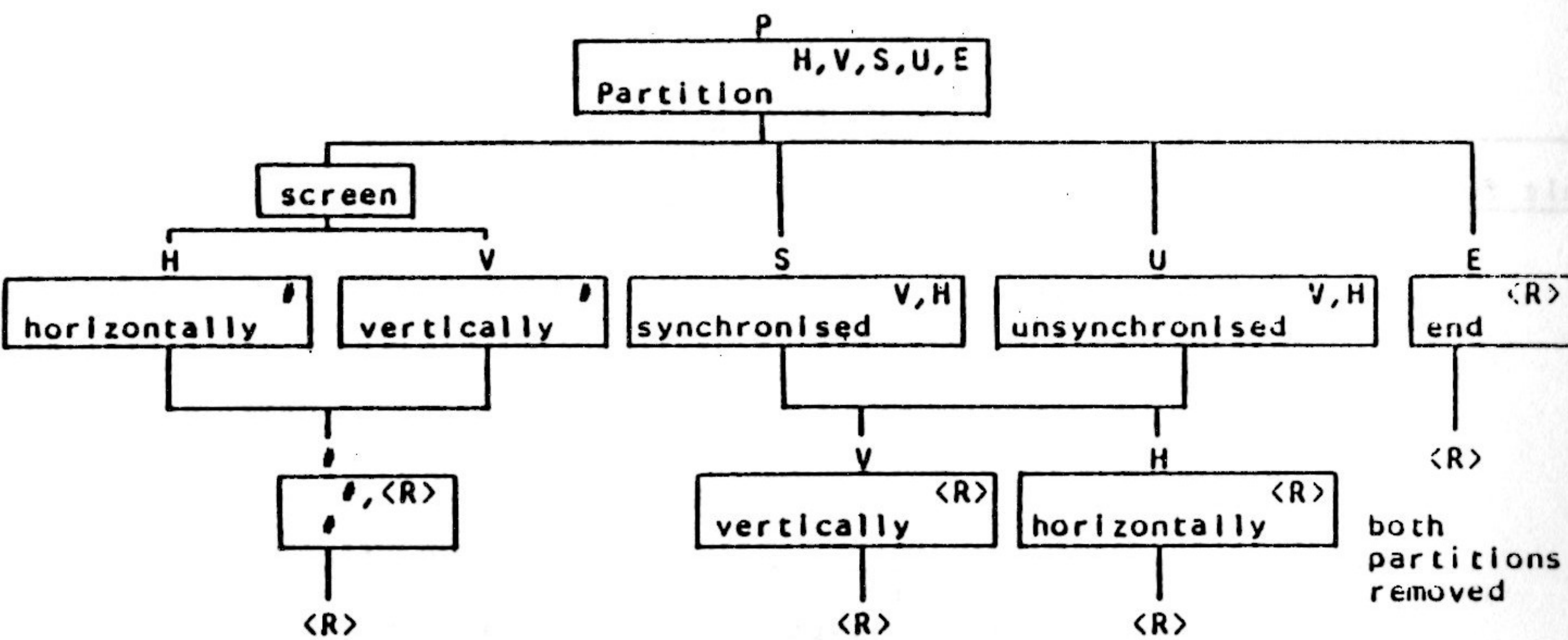
## OUT COMMAND



The OUT command is way of outputting commands directly to the printer. With this command you can set up the printer for special features such as condensed print or a different character font. All characters including control codes are valid. To finish the command use <R>. If you make a mistake finish off that entry with a <R> and start it again.



PARTITION COMMAND

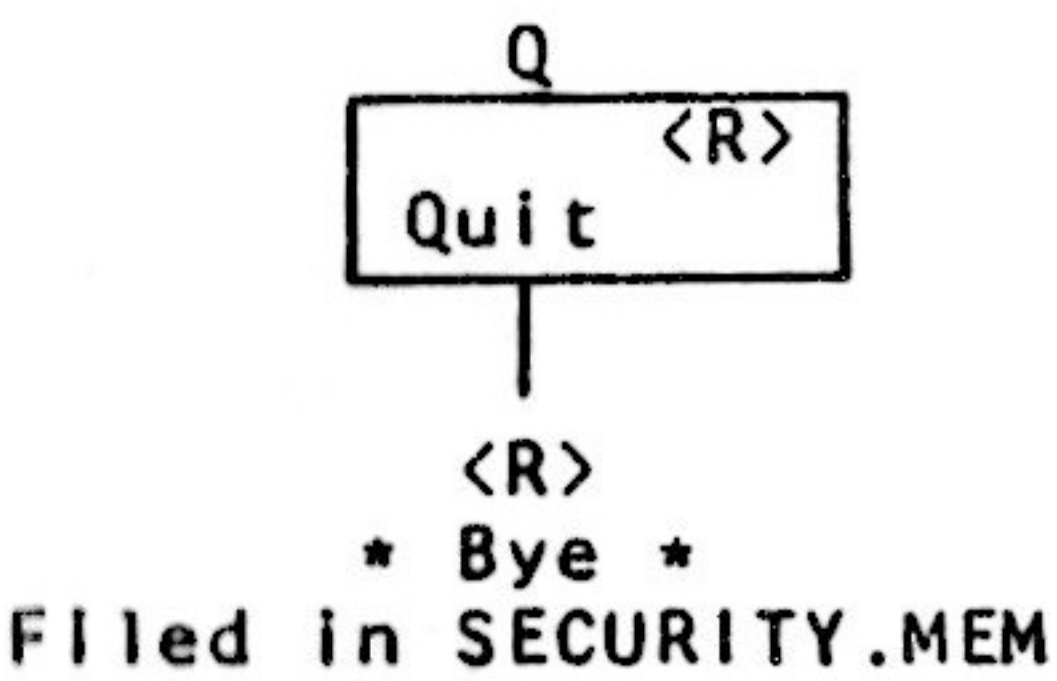


The screen may be partitioned either vertically or horizontally to give 1, 2 or 4 available windows and the movements of the cursor locations within these windows may be defined as synchronised or unsynchronised both for the horizontal and the vertical partition. Synchronising means that movements in one window will be matched by equivalent movements in the other. Without it the inactive window will be unchanging.

The horizontal partition location can be specified by taking a number from a grid that will be displayed on the screen. The vertical partition is similarly defined. Partitions may be removed by the PE partition end command sequence.

To jump the cursor from one window to the other the slash commands are used. / will jump left to right and vice versa and the \ will jump up and down between windows.

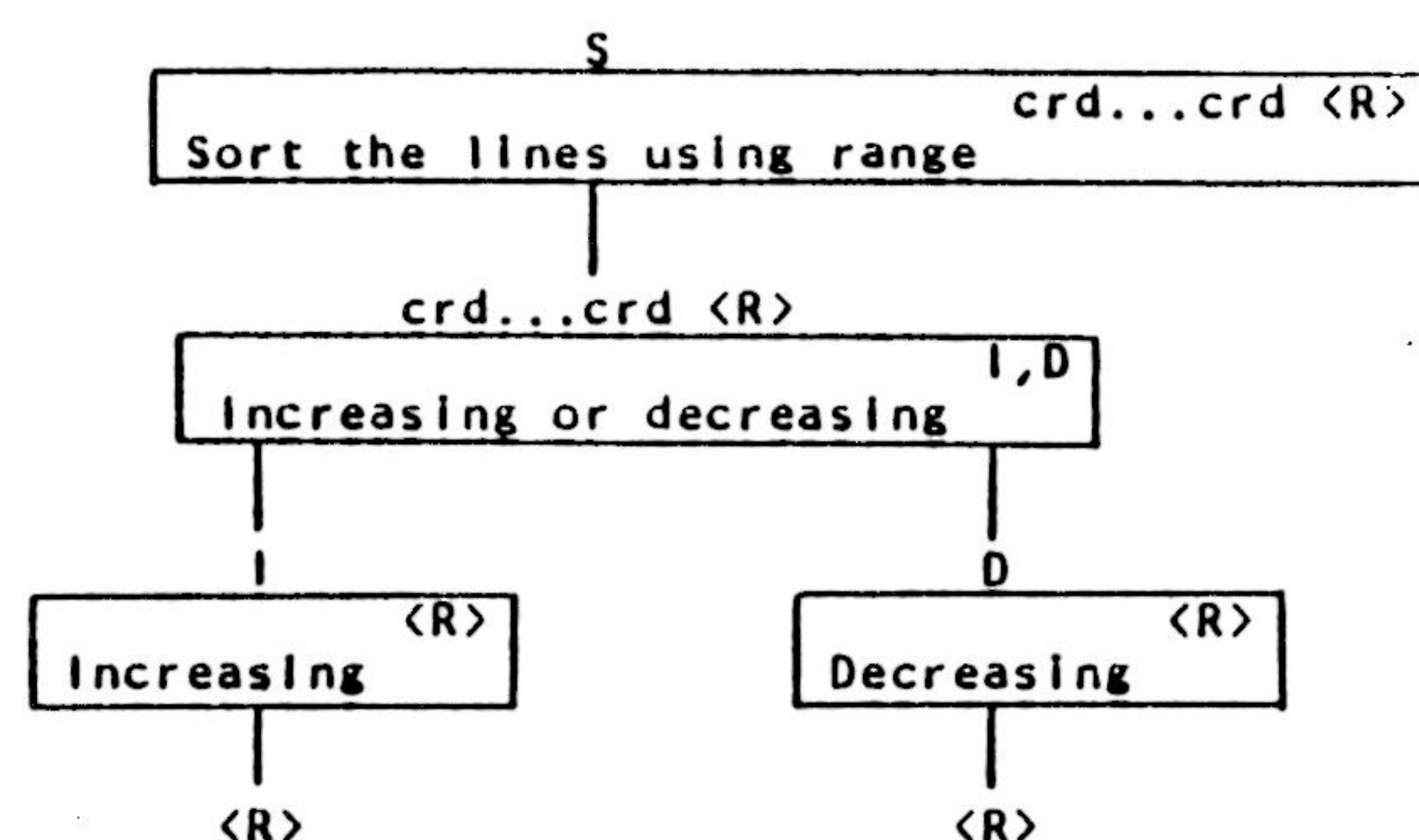
QUIT COMMAND



The QUIT clears the screen and makes a return to CP/M. A copy of your worksheet is always saved on file SECURITY.MEM. Experience shows that without this safety precaution it would not be long before you lose valuable work by mistake.



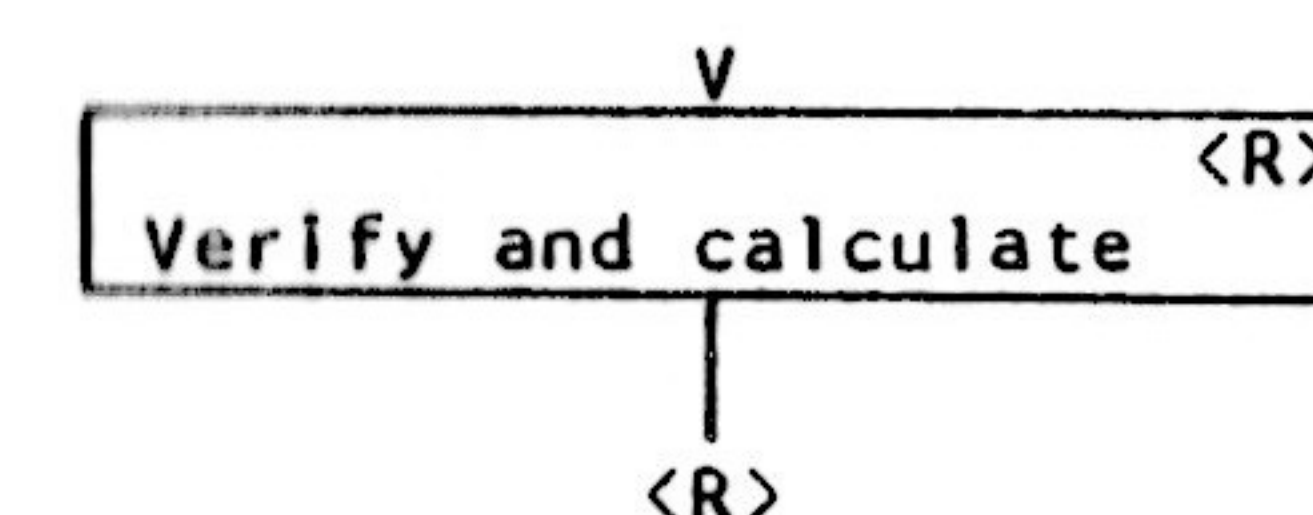
## SORT COMMAND



The SORT command allows sorting of lines using any specified part of any particular column as the key. The sort may be by increasing or decreasing values and may be carried out on both numerical values and characters.

Note that mixing of text and numerical values in the sort should be avoided as this will give indeterminate results. Capital letters and lower case letters are taken to be of equal value. With text sorts numbers come after letters followed by, blanks, punctuation and finally control codes. All punctuation is taken to be of equal value.

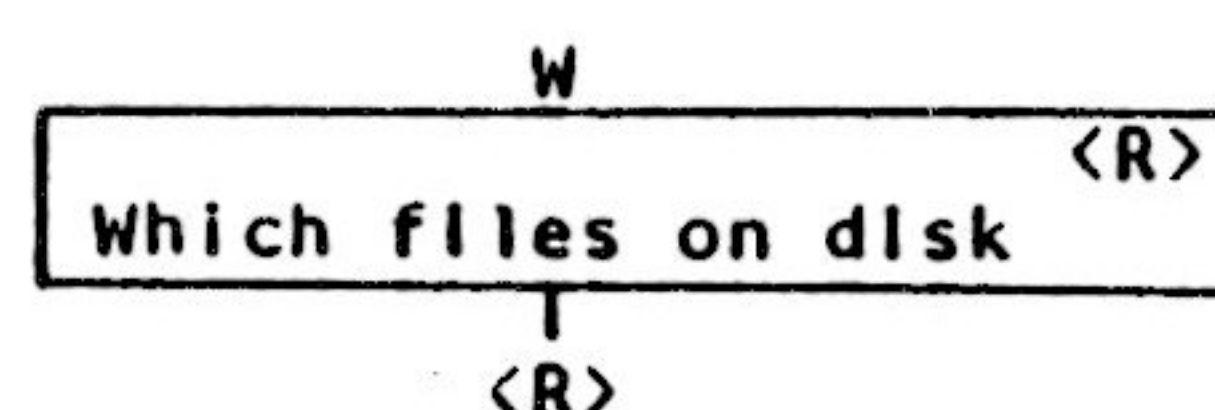
## VERIFY AND CALCULATE COMMAND



This command is initiated by typing V followed by a carriage return. The whole worksheet will be cleared, recalculated and displayed.

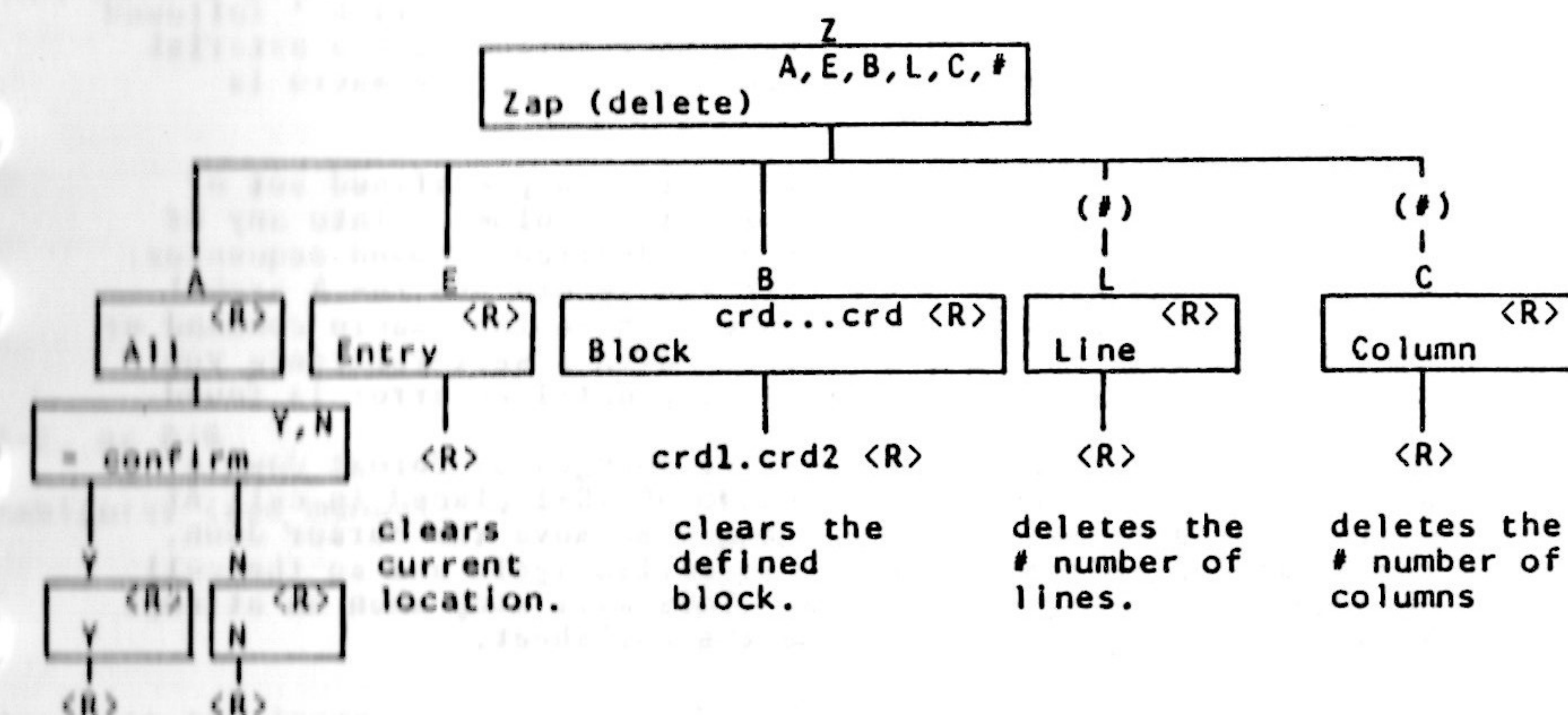


## WHICH FILES COMMAND



The WHICH FILES command is initiated by typing a W followed by a carriage return. The CP/M file directory will be read and all the compatible files, i.e. .MEM, .DAT and .TXT on the disk will be put up on the screen. When the information has been assimilated the carriage return may be depressed and the original screen of information will be returned.

## ZAP (delete) COMMAND



ZAP deletes the defined entries from memory. If the command refers to an entry or a block then the command acts just like the BLANK command. For references to line, column or all then the size worksheet is reduced by the command. Deleted entries cannot be retrieved so if in doubt first copy the memory to a file before embarking on complex rearrangements.

Before the deleting is carried out the program will check whether any of the items to be deleted are referred to elsewhere in the worksheet. If cross-references are found the command will not be carried out.



# • MACRO COMMAND

The \* MACRO command is initiated by typing an asterisk \* followed by a number between 1 and 9. The number following the asterisk refers to the line number of column A in which the macro is stored.

The purpose of this command is to call in a predefined set of commands. Start by inserting a new first column. Into any of the first 9 lines you can insert your desired command sequences. Where you would want to put a carriage return use the @ symbol. It is valid to finish your sequence with another macro command or even a reference to the same macro. With a self-reference you can create macros that go on repeating until an error is found.

This command is ideal for repetitive changes of format down a whole column for example. The macro NFG@D\*1 placed in cell A1 will change a format to GENERAL and then move the cursor down. The \*1 at the end means that it is called again and so the cell underneath is changed and so on. This will stop when an attempt is made to change a cell outside the worksheet.

## THE CRACKER MORE DETAIL ON ERROR MESSAGES

A-Z, or 0-9

Ambiguity (see manual)

Argument too large

Argument > 1.0

Can not multiple move there

Column first

Comma only

Def function badly set up

Disk directory full

Only a letter or numeral may be used here.

The order of calculation number can not be resolved without ambiguity. The expression must be preceded by a reference to the cell that created the SET( ) to which this cell refers. See index under 'ambiguities'.

If calculated the EXP( ) of this value would be more than 1E+38.

ASIN, ACOS, ASINR, ACOSR can only have arguments between 0 and 1.

There is not sufficient space at your specified destination to move this group of lines or columns.

Please insert a first column before you try to enter lines.

You seem to have confused the layout of this function's argument.

The layout of the defined function is wrong.

This message should not happen. You have more than the allowable number of files on the disk already. THE CRACKER can not therefore write to the disk.



**Disk space full** There is no more space on the disk. If you are trying to save your work put in a blank disk and try again. The disk must be blank if you are using CP/M 1.4 or 2.2 or and early version of CDOS as the work on the disk may be indiscriminately overwritten.

**Division by zero** You are trying to divide by zero which would give infinity. THE CRACKER for all its power can not handle that.

**Do split first** You have used one of the slash commands before you have done the partition on which these commands work.

**DO( ) needed with WHILE( )** You can not use the WHILE( ) function on its own. Part of its job is to seek out the DO( ) on the same line.

**E+38 maximum** The maximum exponent that is available in the floating point numbers used in THE CRACKER is +38.

**Entry too long** The maximum entry width is the free area in the middle of the entry line.

**ERROR called from (A22)** This is an error you yourself have called by using the ERROR( ) function.

**FILENAME.EXT not on this disk** The file named can not be found on the disk. Use the W for Which files command to find which files are available.

**File, Print or Mail Label** You asked for an option other than one of these after Copy All.

**Function ((((((5)))))) max** You may only nest functions to 5 levels.

**Logical expression only** The IF( ) function must resolve to a value of TRUE, FALSE, -1 or 0.

**Missing "("** You have too many right hand brackets.

**Missing ")"** You have too many left hand brackets.

**Move cursor** This command does not make sense unless you actually move the cursor.

**Must be a constant** You can not overwrite a formula with a SET( ) or INIT( ) function.

**Must be a letter** A column letter is expected.

**Must be "logical" (-1 or 0)**

**Must be one of +\*/ <>=|[]), or <R>** One of these operators is required.

**Must not refer to this cell** The DO( ) function itself is in the range of its own argument. This is a circular call.

**Negative argument** Can not have a negative argument to a logarithm or square root.

**No function "FUNC( "** This function is not one of the built in functions.

**No such column** You have made a reference to a column outside the worksheet area currently defined.

**No such line** You have referred to a line outside the worksheet area.

**Not enough memory** There is not enough memory left to read in this file.

**Not enough space** Part of your destination area would be outside the worksheet.

**Not found in this range** The LOOKUP( ) or INTERP( ) first argument was not enclosed by any two values in the following list.

**Not in marked zone** You are trying to partition outside the area marked by a grid on the screen.

**Number or value missing** With LOOKUP( ), CHOOSE( ) and INTERP( ) none of the items in the list must be blank.

**Odd number of values required** Simpsons rule works on an odd number of values only.

**Operator at end** You can not end an expression with an operator.

**Out of memory** Your memory is all gone. Your work to date will not be damaged. Try to split your work and write a portion to a file so that you can later consolidate the results.

**Reference outside worksheet** This cell reference is outside the area of the worksheet as you have currently defined it.

**Result more than 1E+38** You have done a mathematical operation that gives too large a result. This is probably



because of an error in one of your constants.

Second "." You can only have one decimal point in a number.

Separate these operators You have two operators together. If you have )( then you must separate them with a \*, thus )\*(.

should be one of ()+\*/ %<>=|[],!f() These are the valid operators at this point.

Stack overflow, shorten entry There are too many pending operations. Cure this by splitting the expression into two or more cells.

Text reference in an expression In this expression you have made a cell reference to a text entry.

Text/Value change You are trying to change a formula into a text entry or vice versa.

The range must increase Only ranges that increase either by column or line are valid.

Too long The search string can only be 30 characters at maximum.

Too many chars Your filename has more than 8 characters.

Worksheet too small You will have to expand the worksheet area before you can copy in the file.

Wrong conditional layout You have confused the layout of this conditional entry.

Wrong range layout

Zero argument You can not have the logarithm of 0.

0...9 or "." This is the start of an exponent and a numeral is expected.

15 sig figs max THE CRACKER calculates to 16-17 significant figures but for security always rounds down to 15.

255 lines max The maximum numbers of lines is 255.

38 places max The floating point number system of THE CRACKER only goes down to E-38.

52 columns max

DAT or .MEM

"end" wrong here

"(" or "," first

"# cannot follow a range

"?" is wrong after a coordinate

"?" is wrong here

"?" should be a numeral

(B22) is used in (C3)

11 to 121

The maximum number of columns is 52, A-Z then a-z.

Only these two file extensions are valid if you are trying to read in a file.

It does not make sense to have "end" as the destination here.

A range must always be preceded by a bracket or comma. It can not be used as part of an expression.

You can not get the current value of a range.

This character is not correct following operator.

This character is not one of the options you were offered in the prompt list.

Only a numeral would be valid here.

You have attempted to remove or overwrite an expression that is referred to in another expression. To do this would stop THE CRACKER being able to do a full recalculation.

These are system error flags. All being well you should never see one however they are the long stops in the event of system bugs. If you get one please note the exact circumstances of how it occurred and inform your supplier. As a general rule your work will not be damaged even if you find such a bug. Just press the ESC key in the usual way and you will be able to carry on.



THE CRACKER  
MORE DETAIL ON PROMPTS

ABCDEFGHIJKLMNOPQRSTUVWXYZ+~\>.\* arrows

The main prompt selection. Auto., Blank, Copy, Down, Edit, Format, Get, Insert, Jump, Left, Move, New, Partition, Quit, Right, Sort, Up, Which files, eXchange, Zap, + page forward, - page back, \ jump up and down windows, / jump left and right windows, > enter, . enter, \* macro, arrows cursor movement.

C,I,W

Copy, Insert, Which files.

crd <R>

Cell reference coordinate then RETURN.

crd...crd <R>

Coordinate range or block reference. Only one period need be inserted, THE CRACKER will insert the other two.

D,F,W

Default format, Format, Width of column.

FILENAME.EXT <R>

Filename in CP/M form followed by a carriage return.

F,P,M

File, Printer, Mail labels.

H,V,S,U,E

Horizontally, Vertically, Synchronised, Unsynchronised, End.

I,D

Increasing, Decreasing.

I,X,Z,space,<DEL>,<R>

Insert, eXchange, Zap (delete), space to move on, DEL key for backspace, Carriage return.

L,R

Left, Right.

L,G,

Line, Column,

r,A,E,B,

File, All, Entry, Block,

U,D,L,R,arrows,J,<R>

Up, Down, Left, Right, arrows, Jump, Carriage return.

V,H

Vertically, Horizontally.

Y,H

Yes, No.

Insert" char,<DEL>,<R>

Insert mode, character, DEL key for backspace, Carriage return to leave mode.

"Xchange" char,<R>

eXchange mode, character, Carriage return to leave mode.

"zap" space,<R>

Zap (delete) mode, space to delete character, Carriage return to leave mode.

R)

Carriage return.

/string/

Delineator, string of characters, delineator.

#,G ,G,H,I,T,P

Numeral, Carriage return format, General, Heading, Integer, Text, Plot.

#,D,E

Numeral, Decimal, Exponent.

#,<R>

Numeral, Carriage return.

,

Numeral.

Calculating

THE CRACKER is at work, you can however enter commands to be carried out when the calculation is complete. (flashing)

Enter characters

You are in Entry mode and THE CRACKER is waiting for you to enter text.

Escape coming

You have pressed the ESC key during a calculation. THE CRACKER is looking for a suitable place to stop without damaging your work. (flashing)

Rewriting screen

The screen is being rewritten. Not always obvious when there are only small changes. (flashing)

Searching

A search is under way for your string.

Sorting

The lines are being sorted. Please be patient if there are a lot of lines to be sorted. (flashing)



INDEX

A  
abandon operation, 8, 13, 16, 21  
ABS, 85, 131, 144  
ACOS, ACOSR, 83, 142  
active cell, 5  
address lists, 94, 97  
Adjust references, 52, 54, 119, 126, 133  
ambiguities, 28, 123  
AND, 143  
arguments, 31  
ASIN, ASINR, 83, 142  
ATAN, ATANR, 83, 142  
AUTOMATIC CALCULATION COMMAND, 6, 128, 136, 139  
AVERAGE, 87, 144

B  
backslash, 63, 159  
backspace, 7, 8, 16, 21, 140, 156, 174  
BLANK COMMAND, 49, 129, 164

C  
carriage return format, 149  
cell, 5  
cell cursor, 5  
CHOOSE, 80, 144  
circular references, 61  
clear, 16, 41, 49  
COMB, combination, 87, 144  
comma, 102, 104, 107, 140  
conditional, 75  
contents line, 7, 20, 139  
CONTROL CODE COMMANDS, 131  
coordinate reference, 5, 8, 139  
COPY COMMAND, 44, 53, 132  
Copying formulae, 133  
copying from files, 55, 133  
copying to files, 55, 57, 134  
copying to printer, 55  
COS, COSR, 83, 142  
COUNT, 85, 144  
CP/M, 3, 55, 131, 160, 163, 173  
CRD, 116, 119, 125, 144, 147  
Cross-references, 129, 164  
CTRL, 13, 131, 135  
cue, 4, 7, 136  
cursor movement, 6, 8, 10, 12, 14, 135

D  
DEC, decrement, 102, 125, 144  
decimal format, 35, 150  
default format, 40, 42, 149, 156, 158

defined functions, 89  
delineator, 92, 152  
DO, 147  
DO, 102, 106, 110, 112, 115, 116, 124, 144  
DPART, 85, 144

E  
EDIT COMMAND, 22, 119, 136  
edit line, 136, 139, 156  
ELSE, 75, 145  
entry line, 4  
ERROR, 75, 144  
error message, 16, 73, 77  
escape key, 5, 7, 13, 16, 18, 21, 77  
EXCHANGE COMMAND, 113, 120, 133, 138  
eXchange mode, 25, 137  
exclamation mark, 34, 140  
EXP, 84, 145  
exponent format, 35, 149, 150  
exponential constant, 84, 144  
EXPRESSION ENTRY, 7, 8, 28, 139

F  
FACT, factorial, 87, 145  
FALSE, 75, 103, 143, 145  
file types, 55, 57, 134  
filename, 55  
finance format, 35, 56, 149  
force recalculation, 120, 122, 128  
foreward slash, 159  
FORMAT COMMAND, 35, 148  
Format types, 149  
forward slash, 68  
functions, 29, 74, 78, 83, 85, 87, 89

G  
general format, 35, 53, 56  
GET COMMAND, 92, 152  
grid, 62, 67  
GROW, 85, 145

H  
hash symbol, 33  
headings, 17, 19, 20, 150

I  
IF, 75, 145  
INC, increment, 102, 125, 145  
INIT, initialise, 102, 110, 125, 145  
initialise, 115  
INSERT COMMAND, 8, 9, 41, 153  
insert mode, 24, 137



INT, 85, 145  
integer format, 35, 150  
internal rate of return, 81, 102  
INTERP, interpolate, 79, 123, 145

J  
JUMP COMMAND, 155  
justification, 17, 19

K  
keyboard, 13

L  
labels, 97, 134  
lists, 82  
LN, 83  
LN, 146  
LOGIO, 84, 145  
logical expression, 75, 102, 145, 147  
LOGICAL OPERATORS, 75  
LOOKUP, 78, 146  
loop counter, 102, 110, 115

M  
MACRO COMMAND, 70, 165  
mathematical operators, 28  
MAX, 85, 146  
memory, 5, 6, 18, 43, 154  
merging, 133  
MIN, 85, 146  
mistake, 5, 7, 16  
MOVE COMMAND, 51, 157

N  
New column width, 19, 40  
NEW COMMAND, 40, 158  
New default format, 40  
New format, 18, 158  
NINT, 85, 146  
normal algebra, 29, 139  
NOT, 143  
NPV, net present value, 80, 146

O  
OR, 143  
order of calculation number, 8, 61, 102, 113, 124, 139  
order of precedence, 29, 139  
Overflow, 151

P  
PARTITION COMMAND, 62, 159  
PERM, permutation, 87, 146

PI, 29, 84, 142, 146  
plot format, 35, 151  
POS, 85, 146  
printer, 48, 131  
prompt line, 4  
prompt list, 4, 44

Q  
QUIT COMMAND, 58, 131, 160

R  
range, 74, 82, 102, 107, 117, 129, 147  
RND, random number, 85, 146  
ROUND, rounded value, 85, 146

S  
searching, 92, 152  
SET, 102, 110, 116, 125, 145, 147  
shift key, 13  
SIMPRULE, Simpsons rule, 81, 147  
SIN, SINR, 30, 81, 83  
SORT COMMAND, 94, 161  
space bar, 26, 120, 136  
SQRT, 84, 147  
STDEV, standard deviation, 87, 147  
SUM, 74, 85, 106, 129, 147  
synchronised, 62, 65, 69

T  
TAN, TANR, 83, 142  
TEXT ENTRY, 9, 17, 156  
text left justified format, 150  
text right justified format, 150  
THEN, 75, 145  
TRUE, 75, 103, 112, 143, 145, 147  
type ahead facility, 13

V  
VAL, value, 125, 147  
VAR, variance, 87, 147  
VERIFY AND CALCULATE COMMAND, 128, 162

W  
what if?, 8  
WHICH FILES COMMAND, 163  
WHILE, 102, 110, 112, 144, 147  
word processor, 55, 131, 134

Z  
ZAP (delete) COMMAND, 41, 164  
zap mode, 26, 137