ACORNSOFT

An electronic spreadsheet
for the BBC Microcomputer

User Guide

VIEWSHEET

ViewSheet

An electronic spreadsheet for the BBC Microcomputer

ACORNSOFT

First published in 1984 by Acornsoft Limited

FIRST EDITION

ISBN 0 907876 82 X

Acornsoft Limited, Betjeman House, 104 Hills Road, Cambridge CB2 1LQ
Telephone (0223) 316039

Contents

Part One Introducing ViewSheet

Part Two ViewSheet facilities

Part One

Introducing ViewSheet

Contents

1 What is a spreadsheet?

The spreadsheet is one of those amazingly useful ideas which is so simple in principle that we all wish we had thought of it first.

Imagine a huge chart composed entirely of rectangular pigeon-holes – in ViewSheet there are as many as 255 by 255.

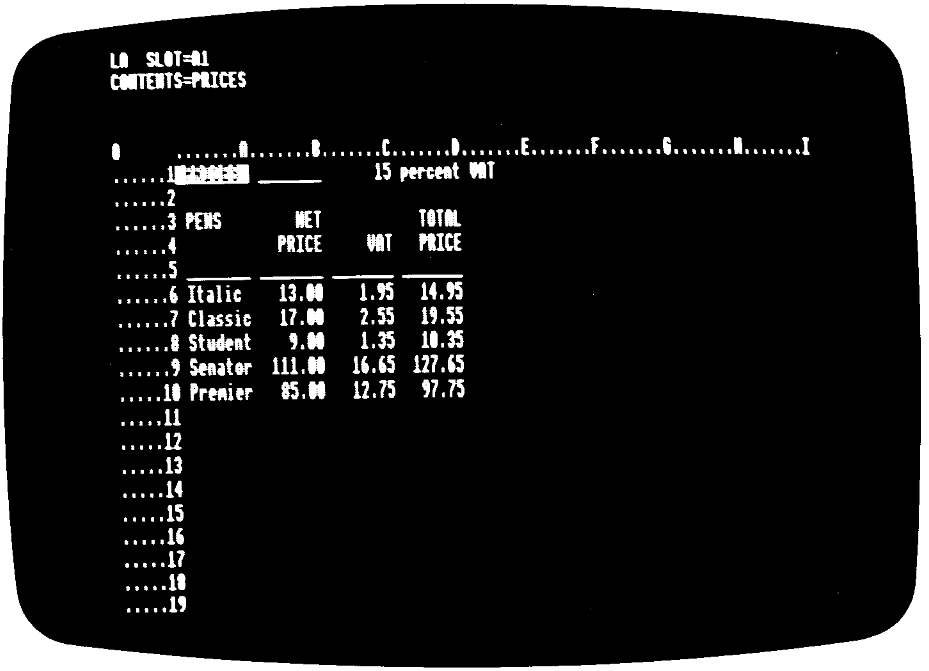
These pigeon-holes (we will call them 'slots') may each be filled with any of the following:

- A number.
- A formula (for example, to multiply the value in another slot by 55).
- A 'label' (for example: TOTAL).

Any slot may be linked with any other slot, so that slots may interact with each other across the length and breadth of the sheet.

Once a sheet is set up, any new amounts entered in it can automatically be related to what is recorded already. Calculations that would be very repetitive if carried out by conventional means are achieved at no greater effort than that of entering the data on which they are based.

The table below is an example of the kind of calculations for which spreadsheets are used. It shows a list of prices followed by the VAT for each item, and the total price including VAT. The rate of VAT is shown in slot C1. Such is the power of the spreadsheet that you have only to enter another rate of VAT in slot C1 and all the prices in the TOTAL PRICE column and all the amounts of VAT in the sheet will adjust themselves automatically.



```
LA  SLOT=A1
CONTENTS=PRICES


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1PRICES  ______       15 percent VAT
......2
......3 PENS        NET            TOTAL
......4            PRICE     VAT   PRICE
......5 _______  _______  _______  _______
......6 Italic     13.00     1.95   14.95
......7 Classic    17.00     2.55   19.55
......8 Student     9.00     1.35   10.35
......9 Senator   111.00    16.65  127.65
.....10 Premier    85.00    12.75   97.75
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

Without going into detail of how this particular display is made up, it is clear that facilities of this kind make spreadsheets extremely valuable for forecasting and testing out possibilities in many areas of work. Once a display is set up, you can vary any key quantities – which might for example be interest or tax rates, prices, sales volume, membership, or growth rate – and see what the effect would be on the system or operation as a whole.

Spreadsheets are widely used, in industry, commerce and research, for recording and comparing data, and for planning and forecasting. At home and in small businesses they are used for basic accounting and records.

ViewSheet is the spreadsheet for the BBC Microcomputer. It allows you to set up displays, vary them at will, save them onto disc or cassette, retrieve them, and print them out in whole or in part. The following pages go into all this in detail.

2 Hands on ViewSheet

The point of this chapter is to gain some idea of what you can do with ViewSheet, so that when you come to explore its various facilities you will see where they all fit in.

Before you start you should place the function key card under the clear plastic strip at the top of the keyboard, lining it up so that DELETE CHARACTER is immediately above function key f9.

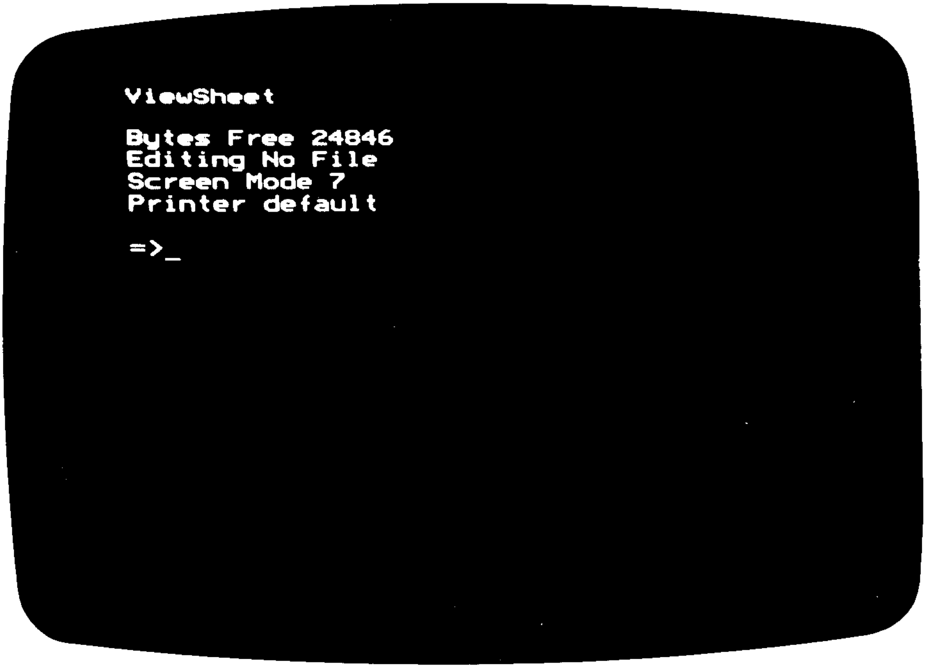
CTRL ▶	AUTO ENTRY	DELETE COLUMN	DELETE ROW							
SHIFT ▶	RECALCULATE MODE	INSERT COLUMN	INSERT ROW	COLUMN HEADING	ROW HEADING	PROTECT COLUMN	PROTECT ROW	RECALCULATE	JUSTIFY LABEL	DELETE SLOT
ViewSheet	REPLICATE	EDIT WINDOW	NEXT WINDOW	DELETE END OF LINE	BEGINNING OF LINE	END OF LINE	EDIT SLOT FORMAT	GO TO SLOT	INSERT CHARACTER	DELETE CHARACTER

From now on the assumption is made that ViewSheet is installed correctly in your computer. See the leaflet accompanying the ROM for details of how this is done.

According to how it is installed, you may be in ViewSheet immediately you switch on the computer or, more likely, you may have to ask the computer for it. To ask the computer for ViewSheet you should type:

`*SHEET` and press **RETURN**.

You should now be looking at something like this:



```
ViewSheet

Bytes Free 24846
Editing No File
Screen Mode 7
Printer default

=>_
```

You will probably also find that the screen is flickering slightly. To stop it doing so, and to change to `MODE 3`, type:

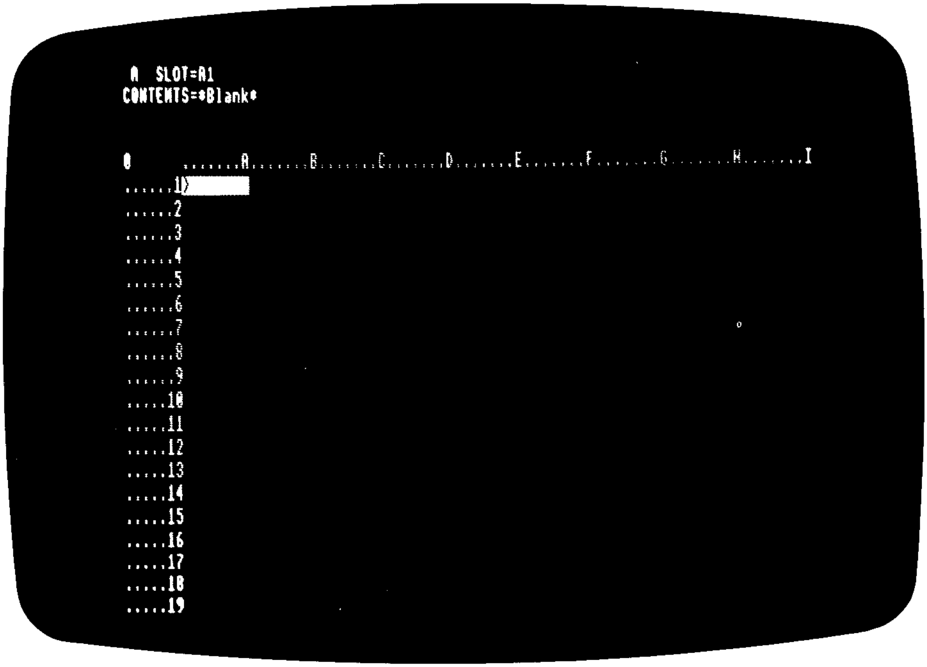
`*TV0,1` **RETURN**
`MODE 3` **RETURN**

From this point onwards in the book the assumption will be made that you are working in `MODE 3`, which means that the screen shows 25 lines of 80 characters each. While other modes are possible, `MODE 3` is the most generally useful.

ViewSheet is now in command mode, the mode in which you can tell the computer to `SAVE`, `LOAD` and `PRINT`. The main mode in which ViewSheet operates is sheet mode. To move into sheet mode press **ESCAPE**.

The **ESCAPE** key switches between command and sheet mode and back again. Switching between modes has no effect whatever on anything you have recorded in sheet mode.

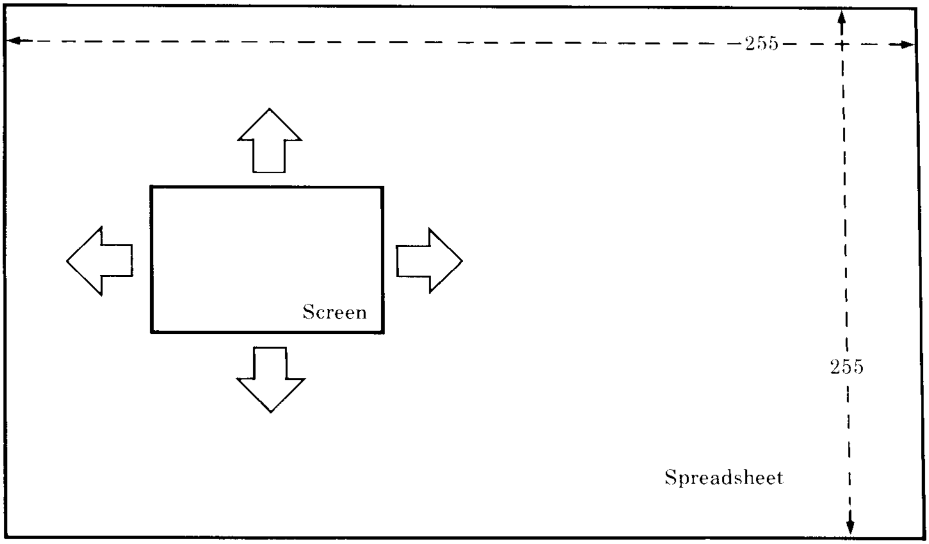
The sheet mode display is like this:



```
 A  SLOT=A1
CONTENTS=*Blank*


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1)
......2
......3
......4
......5
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

The slot which appears on the screen as a white block (this is called 'reverse video') is the slot into which you can enter numbers immediately. This reverse video display is known as the 'cursor', and you can move the cursor to any slot you like by pressing the arrow keys at the right of the keyboard.

Try this for yourself. Press the right or downward arrow key *briefly* several times. Don't try holding it down yet. The cursor jumps from slot to slot each time you press the arrow key.



If you start from the A column and press the right arrow key eight times, the cursor moves to the extreme right position on the screen (ie assuming you are in `MODE 3`). The next time you press the right arrow key you will find the letters marking the columns changing, so that the left-hand column is no longer 'A'.

What is happening is that the screen is moving across the spreadsheet. The columns along the top are labelled A to Z, followed by AA to AZ and so on through to column IU, which is the 255th column. Rows down go from 1 to 255.

If you now hold one of the arrow keys down for a second or two you will notice that the cursor moves several rows or columns one after the other, even though you have pressed the key only once.

This is a BBC Microcomputer facility known as 'auto-repeat'. If you hold a key down, the effect is as if you had pressed that key several times in quick succession.

If you have tried all this you may begin to wonder where exactly you are in the spreadsheet. In order to help you check this ViewSheet gives you a reminder in the top left corner. After `SLOT=` is the reference of the slot where the cursor rests. The form is always the same: column before row – so that A1 is the top left corner. The `CONTENTS=` line below this shows what is in the slot.

When you want to return to slot A1, this is also quite easy. Hold down **SHIFT** and press the upward arrow key a few times; then still holding down **SHIFT** press the left arrow key a few times. The effect of **SHIFT** is to magnify the action of the arrow keys – instead of going from one slot to the next, the window moves in jumps across the sheet each time you press them.

Entering values

Move the cursor back to slot A1 and type any number you like into it. To do this just type the number and press **RETURN**.

Notice what happens. The number you type appears first at the top left of the sheet, and only appears in the slot after you have pressed **RETURN**. At the same time the `CONTENTS=` line shows the number you have entered as the contents of the slot, and the letter `V` appears at top left to show that you have entered a value.

Try several numbers for practice.

Slot references and formulae

We mentioned earlier that you can enter a formula in a slot. In ViewSheet formulae can be as simple as an instruction to add 3 to the amount in another slot, or as complex as those used in statistics.

Note: Multiplying and dividing are done with the signs `*` and `/`.

Move the cursor to any slot you wish and type:

`3.14*123` **RETURN**

The number 386.22 should appear in the slot. Notice that *there is no equals sign.* It is not necessary in ViewSheet since each slot shows the result of whatever value or calculation there may be in it. Try a few more formulae with plus and minus.

By now you will probably have made a few mistakes. If you are typing in a value and find you have typed a wrong digit, you should use the black **DELETE** key at the bottom right of the keyboard. This rubs out the character to its left. If you have already pressed **RETURN**, the simplest way is to place the cursor in the slot that is wrong and overtype the entry by entering another value.

Move to a blank slot and type:

```
12345*5.5 RETURN
```

and the number 67897.5 should appear in the slot.

And you can check the entry by typing:

```
67897.5/5.5 RETURN
```

In practice formulae often contain references to other slots. The simplest possible slot reference occurs when you wish to repeat a number in another slot. Move the cursor to a blank slot, and instead of entering a number type:

```
A1 RETURN
```

Assuming that you have, as suggested above, actually entered something in slot A1 you will see that number reproduced again in the current slot. If you have not, the sheet will show zero for the slot with the formula in it until you enter a value in slot A1.

Notice what happens to the `CONTENTS=` line here. Instead of showing the number in the slot it shows the slot reference you have entered. This is convenient when you are using complex formulae, since by placing the cursor on the slot you can see both the formula (in the `CONTENTS=` line) and the result (in the slot).

Notice also that the letter `V` appears at top left when you enter a reference or formula: these too are values.

Calculation

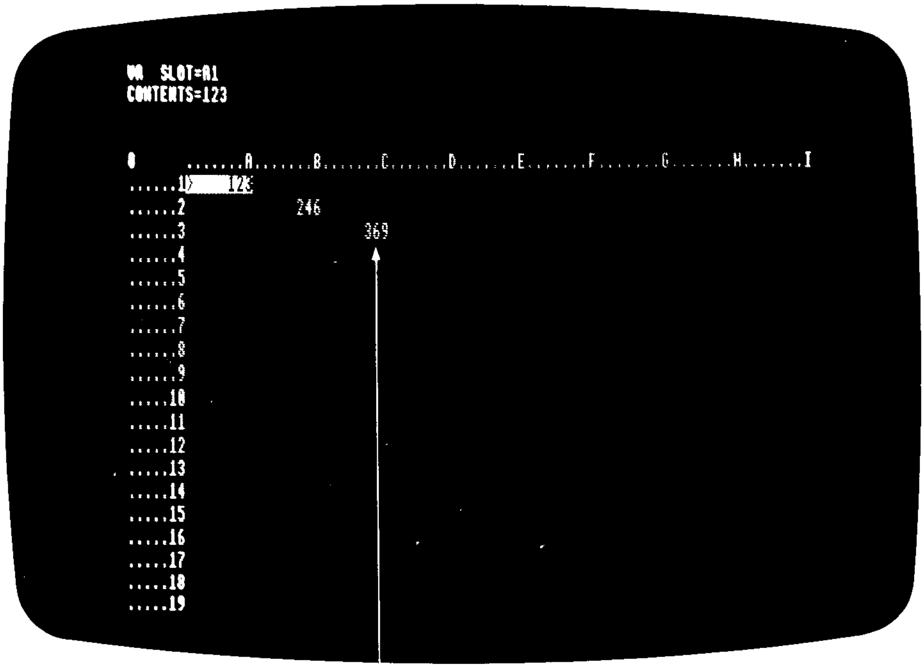
Now to set up an actual calculation. By now you will have a good many unwanted entries all over the screen. To clear the screen completely, press **ESCAPE** to switch to command mode and type:

```
NEW RETURN
```

Press **ESCAPE** again to get back to the sheet (which will now be blank) and enter a value in slot A1. Enter another number in slot B2. Now move to slot C3 and type:

```
A1+B2 RETURN
```

When you press **RETURN** the number that appears in C3 will be the sum of the other two numbers. To prove the working of the spreadsheet, change the numbers in A1 and B2 a few times, and watch the number in C3 change accordingly.



```
VA  SLOT=A1
CONTENTS=123

0     .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1>      123
......2               246
......3                       369
......4
......5
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

A1+B2

Experiment with a few other numbers. For example:

`A1/0.125*D3`	Divides the contents of A1 by 0.125 and multiplies the result by the contents of D3.
`((A6+A7)*D4)+15`	Adds the contents of A6 to the contents of A7, multiplies the result by the contents of D4 and adds 15 to the result.

Notice the use of parentheses here. You will find other mathematical symbols and operators in Part Two.

Recalculation

Every time you change a value, ViewSheet automatically 'recalculates', but it is important to realise the direction in which it does this. Recalculation is from left to right along each succeeding row from top to bottom.

To recalculate a sheet hold down **SHIFT** and press function key **f7** – which you will see on the function key card is 'RECALCULATE'. Alternatively press **TAB** – which also causes the sheet to recalculate.

What is a slot?

Before we go on to entering more complex things in slots, we need to be quite clear as to what a slot is. When you look at the sheet mode display, as it is at the moment, it is clear that seven characters will fit into each slot. The first character space is always blank so as to divide columns from each other.

Formulae can of course be much longer than seven characters, but then they do not appear in the slots. However the values themselves can be longer too. You can prove this by typing:

`1.23456789` **RETURN**

The number that appears in the slot is: 1.23457. ViewSheet has rounded the number to fit the slot, but the number which appears on the `CONTENTS=` line is the full number you entered, and this is the number which will be used in calculations.

As we shall see later, many adjustments can be made both to slots and their contents. Column widths can be increased, values can be ranged left or right, the number of decimal places can be specified, and so on.

Entering labels

So far we have dealt only with entering values in slots, but ViewSheet allows you to enter characters in slots which do not make up any value at all. This is useful for headings and explanations. Such entries are known as 'labels'.

As you know, when you enter a value into a slot the letter `V` for 'value' appears at top left. Similarly when you enter a label the letter `L` appears.

Try entering a word such as 'TOTAL' or 'BUDGET' in a slot. Type the word and press **RETURN** – and the word itself appears in the slot. In contrast to values labels are normally ranged left.

When you press **RETURN**, ViewSheet examines whatever you have placed on the editing line. If it makes sense as a value ViewSheet will place it in the slot, ranged right. Otherwise ViewSheet will assume it is a label, range it left, and not use it in any calculation.

Occasionally this causes a problem. Suppose you want to use the date '1984' as a heading: the simplest way to make sure this is read as a label, not as a value, is to place it in single quotes or between some kind of markers.

Another situation in which this ViewSheet facility may produce puzzling results is when you make a mistake in entering a formula. ViewSheet will then examine your formula, which does not make sense as a value, conclude that it must be a label, and enter it in the slot as such.

Editing slot contents

When this happens, the best way is to *edit* the contents of the slot. We have in fact been using the *editing line* for some time – this is the line on which you type when you place values in slots.

The most radical way of editing the slot contents is simply to replace what is in the slot completely, ie place the cursor on the slot, type new contents and press **RETURN**.

However, if the slot contains a complex formula you may prefer to modify it rather than retype it all. The first step is to *copy* the contents of the slot back into the editing line at the top. To do this, place the cursor in the slot and press **COPY**.

Once the slot contents are in the editing line, a flashing white block is positioned over the first character. This is the line cursor and it is an indication that you can now use ViewSheet's editing facilities to modify the entry. The following are the most important. For the rest see Part Two.

Moving the line cursor

All editing takes place at the cursor position, so you need to be able to move the cursor along the line. To do this hold down **CTRL** and press a right or left arrow key.

Inserting and deleting characters

The INSERT CHARACTER and DELETE CHARACTER keys are **f8** and **f9** – see the function key card. To insert, place the line cursor on the character after the point where you wish the new character to appear. Then press f8 ; the text opens up and you have only to type in the character. For example:

Place the cursor on 'L': `TOTLS`
Press function key f8 : `TOT_LS`
Type 'A': `TOTALS`

Deleting is even simpler: place the line cursor on the character you want deleted and press the DELETE CHARACTER key.

Deleting to the end of a line

For this you use function key f3 . What it does is to delete the character which the line cursor is on, and everything to the right of it on the editing line.

Rubbing out – part of entry

We have mentioned this before. If you have just typed a character which you realise is wrong, press DELETE at bottom right on the keyboard, and the character to the left of the cursor will be deleted.

Rubbing out – whole entry

If at any time you change your mind about the entry you are making and want to rub it out completely and go out of editing mode, you have only to press ESCAPE . This can be used to discontinue any operation on the spreadsheet.

Deleting slots

Another kind of deleting is also available in ViewSheet. This is for deleting the whole contents of a slot. To do this, place the sheet cursor on the slot to be deleted, hold down SHIFT and press function key f9 .

Exercise

Try to construct a 'magic square' just like the one below. Enter the figures in the square simply as numbers. The equals signs and the title will be labels. Use formulae for the totals of the rows and columns.

Magic square

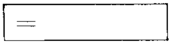
4	9	2	=	15
3	5	7	=	15
8	1	6	=	15
=	=	=		
15	15	15		

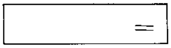
Remember that slot references are entered in formulae like this:

```
A1+A2+A3
```

Did you have trouble placing the labels in line with the numbers? Normally numbers range right and labels left, but you can use spaces to even them up.

A neater way of aligning labels is to make use of ViewSheet's facility for right-justifying labels. Enter your label in the usual way; then leaving the cursor in the same slot press **SHIFT** **f8** (JUSTIFY LABEL) and the slot will change

from 

to 

Additional facilities for entering

With a sheet as small as the magic square it is easy enough to type in all the slot references for the additions. However, there are times when you need to make up a formula containing references from all over the sheet. ViewSheet has two facilities which are particularly useful for this.

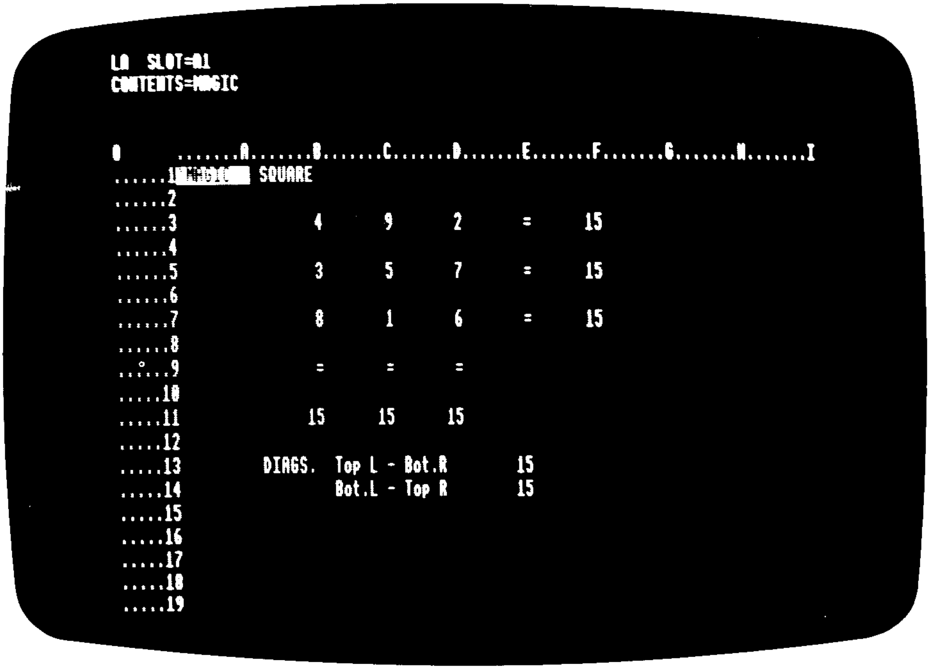
Moving the sheet cursor while editing

Once you have started typing the contents of a slot, you can move the sheet cursor where you like – which can be very useful when you want to check up on a slot reference you are thinking of giving in a formula. When you press **RETURN** the entry appears in the slot you started with, no matter where the sheet cursor may actually be.

Automatic copying of slot references

Position the cursor on a slot. Then hold down **SHIFT** and press **COPY**. The slot reference appears in the editing line. This is a quick and accurate method which can be very useful when making up complex formulae.

For example suppose your magic square is set up like this . . .



```
LA  SLOT=A1
CONTENTS=MAGIC


0       .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1MAGIC    SQUARE
......2
......3              4       9       2       =      15
......4
......5              3       5       7       =      15
......6
......7              8       1       6       =      15
......8
......9              =       =       =
.....10
.....11             15      15      15
.....12
.....13         DIAGS.  Top L - Bot.R       15
.....14                 Bot.L - Top R       15
.....15
.....16
.....17
.....18
.....19
```

. . . and you want to add the totals for the left-right and right-left diagonals at the bottom. Using the two facilities just described the method would be to place the sheet cursor on slot E13 and press the Space Bar once – this gets you into the editing process and ensures that when you eventually press **RETURN** the entry will appear in this slot.

Then move to slot B3, hold down **SHIFT** and press **COPY** – the reference `B3` will appear in the editing line. Type the plus sign and move on to slot C5. Hold down **SHIFT** and press **COPY** again and `C5` will appear in the editing line – and so on.

The whole process is as follows:

Slot	Action	Result in editing line
D13	Space Bar	Editing begins
B3	**SHIFT** **COPY**	`B3`
	+	`B3+`
C5	**SHIFT** **COPY**	`B3+C5`
	+	`B3+C5+`
D7	**SHIFT** **COPY**	`B3+C5+D7`

Then press **RETURN** and the job is done. Use the same method for slot E14.

3 Ranges and replication

It is easy to get the impression, when looking at a finished spreadsheet, that although it is remarkably useful, it must have been very boring to set up. Column after column of figures, differing perhaps only slightly in their makeup; row upon row of statistics, progressing in regular formation across the sheet – surely there ought to be some easier way of setting them up than painstakingly entering every variation of every formula for every slot.

ViewSheet has several labour-saving devices of this kind for entering data, grouping items together, and copying too.

Ranges

Suppose you want to add a whole column of figures together. Rather than specify all the slots in a formula, ViewSheet allows you to specify a 'range', either vertically or horizontally, and add together everything within that range.

To try it out, clear the sheet by pressing **ESCAPE** (and so returning to command mode), typing `NEW` and pressing **RETURN**. Now press **ESCAPE** again and enter a column of figures in column A.

An easy way of doing this is to make use of the AUTO ENTRY mode.

Press function key **CTRL** **f0**
The letter `R` appears at top left on the screen.
Press it again and the `R` changes to `D`.
Press it again and the `D` disappears.

`R` means right and `D` means down. What AUTO ENTRY does is to save you the trouble of pressing the arrow keys every time you want to move to the next slot. Just type the number in and press **RETURN** – and the system moves you on to the next slot automatically.

For present purposes we need the `D` (down) operation. Use it to enter ten figures in A1 to A10. Then move the cursor to slot A15 and type

`A1A14` **RETURN**

Placing the two slot references together in this way is a signal to ViewSheet that you are telling it about a *range* of slots, bounded by and including the two slots named.

The result should be the sum of the whole range in slot A15. Ranges work equally well if all the slots are filled or if some are empty, so you can set a total for a row or column in the form of a range statement, and fill the column in later if you wish.

Replication

Another extremely useful feature of ViewSheet is the way in which it allows you to copy whole ranges of slots either unchanged or with regular variations.

The simplest form of replication is to copy the contents of one slot into another. For example, clear the sheet (with `NEW`) and enter a number into slot A1. Now press the function key for REPLICATE (key **f0**).

The system replies with a prompt:

```
From - To?
```

Type: `A1-C1` **RETURN**

This tells the system to copy the number in slot A1 into slot C1.

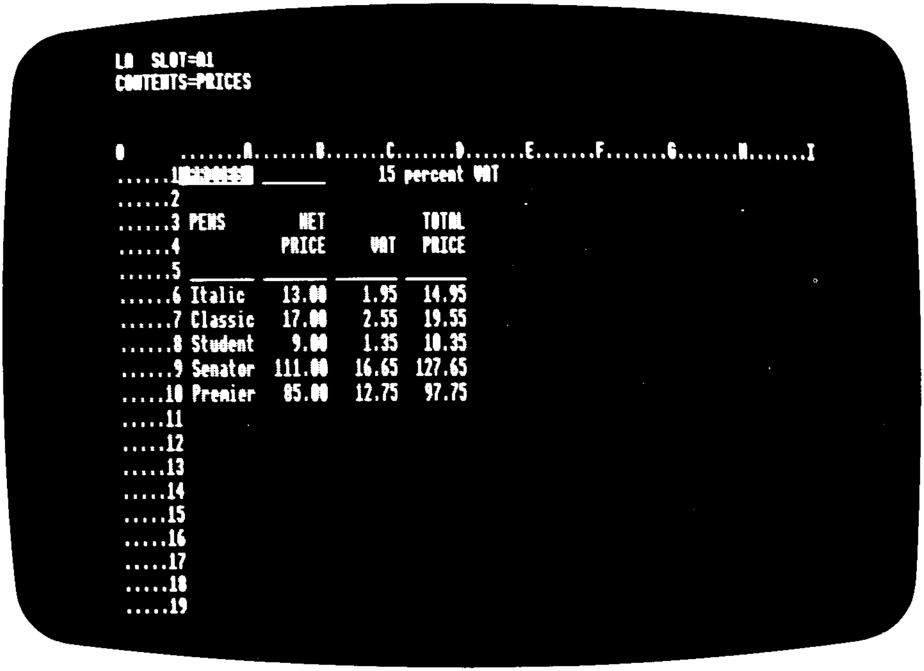
Replication works equally well with slots or ranges. Press REPLICATE (**f0**) again.

Reply: `From - To?`

Type: `A1-B1B19` **RETURN**

This replicates the contents of slot A1 into rows 1 to 19 of column B.

In the examples above, the contents of the slot are copied just as they are with no change whatsoever. This is known as *absolute* replication. An even more useful facility is *relative* replication. For example, consider again the example we looked at in the first chapter.



```
L0  SLOT=A1
CONTENTS=PRICES

0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1PRICES  _______       15 percent VAT
......2
......3 PENS        NET            TOTAL
......4           PRICE      VAT   PRICE
......5 _______ _______  _______ _______
......6 Italic    13.00     1.95   14.95
......7 Classic   17.00     2.55   19.55
......8 Student    9.00     1.35   10.35
......9 Senator  111.00    16.65  127.65
.....10 Premier   85.00    12.75   97.75
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

Obviously the formulae in the VAT column must all involve multiplying by the VAT rate, and the formulae in the TOTAL PRICE column must all involve adding the net price to the VAT amount. So in each of these columns the individual items are very similar to each other, differing only in the slots they refer to. To type them all individually would be a great waste of time.

To show how ViewSheet overcomes this problem we can construct a simplified version of this sheet. You can omit most of the titles, and use just the three columns for NET PRICE, VAT and TOTAL PRICE.

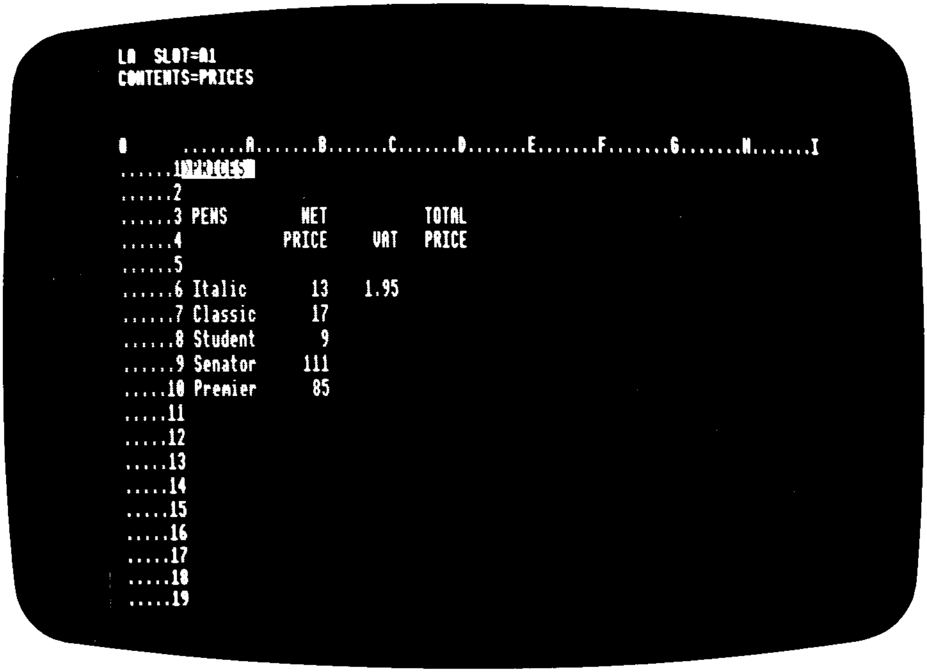
Clear the sheet again, and type the numbers from B6 to B10 – you can use AUTO ENTRY again for practice. You will notice that the amounts will appear without decimal places, eg '85' not '85.00'. We shall deal with ways of lining up the decimal points later.

We now need to type the first item in the VAT column. To make things simpler we will assume a fixed rate of 15 per cent for VAT.

Move the cursor to slot C6 and type:

`B6*0.15` **RETURN**

By now your sheet should look something like this:



```
L0  SLOT=A1
CONTENTS=PRICES


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1PRICES
......2
......3 PENS        NET          TOTAL
......4            PRICE    VAT  PRICE
......5
......6 Italic       13    1.95
......7 Classic      17
......8 Student       9
......9 Senator     111
.....10 Premier      85
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

We now need to replicate this formula in slots C7 to C10, changing only the slot referred to in the formula in each case.

Press: REPLICATE (f0)

Reply: `From - To?`

Type: `C6-C7C10` **RETURN**

Reply: `R)elative, N)o change?`

`B6*0.15`

The reference `B6` appears in black on a white background and the rest of the expression follows, so that you can easily locate the reference.

Whenever ViewSheet is asked to replicate a slot reference it always gives you the choice of replicating it *absolutely*, ie exactly as it is, or replicating it *relatively*. In this case ViewSheet is asking if you wish to multiply slot B6 by 0.15 in every slot from C7 to C10, or whether you want to change the B6 to B7, B8, B9 and B10. Of course this is what we need to do.

Press: R

Immediately the column is filled with figures and if you pass the cursor down you will see in the `CONTENTS=` line at the top of the screen how the formula has been copied with the slot reference updated each time, so that the 15 per cent is applied to the relevant slot in the B column.

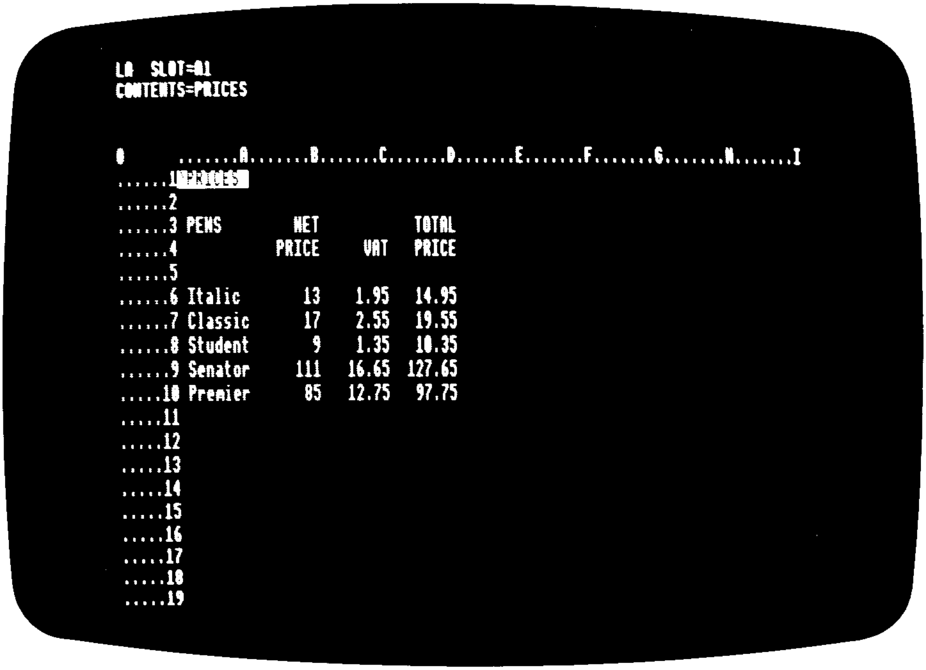
To finish the job enter the formula:

```
B6+C6
```

in slot D6 and replicate that column too, like this.

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `D6-D7D10` **RETURN**
Reply: `R)elative, N)o change?`
`B6+C6`
Press: R
Reply: `R)elative, N)o change?`
`C6`
Press: R

The result should be something like this:



```
LA  SLOT=A1
CONTENTS=PRICES


0     .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1PRICES
......2
......3 PENS        NET          TOTAL
......4            PRICE    VAT  PRICE
......5
......6 Italic        13   1.95  14.95
......7 Classic       17   2.55  19.55
......8 Student        9   1.35  10.35
......9 Senator      111  16.65 127.65
.....10 Premier       85  12.75  97.75
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

A very large display can be constructed quite quickly using methods such as these.

An experiment with replication

For a dramatic illustration of the power of replication, clear the sheet and try doubling numbers starting from 1. That is to say, enter the number `1` in slot A1 and the formula `A1*2` in slot A2. Then replicate slot A2 *relatively* into slots A3 to A255 (ie `A2-A3A255`).

When you do this, ViewSheet appears to pause for some seconds while it works through the procedure. Notice the moving dot at the top left on the screen – just to assure you that ViewSheet is still working on the problem. Notice also the `SLOT=` line: the reference changes continually throughout the replication so that you can check progress. Finally the numbers appear in the slots. As you look down the sheet, you will see that they rapidly become very large indeed, passing into scientific notation at slot A25 and disappearing altogether at slot A128.

Replication in two dimensions

There are times when replication of single slots, of rows or of columns is not enough. Suppose we want to replicate a whole block of rows and columns, not just absolutely but relatively.

Even this is possible. The rule is that you can replicate a column of slots into a number of columns by giving the row at the top of those columns in the 'To' part of your 'From – To' specification. All this is defined more closely in Part Two, but here is an illustration.

Suppose we want to make a 'multiplication square' like the one below, extending it to the traditional 12 × 12 pattern.

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

Try to think how you would solve the problem yourself, in principle at least; then work through the procedure below.

Clear the sheet by typing `NEW` **RETURN**

The sheet cursor will be in slot A1

Type: `1` **RETURN**

Move the cursor to A2
Type: `A1+1` **RETURN**

Slot A2 now shows the value 2

Now replicate the numbers down column A

Press:	REPLICATE (f0)
Reply:	`From - To?`
Type:	`A2-A3A12` **RETURN**
Reply:	`R)elative, N)o change?`
	`A1+1`
Press:	`R`

Slots A1 to A12 now show values 1 to 12

We now have to do the same for slots B1 to L1

Move the cursor to B1

Type:	`A1+1` **RETURN**
Press:	REPLICATE (f0)
Reply:	`From - To?`
Type:	`B1-C1L1` **RETURN**
Reply:	`R)elative, N)o change?`
	`A1+1`
Press:	`R`

Slots A1 to L1 now show values 1 to 12

Now we have to set up the multiplication. We start with column B.

Move the cursor to slot B2

Type:	`A2*B1` **RETURN**

Slot B2 now shows the value 4

To complete column B we replicate the formula in slot B2 down to B12

Press:	REPLICATE (f0)
Reply:	`From - To?`
Type:	`B2-B3B12` **RETURN**
Reply:	`R)elative, N)o change?`
	`A2*B1`
Press:	`R`
Reply:	`R)elative, N)o change?`
	`B1`
Press:	`N`

The second reply has to be `N` so that we multiply by the contents of slot B1 every time. Column B will now be filled in. All we now have to do is to replicate this column over the rest of the sheet.

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `B2B12-C2L2` RETURN
Reply: `R)elative, N)o change?`
`A12*B1`

ViewSheet will now ask you for a decision on many slots one after the other. Every time the slot reference begins with B, press `R` (since we want the values multiplied by the contents of slots C1 to L1). Every time the slot reference begins with A, press `N` (since we want the value multiplied by the contents of slots A1 to A12).

The result should be the complete table.

An alternative method

While the method shown above makes for an extensive illustration of replication methods, it is not the only way of doing this particular job, and certainly not the easiest.

To try another way, clear the sheet. Then place the sheet cursor in slot C6 and type:

`ROW` RETURN

The number 6 appears in the slot. Now move to slot C8 and type:

`COL` RETURN

The number 3 appears in the slot. The commands `ROW` and `COL` give the number of the row and column where the entry is typed, the columns being considered as numbered 1 to 255. To confirm this move to slot D12 and type:

`ROW*COL` RETURN

The result will be the value 48.

We will show something of the value of these functions later. For the moment we can use them to make up a multiplication square very simply. Think out how you would do it; then read the method below.

Clear the sheet. The cursor will be in slot A1.

Type: `COL*ROW` **RETURN**
Press: REPLICATE (**f0**)
Reply: `From - To?`
Type: `A1-B1L1` **RETURN**

Slots A1 to L1 will contain values 1 to 12

Press: REPLICATE (**f0**)
Reply: `From - To?`
Type: `A1L1-A2A12` **RETURN**

This replicates the top row into the subsequent 12 rows. Notice that every slot has the same formula: `COL*ROW`

Although we have used the values from `COL` and `ROW` as they are, these values can of course be adjusted at will. For example if you enter `ROW-100` in slot A101 the result will be 1.

4 Windows and headings

The size of ViewSheet – 255 by 255 slots – is impressive, allowing room for you to develop all sorts of intricate and interrelated calculations. If the calculations spread at all widely, however, you may find yourself hunting all over the sheet for the key totals which show you the results of all your forecasting or planning.

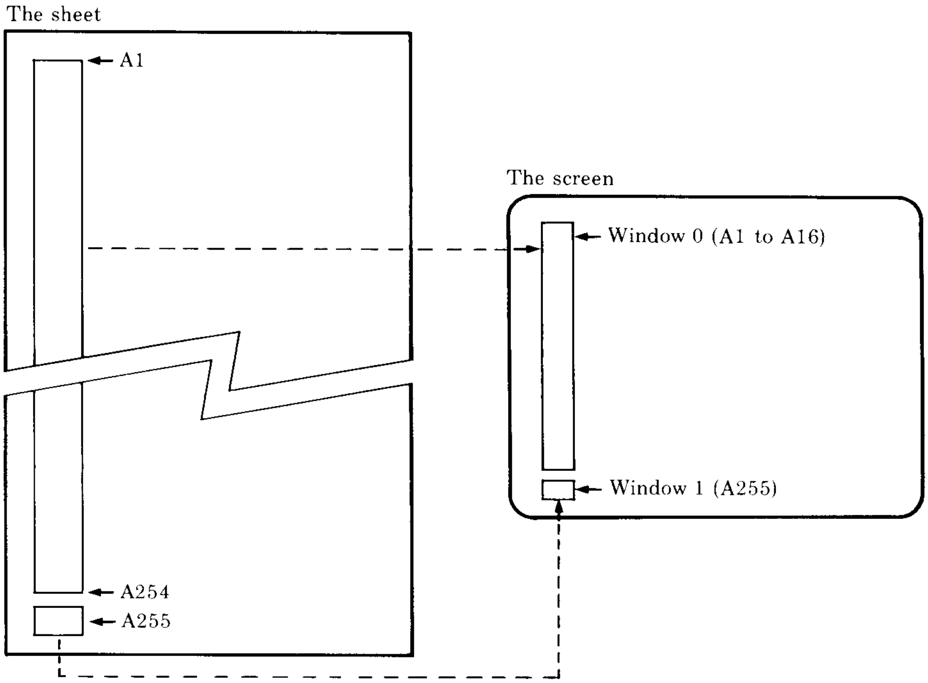
Once you are satisfied that your display is working as it should, ViewSheet offers a way in which you can group all the key figures together on the screen, even though they may come from widely separated parts of the sheet.

To take a simple example, suppose you want to add together all the numbers you may happen to enter in column A. You do not know how many there are going to be so you enter in slot A255 the formula to add all numbers in the range:

```
A1 A254
```

The only problem is that whenever you want to check the total you have to go down to row 255. Even if you use the GO TO SLOT key (function key f7) it is a tiresome process.

The window facility offers a solution to this problem. It allows you to change the screen display completely, so that part of the screen is designated as a window showing column A, and another part of the screen is designated as a window showing slot A255. The rest of the screen is blank.



In ViewSheet you are allowed up to ten screen windows, numbered 0 to 9. To show how the system works, set up the example we have just considered as follows:

To enter the range in slot A255:

Press: GO TO SLOT (f7)
Type: `A255` RETURN

The cursor moves to slot A255

Type: `A1A254` RETURN
Press: GO TO SLOT (f7)
Type: `A1` RETURN

From now on anything you type in column A is totalled in slot A255. Enter a few numbers in the column to show how it is working. Now to make the sheet more convenient to use. First we set up part of the screen as window 0, which will show column A only.

Press: EDIT WINDOW (f1)

Reply: `Window?`

Press: 0 **RETURN**

Reply:

```
Wi      TopL   BotR   Pos    Cw     Bw     Fmt    Opt
0       A1     I19           7      7      FRM
```

Part Two goes into the meanings of all these in greater detail. For the moment, you need only know that:

`Wi` = Window number
`TopL` = Top left
`BotR` = Bottom right
`Pos` = Position
`Cw` = Column width
`Bw` = Border width
`Fmt` = Format
`Opt` = Options

The next task is to reset window 0 so that it occupies a strip on the left of the screen. To change the window parameters, all you have to do is to type new numbers in place of the old ones.

When you do this remember to hold down **CTRL** as you move the cursor with the arrow keys. To set a window consisting of a single column on the left, reset them as follows:

```
Wi      TopL   BotR   Pos    Cw     Bw     Fmt    Opt
0       A1     A16           7      7      FRM
```

Press **RETURN** to enter the new parameters and you should see something like the following (the numbers shown here are purely random):

```
            ........A
........1      8274      ........9      11
........2       536      .......10     334
........3        34      .......11      45
........4      1631      .......12    2311
........5        74      .......13      32
........6         8      .......14       2
........7       173      .......15      56
........8      1234      .......16    1234
```

It is, however, possible that instead of giving you a display like this, the computer emits a bleep and the cursor positions itself on one of the window parameters. This is a signal that you have got something wrong. Compare what you have entered with the information above and correct it before pressing **RETURN** again.

The slots in this window (consisting of a single column) can be scrolled up and down and sideways in the same way as a complete screen.

Since the rest of the screen is blank, we can place the total (slot A255) either below or to the right of window 0. To do this we need to define another window which will consist simply of slot A255. We can make this window 1.

Press: EDIT WINDOW (**f1**)
Reply: `Window?`
Press: 1 **RETURN**
Reply:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
1     A1    A1    R0    0     0     FRM   O
```

(Since window 1 is not operating at switch-on, the numbers under these headings are meaningless.)

Type in new numbers as follows:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
1     A255  A255  B0    7     7     FRM
```

Notice that A255 is given as both top left and bottom right, since it is the only slot. The position of window 1 is to be *below* window 0 and this is shown by the characters: `B0`.

It is particularly important to delete `O` from the `Opt` column. `O` turns the window off.

When you have done all this press **RETURN**, and if you have entered the numbers correctly you should see the two windows on the screen. Scroll window 0 downward with the arrow key and enter another item or two, noticing how the total changes.

Having set up a two-window display and seen how it works, it is important to understand what is actually happening. When you switch ViewSheet on and start using sheet mode, what you are actually looking at on the screen is screen window 0. The default size of screen window 0 is the full size of the screen, so in `MODE 3` it is A1 to I19.

Obviously while window 0 is occupying the whole of the screen, it is impossible to set up any other windows. So before you set up others you must reduce the size of window 0, as we did in the example above.

When placing windows on the screen, the position is specified by the characters under `Pos`. A new window can be positioned either below or to the right of an existing window. `B` means below and `R` means right. Thus as in the example above, `B0` means below window 0. `R1` would be to the right of window 1.

If you specify a window for which there is not enough room on the screen, ViewSheet will try to fit in as much of it as possible, by reducing its size. If even this is impossible ViewSheet will turn the new window off.

The parameter `Cw` means column width. Window 0 defaults to seven characters, but you can change this as you wish. When you do, bear in mind that wider columns must mean that fewer columns will fit onto the screen.

Scrolling

We mentioned earlier that when you move the cursor to the right or bottom of the screen, and then continue to press the arrow key, the screen moves over the sheet, and the row and column headings change accordingly. This effect is known as 'scrolling'.

It would in fact be more correct to say that it is the *window* which moves over the sheet, rather than the *screen*, and this effect can be demonstrated with any window you set up.

To move the cursor to another window:

Press: NEXT WINDOW (f2)

This causes the cursor to move to each window on the screen in succession.

You can in fact cause several different windows to scroll in harmony with each other. For example you might have names of competitors and their scores in adjacent windows. In such a case you can use the `Opt` parameter. To cause two columns to scroll vertically together, place `V` under `Opt`. To cause them to scroll horizontally together, use `H`.

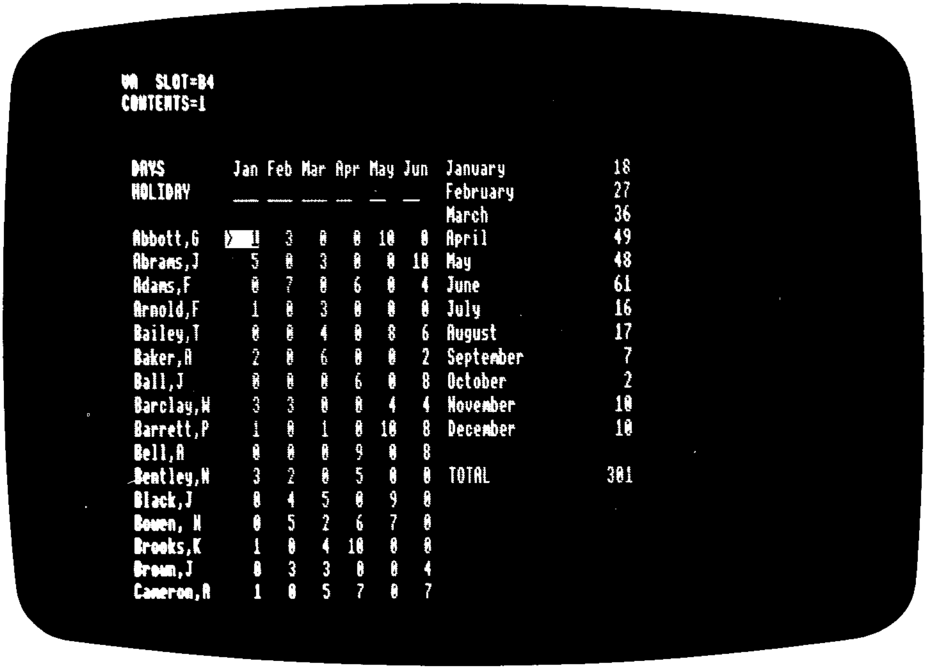
Turning off headings

As you will have noticed if you tried out the last example, row and column headings can take up a good deal of room. In some displays of screen windows it is convenient to turn them off. The character `T` under `Opt` turns off the 'top' headings, ie the column headings, while `S` turns off the side headings. You can use `TS` to turn off both.

Windows are also used in printing out parts of the sheet. This is dealt with in the next chapter.

Example

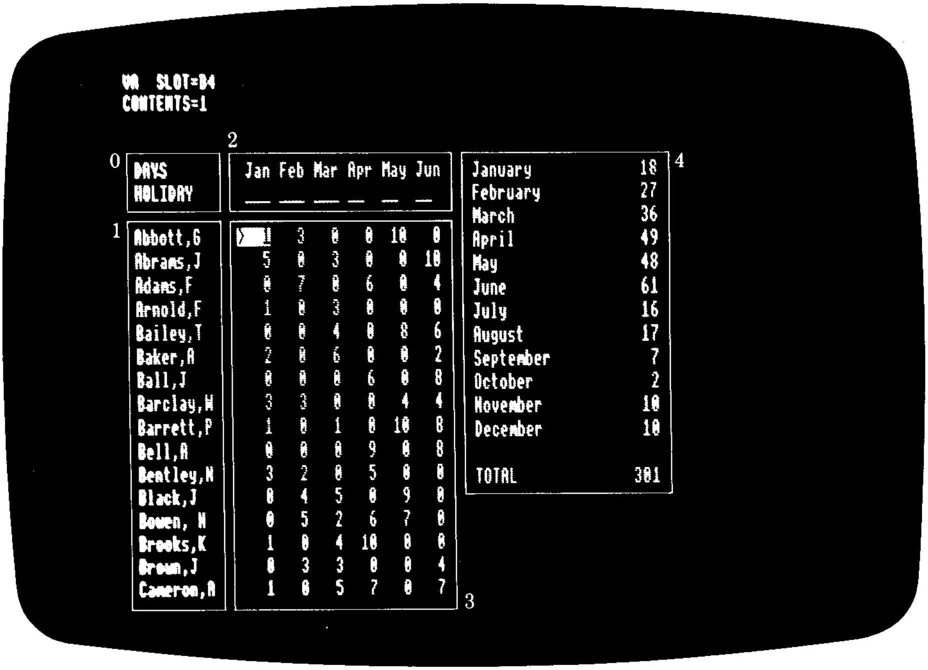
The following sheet makes use of several of these facilities. The objective is to make up a holiday chart like this:



```
WA  SLOT=B4
CONTENTS=1


DAYS        Jan Feb Mar Apr May Jun  January         18
HOLIDAY     ___ ___ ___ ___ ___ ___  February        27
                                     March           36
Abbott,G    > 1   3   0   0  10   0  April           49
Abrams,J      5   0   3   0   0  10  May             48
Adams,F       0   7   0   6   0   4  June            61
Arnold,F      1   0   3   0   0   0  July            16
Bailey,T      0   0   4   0   8   6  August          17
Baker,A       2   0   6   0   0   2  September        7
Ball,J        0   0   0   6   0   8  October          2
Barclay,W     3   3   0   0   4   4  November        10
Barrett,P     1   0   1   0  10   8  December        10
Bell,A        0   0   0   9   0   8
Bentley,N     3   2   0   5   0   0  TOTAL          301
Black,J       0   4   5   0   9   0
Bowen, N      0   5   2   6   7   0
Brooks,K      1   0   4  10   0   0
Brown,J       0   3   3   0   0   4
Cameron,A     1   0   5   7   0   7
```

The heading DAYS HOLIDAY and the monthly summary and TOTAL at the right do not scroll in any direction. The names and numbers under the months scroll vertically in harmony with each other so that you can move to the appropriate name and enter that person's days. The numbers and months scroll horizontally in harmony with each other, so that you can move through the year.

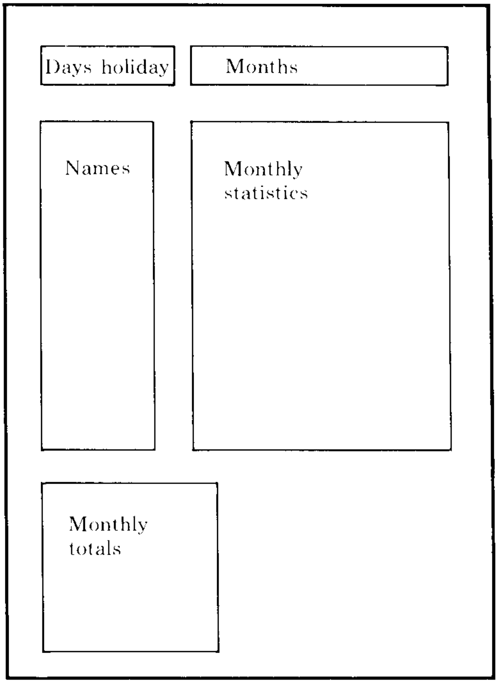
To do all that we need five windows arranged like this:



```
VA  SLOT=B4
CONTENTS=1

DAYS          Jan Feb Mar Apr May Jun    January          18
HOLIDAY       ___ ___ ___ ___ ___ ___    February         27
                                         March            36
Abbott,G      >  1   3   0   0  10   0   April            49
Abrams,J         5   0   3   0   0  10   May              48
Adams,F          0   7   0   6   0   4   June             61
Arnold,F         1   0   3   0   0   0   July             16
Bailey,T         0   0   4   0   8   6   August           17
Baker,A          2   0   6   0   0   2   September         7
Ball,J           0   0   0   6   0   8   October           2
Barclay,W        3   3   0   0   4   4   November         10
Barrett,P        1   0   1   0  10   8   December         10
Bell,A           0   0   0   9   0   8
Bentley,N        3   2   0   5   0   0   TOTAL           301
Black,J          0   4   5   0   9   0
Bowen, N         0   5   2   6   7   0
Brooks,K         1   0   4  10   0   0
Brown,J          0   3   3   0   0   4
Cameron,A        1   0   5   7   0   7
```

Note that the boxes and window numbers on the above example have been inserted for clarity. They do not appear on the screen.

The easiest way to construct a complex sheet like this is first to set it up with window 0 only, and only then to apply the special window arrangement which makes it really useful. The original layout of the sheet is like this:



It is of course necessary to place the monthly summary below the main statistics, because the sheet recalculates row by row downwards.

Before you enter the information in the slots you need to adjust the width of the columns. Few names are as short as seven characters, so we will increase the width of the columns to ten characters, like this.

Press: EDIT WINDOW (f1)
Reply: `Window?`
Type: `0` **RETURN**
Reply:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
0     A1    I19         7     7     FRM
```

Change the column width from 7 to 10 and press **RETURN**. Notice that window 0 now extends from A1 to F19 only, because of the additional column width.

The slot contents are as follows. We give them here so that when you construct your sheet you will be able to compare the results with those shown in this book.

A1 DAYS
A2 HOLIDAY
A4 to A53 a list of names – see example above

A57	January	B1	Jan	B2	____
A58	February	C1	Feb	C2	____
A59	March	D1	Mar	D2	____
A60	April	E1	Apr	E2	____
A61	May	F1	May	F2	____
A62	June	G1	Jun	G2	____
A63	July	H1	Jul	H2	____
A64	August	I1	Aug	I2	____
A65	September	J1	Sep	J2	____
A66	October	K1	Oct	K2	____
A67	November	L1	Nov	L2	____
A68	December	M1	Dec	M2	____
A70	TOTAL				

B57	B4B53	B63	H4H53
B58	C4C53	B64	I4I53
B59	D4D53	B65	J4J53
B60	E4E53	B66	K4K53
B61	F4F53	B67	L4L53
B62	G4G53	B68	M4M53
		B70	B57B68

The names of the months are shown in abbreviated form in row 1 so that we can later reduce the size of these columns to only three characters in order to display more columns than would otherwise be possible on the screen. The rules in row 4 divide the headings from the rest of the columns.

You should now be left with a large empty area into which the holiday statistics will eventually be entered. Since we want to place this on the screen with window definitions later, it would be wise to fill all the slots with zeros now, so that the slots are made visible.

Move the cursor to slot B4

Type: `0` **RETURN**
Press: REPLICATE (**f0**)
Reply: `From - To?`
Type: `B4-C4M4` **RETURN**

This fills row 4 with zeros

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `B4M4-B5B53` **RETURN**

This fills the whole block with zeros (which takes a little time, so be patient!)

We are now ready to set up the window definitions. Look back first at the window diagram above and then proceed as follows.

The first thing to do is to reset window 0 (which at present covers the whole screen) and so make room for the other windows.

Press: EDIT WINDOW (f1)
Reply: `Window?`
Type: `0` **RETURN**
Reply:

```
Wi   TopL  BotR  Pos   Cw   Bw    Fmt   Opt
0    A1    F19         10   7     FRM
```

Change this to:

```
Wi   TopL  BotR  Pos   Cw   Bw    Fmt   Opt
0    A1    A2          10   7     FRM   TS
```

Note that the border widths in this and the following windows have no meaning, since the borders are turned off.

This should leave you with the words `DAYS HOLIDAY` in the top left of the sheet, with the rest of the screen blank.

Follow the same procedure with the other windows, setting parameters as follows:

Window 1:

```
Wi   TopL  BotR  Pos   Cw   Bw    Fmt   Opt
1    A4    A19   B0    10   10    FRM   VTS
```

Window 2:

```
Wi   TopL  BotR  Pos   Cw   Bw    Fmt   Opt
2    B1    G2    R0    3    3     FRM   HTS
```

Window 3:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
3     B4    G19   B2    3     3     FRM   VHTS
```

Window 4:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
4     A57   B70   R2    10    10    FRM   TS
```

As you go through this you will see the display build up step by step until you finish with the display shown earlier in this chapter.

If you make a mistake, simply overtype the definition. Having constructed the display try entering numbers in it, and the value of such a layout will soon become apparent. Check that the columns and months align with each other horizontally, and the names and numbers vertically. When you do this you may find yourself moving the columns too far, but you can easily recover if you do it slowly. If you do get lost you can restore the display to its original position by pressing GO TO SLOT (**f7**) and asking for slot B4.

Saving for later use

We shall go into the use of cassettes and discs later. It would, however, be very useful to keep a copy of the HOLIDAY sheet now, for use in the next chapter of this book. To do this, first decide on a name for your sheet (for example, HOLIDAY – since the system will accept names of up to seven characters). Then proceed as follows:

Cassette

Place a blank cassette in the recorder.
Type: `SAVE filename` **RETURN**
Reply: `RECORD THEN RETURN`

Press RECORD on your cassette recorder and **RETURN** on the computer. When the prompt (=>) returns, the file is recorded.

Disc

Make sure there is a disc in the disc drive.
Type: `SAVE filename` **RETURN**

Headings

We have already looked at ViewSheet's use of labels in slots. Another way in which you can indicate the meaning of the numbers displayed is by changing the row and column headings from the normal 'A, B, C' and '1, 2, 3' etc to words of your own choosing.

Two commands listed on the function key card, COLUMN HEADING and ROW HEADING, allow you to do this. For example suppose you wish to change the column heading 'A' into 'GROUP'.

Place the cursor in column A
Press: COLUMN HEADING (**SHIFT** **f3**)
Reply: `Column heading?`
Type: `GROUP` **RETURN**

The number of characters shown on the sheet will of course be limited by the width of the column.

Replacing row headings works in the same way using the ROW HEADING command (**SHIFT** **f4**). The space for row headings can also be adjusted and this will be shown later.

Notice how the `SLOT=` line at the top shows the slot reference. If for example you are in what was previously slot A5, and you have changed the column heading to 'GROUP' as suggested, the reference will be: `SLOT="GROUP"5`

If you then change the heading for row 5 into 'April', the slot reference for the former slot A5 will become: `SLOT="GROUP"April"`

In future whenever you refer to this slot you can either use the original slot reference (A5) or the new one, but if you use the new one you must use the quotes as above: `"GROUP"April"`. Notice that it is unnecessary to double the quotes in between two references, although ViewSheet will accept the reference if you do so.

ViewSheet will *accept* the entry if you fail to use quotes at all, but it will not understand it as a slot reference. The entry will therefore make no sense as a value and ViewSheet will assume it is a label.

An advantage of using headings of this kind is that if for example you move down to row 220, the heading 'GROUP' will still be visible in the first column – which would not be the case with labels in specific slots.

You can also turn your special headings on and off at will. Press **ESCAPE** to change to command mode and type: `HEADINGS OFF` **RETURN**

When you press **ESCAPE** to return to sheet mode you will find the standard row and column headings restored. To bring back your alternative headings you have only to switch to command mode and type: `HEADINGS ON` **RETURN**

If you wish to use a standard heading once only, and do not wish to go to the trouble of switching to command mode and back twice over, you can copy the *standard* reference of any slot into the editing line by placing the cursor in the slot, holding down **SHIFT** and pressing **COPY**.

We mentioned this before as a quick way of placing slot references in a formula, but it is worth remembering that it provides standard slot references even where user-defined headings have been set up.

5 Printing out

Printing from ViewSheet is very simple, provided you have the system set up correctly. Any printer which will work with the BBC Microcomputer will work with ViewSheet, and generally we are assuming in this book that you already have your printer connected and working.

Commands to tell the computer what kind of printer is connected and at what rate to transmit data to it are used in command mode.

For parallel printers type: `*FX5,1` **RETURN**
For serial printers type: `*FX5,2` **RETURN**

Transmission rates are selected by one of the following and **RETURN**.

75 baud	`*FX8,1`
150 baud	`*FX8,2`
300 baud	`*FX8,3`
1200 baud	`*FX8,4`
2400 baud	`*FX8,5`
4800 baud	`*FX8,6`
9600 baud	`*FX8,7`
19200 baud	`*FX8,8`

So in order to make the printer work with ViewSheet:

- Enter command mode by pressing **ESCAPE**.
- Enter the `*FX` commands as described above.
- Type `PRINT` and press **RETURN**.

In its simplest application the action of the `PRINT` command is quite straightforward. If you switch on ViewSheet and make no window adjustments, its action is in effect to print out whatever you see on the screen in the top left corner of the sheet. For example if you are in `MODE 3`, everything from A1 to I19 is printed.

To print other parts of the sheet, ViewSheet uses 'print windows', which work in a very similar way to screen windows.

Instead of being placed on the screen, print windows are placed in much the same way on paper. The maximum print area that ViewSheet can handle is 255 lines deep by 255 characters wide, although of course in practice this last measurement may not be usable if your printer actually prints, say, an 80-character line.

ViewSheet allows you to place print windows P0 to P9 anywhere within this area, just as you place screen windows on the screen.

At switch-on, or after typing `NEW` to clear the sheet, the only windows in operation are screen window 0 (which is the whole of the screen) and print window P0 (which is the same size).

Screen windows 1 to 9 and print windows P1 to P9 are not operating, but can be brought into operation if you define them.

If you are in `MODE 3` therefore your screen shows A1 to I19, and if you issue a command to print, the printer gives you the contents of slots A1 to I19.

In the last chapter we demonstrated screen windows by setting up a column of figures in column A, and entered the range `A1 A254` in slot A255 to total all the values entered in column A. We then defined screen window 0 to make the column of values appear on the left of the screen, and window 1 to show the total on the right of the screen.

To give a feel for how print windows work, we can do much the same with them. First enter a few values in, say, slots A1 to A15; then place a range addition for the whole column in slot A25:

```
A1 A24
```

Having done that we can set up print windows, just as we set up screen windows.

Press: EDIT WINDOW (**f1**)

Reply: `Window?`

Type: `P0` **RETURN**

Reply:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
P0    A1    I19         7     7     FRM
```

(This reply assumes you are in `MODE 3`, ie with the window covering A1 to I19.)

Use the arrow keys to move the cursor along and change the definition to:

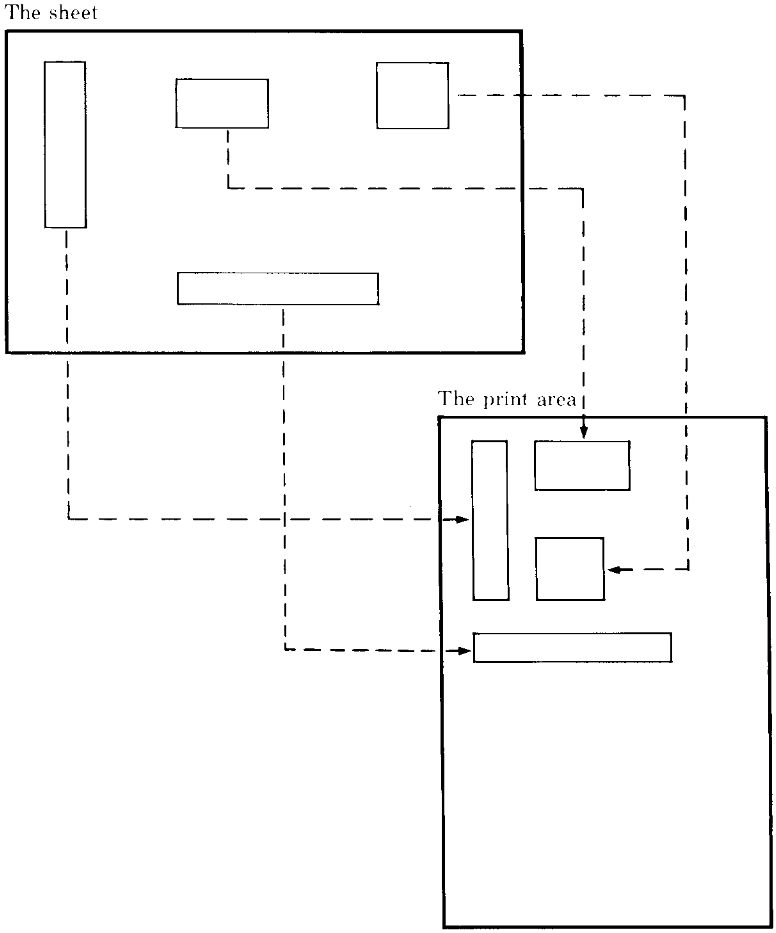
```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
P0    A1    A24         7     7     FRM   TS
```

Press **RETURN** and the system is set to print a single column on the left without row and column headings, which are turned off with option `TS`.

Now use the same methods to set window P1 to:

```
Wi     TopL  BotR  Pos    Cw     Bw     Fmt    Opt
P1     A25   A25   R0     7      7      FRM    TS
```

Press **RETURN** and change to command mode by pressing **ESCAPE**.



An example of how windows can be grouped on the print-out

First make sure your printer is ready. Then type `PRINT` and press **RETURN**. Your printer should produce a column of numbers to the left and their total to the right, just like the screen display in the last chapter.

Example

An ideal example to illustrate print windows is the HOLIDAY sheet we set up with screen windows in the last chapter. We suggested that you should save it for later use, so if you did you should load it again now.

Before loading you should make sure you are in the same mode as when you saved the sheet – we recommended `MODE 3`.

We shall go into methods of using cassettes and discs later. For the present purpose first check on the filename you used to save the sheet (HOLIDAY?); then proceed as follows:

Cassette

Enter command mode by pressing **ESCAPE**.
Place the cassette with the sheet you require in the recorder.

Type: `LOAD filename` **RETURN**
Press PLAY on the cassette recorder.
Press: **RETURN**

When the prompt (=>) returns, the file has been loaded.

Disc

Make sure the right disc is in the drive.

Type: `LOAD filename` **RETURN**

The sheet should now be displayed with windows set exactly as when you last saw it. However the windows set are screen windows, whereas we now need print windows set to the same definitions.

In fact it is surprisingly easy to do this. All that is necessary is to ask for the screen windows to be displayed, and change the window number by putting a 'P' in front of it.

Press: EDIT WINDOW (**f1**)
Reply: `Window?`
Type: `0` **RETURN**

```
Reply:  Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
        0     A1    A2          10    7     FRM   TS
```

Change `0` to `P0` and press **RETURN**.

The rest of the windows should be changed in the same way:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
P1    A4    A19   B0    10    10    FRM   VTS
P2    B1    G2    R0    3     3     FRM   HTS
P3    B4    G19   B2    3     3     FRM   VHTS
P4    A57   B70   R2    10    10    FRM   TS
```

Now press **ESCAPE** to switch to command mode.

Prepare for printing as described above and type `PRINT` **RETURN**.

The result should be a replica in print of the screen display.

SCREEN

The `SCREEN` command can be used in command mode to display the sheet. This can be a convenient way of checking that you have set your print windows correctly, since this command uses print windows only.

Press: **ESCAPE** to enter command mode
Type: `SCREEN` **RETURN**

Press **ESCAPE** to end the display.

Word processing

One of ViewSheet's facilities is its ability to cooperate with VIEW, the word processor.

Instead of printing sheets directly, ViewSheet allows you to incorporate them into text which has been generated on VIEW, and so print both text and sheet together.

Details of how to do this are given in Part Two chapter 11.

6 Managing sheets

If you are trying out all the examples and techniques as you work through this book you will have become aware of the need to modify sheets once they have been designed.

At first of course you become aware of this need through making mistakes and having to correct them. Later it is a matter of enhancement rather than correction. You design a sheet and make it work, only to realise that it could be modified to do the job much better or to do other jobs too. All it needs perhaps is to have a few slots modified, a column or row inserted here or there, a few slots deleted, and the labels moved around to match the numbers. All this can be done.

Slot editing

Editing the contents of a slot, for example, is managed mainly through the function keys. We have mentioned several of these incidentally, but the editing functions are worth a reminder. Try them out for yourself.

Selecting the slot

To move the cursor to a specified slot press GO TO SLOT (f7). ViewSheet replies: `Slot?` Type the slot reference and press **RETURN**.

To delete the slot contents completely place the sheet cursor in the slot and press DELETE SLOT (**SHIFT** f9).

To copy a slot into the editing line place the cursor in the slot and press **COPY** .

Moving the line cursor

To move the line cursor along the slot contents (on the editing line) hold down **CTRL** and use the arrow keys. If the slot contains a very long formula or label, it will not all show in the editing line, but you can still run the cursor along it in the same way. As you move the cursor right more characters appear at the right and disappear to the left. When moving left the same thing happens in reverse.

To move the cursor to the end of the line press END OF LINE (f5).

To move the cursor to the beginning of the line press BEGINNING OF LINE (f4).

Deleting and inserting

To delete a character place the cursor on the character and press DELETE CHARACTER (f9). The character is deleted and the text closes up.

To delete the character to the left of the cursor use **DELETE** (ie the black **DELETE** key at bottom right on the keyboard).

To insert a character place the cursor on the character space where you want the new character to appear (ie usually under the next character) and press INSERT CHARACTER (f8). The text opens up and you can type in the new character.

To delete everything to the right of the cursor together with the character the cursor is on, press DELETE END OF LINE (f3).

Entering slot references

Two facilities which speed entry of slot references are the ability to move the sheet cursor during entry, and the ability to copy the slot reference automatically.

Once an entry has been started, even if only by pressing the Space Bar, the line cursor appears in the editing line and editing has begun. Wherever the sheet cursor is moved after this, the entry will appear in the slot where the sheet cursor rested when editing began. So if for example you are constructing a formula to add together the contents of five slots in widely separated parts of the sheet, you can place the sheet cursor in the slot where you want the entry to appear, press the Space Bar to start editing, then move the sheet cursor around to find the slots you need, and type in the references. When you press **RETURN** the entry will appear in the selected slot.

This process is made even easier when you use the **SHIFT** **COPY** facility. In the example above, instead of typing in the slot references, you have only to place the cursor on the slots concerned, hold down **SHIFT** and press **COPY**. This copies the slot reference into the editing line.

Changing the slot format

The way in which numbers appear in slots is controlled by the slot formatting, which can be set to justify right or left, use a minus sign or brackets for negative numbers, and vary the number of decimal places.

To show how this works, choose a slot and position the cursor in it.

Type: `123.4567` **RETURN**

The slot will now show 123.457, assuming you are still using the default column width of seven characters. When more than seven are entered as a value, ViewSheet rounds the number, although the whole number is shown in the CONTENTS= line above.

Press: EDIT SLOT FORMAT (f6)
Reply: Format?
FRM
Type: D2RM **RETURN**

The slot will now show 123.46 because you have specified two decimal places (D2) and the system rounds upwards.

The letter R in the format definition stands for 'right' in the slot. Change it to L and the number is ranged left.

The M stands for minus. The point is that in normal arithmetic minus quantities are shown with a minus sign, but in accounting minus quantities are shown with brackets. ViewSheet allows for both possibilities by allowing you to replace the M with a B.

Try it, but bear in mind that it is possible to overload the slot by setting too many decimal places. For example, if you enter a number like −123.45, specify two decimal places (D2), and brackets (B), you are in effect telling ViewSheet to display the number (123.45) – which is eight characters. If your column is set to seven characters this is impossible, and ViewSheet will reply by showing a percentage sign in the slot although the formula will be shown in the CONTENTS= line above, and the value of this is the number that will be used in all calculations involving that slot. You can, of course, increase the width of the columns by resetting the window definition and the number would then be displayed.

Protection

Of course you should think twice before modifying a sheet. But there will always be those occasions when you are impatient to get some modifications done and only realise as some precious piece of data or long formula disappears that you should have deleted the slot in the next column instead.

To avoid such problems ViewSheet has the facility to protect rows and columns.

Protection prevents you from destroying slots, columns and rows through typing over them, deleting them, or replicating over them.

To use it simply move the cursor into the right row or column and press PROTECT ROW (**SHIFT** *f*6) or PROTECT COLUMN (**SHIFT** *f*5). As a signal that the row or column is protected the dots in the border are replaced by a continuous line. Thus `......1` is replaced by `_______1`. (As usual this assumes you are in `MODE 3`. In `MODE 7` you will see dashes.)

Protection is especially useful when you are preparing a sheet for other people to fill in. In such cases you will probably want to protect all your formulae and results, and only leave those parts unprotected where people have to enter the new information.

To remove protection from a row or column, simply press the same key again. The line will be replaced by dots and the protection will be removed.

To disable all protection throughout the sheet, enter command mode and type:

```
PROTECT OFF RETURN
```

To restore protection, enter command mode and type:

```
PROTECT ON RETURN
```

It is worth remembering that when you switch ViewSheet on, protection is already enabled, even though you have not yet made use of it. So if you wish to delete rows or columns at any stage you must first switch to command mode and disable protection.

If you do not know whether protection is on or off, switch to command mode and type:

```
PROTECT RETURN
```

Inserting and deleting rows and columns

When modifying sheets you quite often come to the point where you realise that things would be much more convenient if you could insert a row or column.

To insert a row place the cursor on the row below the point where you want the new row inserted and press INSERT ROW (**SHIFT** *f*2).

To insert a column place the cursor in the column to the right of the point where you want the new column inserted and press INSERT COLUMN (**SHIFT** *f*1).

In both cases the new column or row appears correctly numbered, and all slot references in formulae which are affected by the insertion are automatically modified.

This is a valuable feature. To take a very simple example, suppose you enter the value `123` in slot A1 and the value `456` in slot A2 and then the formula `A1+A2` in slot A3. You then wish to insert a row between A1 and A2.

Place the cursor on row 2.
Press: INSERT ROW (**SHIFT** f_2)

The new row will be inserted; the existing rows will be renumbered; and the formula in what is now slot A4 will change from `A1+A2` into `A1+A3`.

If you have entered a lot of material in your sheet the process of modifying headers and formulae may take some time; notice the processing signal – the moving dot – at top left while this process is going on.

Deleting rows and columns is done in much the same way, but as mentioned above, ViewSheet effectively asks you to think before you act. Even if you have not specifically protected the row or column concerned, ViewSheet assumes that it is protected against deletion. So before deleting you must remove protection, by entering command mode and typing: `PROTECT OFF` **RETURN**.

When you have done this you can delete rows with **CTRL** f_2 and columns with **CTRL** f_1. Remember to turn protection back on afterwards.

7 Using discs and cassettes

When you have set up a few useful displays with ViewSheet you will naturally think of modifying them for other purposes. Or you may find that one of your displays is most useful to you if you can store data in it and add to it every week or every month, as new statistics become available. Either way you will want to keep displays and data for future use.

ViewSheet allows you:

- To save a whole sheet onto disc or cassette.
- To save the window definitions only.
- (With discs) to save selected data from a sheet.

Saving a sheet

The `SAVE` command saves everything about the sheet, including slot contents, window definitions, column and row headings, and protection of columns and rows.

Cassette

Enter command mode by pressing **ESCAPE**.
Place a cassette in the recorder.
Type: `SAVE filename` **RETURN**
Reply: `RECORD THEN RETURN`

Press RECORD on the cassette recorder and **RETURN** on the computer. When the prompt (=>) returns, the file is recorded.

If your cassette recorder has no motor control (REM) socket, stop it quickly. If you are using the REM connection, it will stop automatically. You may use the abbreviation `S` instead of `SAVE`.

Disc

Enter command mode by pressing **ESCAPE**.
Place a disc in the drive.
Type: `SAVE filename` **RETURN**

You may use the abbreviation `S` instead of `SAVE`.

Saving the file named in the command mode heading

When you load a file from disc or cassette, as described below, the name of the file is shown in command mode in the form: `Editing (filename)`.

You may save this file under the name shown by typing: `SAVE` **RETURN**

If you wish to change the name given in the command mode heading, type:

`NAME filename` **RETURN**

This name will then appear in the heading and you can use it to `SAVE` as described above.

Loading a sheet

Cassette

Enter command mode by pressing **ESCAPE**.
Place the appropriate cassette in the recorder.
Type: `LOAD filename`
Press PLAY on the cassette recorder.
Press: **RETURN**

Depending on the cassette recorder you may hear high- or low-pitched sounds. When the prompt (=>) returns, the file has been loaded.

You may use the abbreviation `L` instead of `LOAD`.

Disc

Enter command mode by pressing **ESCAPE**.
Place the appropriate disc in the drive.
Type: `LOAD filename` **RETURN**

You may use the abbreviation `L` instead of `LOAD`.

Window definitions

Provided you are using the same screen mode as you did when you saved the file, all window definitions will be loaded along with the headings and slot contents. If you are not, the window definitions will be reset to default values.

Saving window definitions only

ViewSheet allows you to save and load windows quite independently of the sheet they apply to. In this way you can use several sets of windows in succession on the same sheet, without affecting any of the data in it.

The procedure for saving and loading windows is the same as for saving and loading complete sheets, except that the commands are `SW` and `LW` respectively. Suppose you decide to save a set of window definitions under the name 'wind2'.

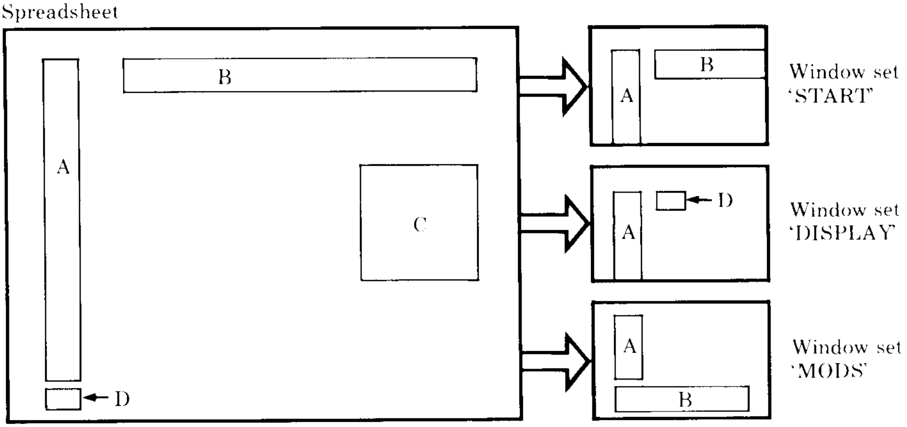
To save windows type: `SW wind2` **RETURN**
To load windows type: `LW wind2` **RETURN**

As an example of the use of this facility, suppose you have made up a complex sheet to your satisfaction, and then want to organise its display on the screen with windows.

The first thing to do is to save the sheet just as it is with the default window 0. Then make up a set of windows and save them too. If you want an alternative set of windows which will display the sheet in a different way, you can then reload the original sheet and set up the next set of windows.

Having done all this, you will have the original sheet on one file, and the two sets of windows on additional files. The diagram below illustrates the idea. Here there would be four files:

- The original sheet.
- Window set 'START'.
- Window set 'DISPLAY'.
- Window set 'MODS'.



By loading the original sheet, followed by one of the windows if required, it would be possible to use the same spreadsheet in four different ways.

Saving and loading data within spreadsheets (discs only)

A problem which is often encountered with spreadsheets is the difficulty of transporting data from one sheet to another.

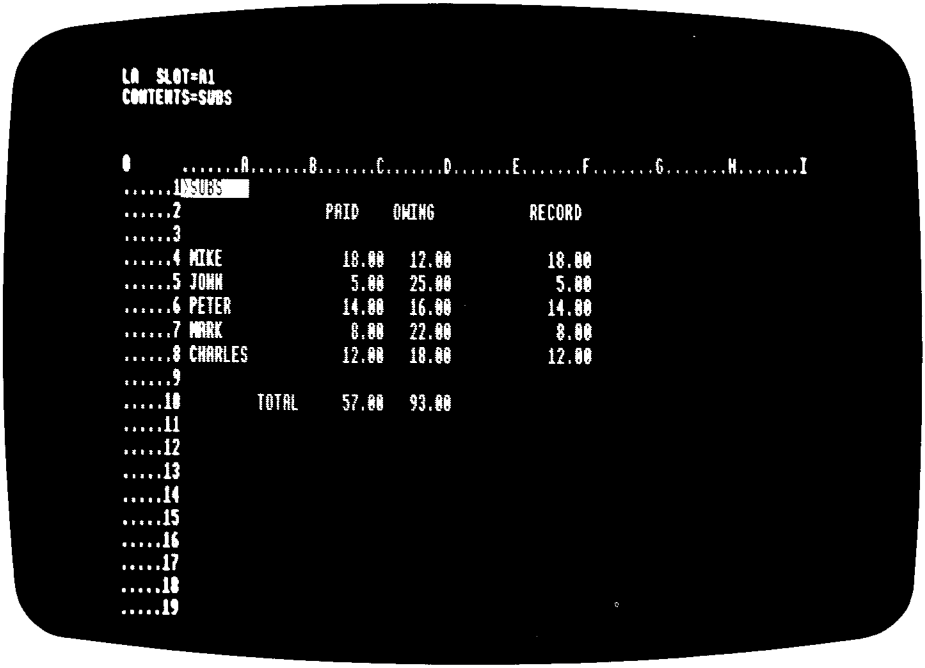
For example, suppose you are compiling monthly statistics. You have set up formulae in your sheet to process the mass of information you are entering, and out at the bottom drop the results you want. So far so good, but what if you want to update the current month's statistics next month? You need to 'carry forward' those results to start you off again. Of course you can print them out, set up a blank sheet for next month, and laboriously copy them all. But that involves a high risk of error.

ViewSheet solves this problem with a facility which enables you to set up the first sheet in such a way that your monthly results will automatically be entered in a special file on the disc. When you start off the next month's sheet, these same results will be taken from that file and entered in the sheet, automatically and without error.

Naturally this is a rather more advanced technique than any described so far, but it can be extremely valuable. The method is described in detail in Part Two, but the following example illustrates all the principles involved.

Example

The accounts of the club are worked out on ViewSheet. Part of its income is from subscriptions, and the subscription account looks like this. (Notice that the format `D2RM` has been set, so as to produce the two decimal places required in accounting.)



```
LA  SLOT=A1
CONTENTS=SUBS

0     .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1 SUBS
......2                 PAID    OWING          RECORD
......3
......4 MIKE             18.00   12.00           18.00
......5 JOHN              5.00   25.00            5.00
......6 PETER            14.00   16.00           14.00
......7 MARK              8.00   22.00            8.00
......8 CHARLES          12.00   18.00           12.00
......9
.....10         TOTAL    57.00   93.00
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

The subscription is £30 a year, and the treasurer sets up the sheet so that he has only to update the amount each member has paid that year, and the totals are updated automatically. By now you will realise how this is done:

- The slots in the `PAID` column are just numbers.
- The slots in the `OWING` column contain formulae: `30-C4` for example (£30 being the club's subscription).
- The totals sum the ranges above them: `C4C8` for example.

The labels and values entered in the sheet would therefore be as follows:

```
0       .......A.......B.......C.......D.......E.......F.....
......1 SUBS
......2                  PAID  OWING        RECORD
......3
......4 MIKE            18.00  30-C4
......5 JOHN             5.00  30-C5
......6 PETER           14.00  30-C6
......7 MARK             8.00  30-C7
......8 CHARLES         12.00  30-C8
......9
.....10         TOTAL    C4C8   D4D8
.....11
.....12
.....13
```

Before setting to work on entering the values we need to define the slot formats into which they should go. Since this is an account, all amounts should have two decimal places so that the numbers are ranged under each other with the decimal points in line.

We dealt earlier with EDIT SLOT FORMAT which can change the format of a slot in this way. In this case we need to change the format of all the slots in the window – like this:

Press: EDIT WINDOW
Reply: `Window?`
Press: `0` **RETURN**
Reply:

```
Wi     TopL   BotR   Pos    Cw     Bw     Fmt    Opt
0      A1     I19           7      7      FRM
```

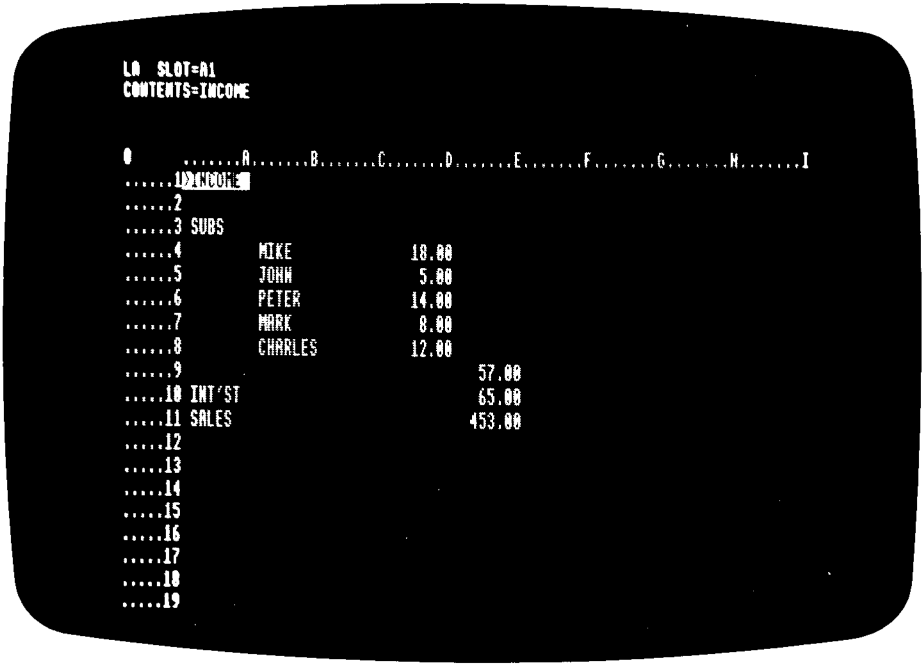
In order to give each number its required two decimal places you need to change the `Fmt` or format. We have already described how to do this in the last chapter. Run the cursor along (holding down **CTRL** and using the arrow keys as usual) and change it from `FRM` to `D2RM`. When you press **RETURN** all numbers in the sheet will acquire their two decimal places.

Having adjusted the format, enter the labels and values into the sheet and check it against the illustrations.

When you have got the sheet working correctly, with the exception of the `RECORD` column which is not needed yet, save it onto the disc.

Income

The club's income is on a separate sheet from subscriptions, as shown below.



```
L0  SLOT=A1
CONTENTS=INCOME


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1>INCOME
......2
......3 SUBS
......4         MIKE       18.00
......5         JOHN        5.00
......6         PETER      14.00
......7         MARK        8.00
......8         CHARLES    12.00
......9                            57.00
.....10 INT'ST                     65.00
.....11 SALES                     453.00
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

(Don't try to set this up on the computer just yet – the slot contents are not quite what they seem.)

What we need now is a means of getting the data from the subscription sheet into slots C4 to C8 of the income sheet. This is where the **RECORD** column of the subscription sheet comes in.

The method of carrying out this transfer between sheets relies on a special kind of disc file. If you are familiar with computing methods you will know it as a 'two-dimensional array'. If not, imagine it as something like a small ViewSheet. The two-dimensional array below is numbered one to six in both directions, and the location of any slot in it can be specified by quoting the number across followed by the number down. So for example the third slot in the second row is slot 3, 2.

	1	2	3	4	5	6
1	18					
2	5		*			
3	14					
4	8					
5	12					
6						

ViewSheet allows us to create a file on the disc containing an array of this kind, and to place data in the elements (ie slots) of the array. Since the array file is quite independent of any sheet, we can read this same information into another sheet and use it as we would any other values in that sheet.

Creating the array file

Before entering data in a file, you have to *create* the file. The command `CREATE` sets up a file like the one illustrated above. This happens to contain 36 elements in a 6 by 6 format, but any size up to 10710 elements (eg 255 by 42) can be used. For our present purposes, however, 6 by 6 is ample. To set this up, get into command mode by pressing **ESCAPE** ; then type:

```
CREATE 1 6 6 RETURN
```

If you check the list of files on the disc (by typing `*.` **RETURN**) you should now find one called `V.VS1`. ViewSheet array files are always named in this way to help you distinguish them, but when creating them you only type the number.

Writing to the file

We are now ready to make up the `RECORD` column in the subscription sheet, which will write the information needed for the income sheet into the array file.

Load the subscription sheet into memory again. Switch to sheet mode by pressing **ESCAPE** . Then place the following entries in column F (the `RECORD` column) of the subscription account.

```
F4  –  WRITE(1,1,1,C4)
F5  –  WRITE(1,1,2,C5)
F6  –  WRITE(1,1,3,C6)
F7  –  WRITE(1,1,4,C7)
F8  –  WRITE(1,1,5,C8)
```

(Once you have become familiar with ViewSheet practices you will use the `REPLICATE` command for such entries as these – and in this case the `ROW` function would be useful too.)

The disc will spin as you enter the first of these commands, and when you finally press **ESCAPE** to return to command mode it will spin again, as the data are written into the file `V.VS1`.

The numbers in brackets in the command are:

- The array file number (ie `V.VS1`).
- The array element (the second and third numbers).
- The ViewSheet slot reference or expression.

Now `SAVE` the new version of the subscription sheet, so that you can experiment with it again to show the effect of using files within a sheet.

Reading from the file

To `READ` from the file, make up an entirely new sheet for the club's income, as shown earlier. (Don't forget to edit window 0 and reset the slot format to provide two decimal places for all values.)

Note particularly the entries for slots C4 to C8. These `READ` back the contents of the array elements in file `V.VS1`.

```
0         .......A.......B.......C.......D.......E.......F.....
......1 INCOME
......2
......3 SUBS
......4          MIKE          ←--------------------READ(1,1,1)
......5          JOHN          ←--------------------READ(1,1,2)
......6          PETER         ←--------------------READ(1,1,3)
......7          MARK          ←--------------------READ(1,1,4)
......8          CHARLES       ←--------------------READ(1,1,5)
......9                             C4C8
.....10 INT'ST                     65.00
.....11 SALES                     453.00
.....12
.....13
```

Again when you enter the first of the `READ` commands, the disc spins, picking up information from the array file `V.VS1`. As you enter the rest of the `READ` commands, the subscription amounts appear in the slots.

It is when the sheet recalculates that the reading from the disc file takes place. To test this, press RECALCULATE (function key **SHIFT** **f7**) and the disc will spin. Of course no change will occur in the sheet this time, since the amounts have already been read in.

Do not forget to save your income sheet too, if you want to carry out further experiments to get yourself familiar with these disc techniques.

8 Displays with bar charts

This chapter is devoted mainly to an example which illustrates ViewSheet's bar chart facility – partly because this is exceptionally useful in itself, and partly because making up a bar chart display involves a combination of several of the techniques we have been discussing.

Bar charts in ViewSheet are quite straightforward: if you edit a window and place the letter `C` in the `Opt` column, all the numbers in that window are replaced by asterisks. The scale is one to one, so the value 22 is replaced by 22 asterisks for example.

So one way of using bar charts in ViewSheet would be to set up a window consisting of just one column and select the bar chart option (`C`) for it. This would convert the column into a horizontal bar chart. Of course it is necessary to have slots wider than seven characters to make much of a chart, but this too can be arranged as you edit the window.

To show how the bar chart option works, clear the sheet by entering command mode and typing `NEW`. Select `MODE 3` and return to sheet mode.

Enter a column of figures in column A, choosing numbers less than 60. Now set up the windows:

Press: EDIT WINDOW (**f1**)
Reply: `Window?`
Press: `0` **RETURN**
Reply:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
0     A1    I19         7     7     FRM
```

Change this to:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
0     A1    A19         7     7     FRM   TS
```

When you press **RETURN** you should see only the column of figures (since you have turned off top and side borders) in the top left of the screen.

Now to set up another window containing the bar chart beside this. We can in fact use the column A1 to A19 again for this window too. (It may seem odd to have two windows looking on to the same part of the sheet, but ViewSheet can do that, and it is just what we need.)

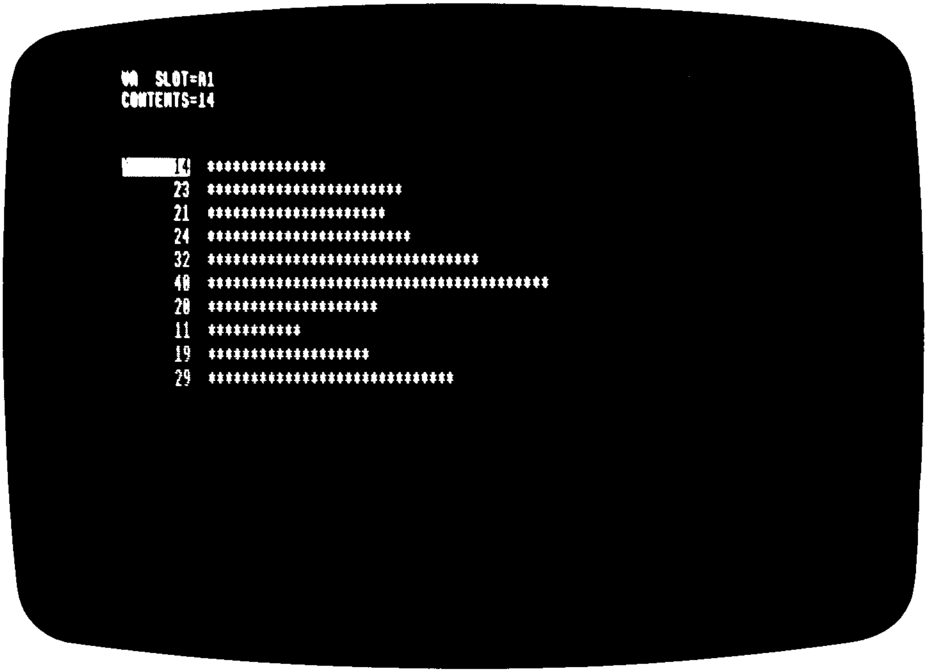
This will be window 1, and we shall place it to the right of window 0, widening the slots (`Cw`) to make room for the bar chart, and selecting the chart option (`C`).

Press: EDIT WINDOW (**f1**)
Reply: `Window?`
Press: `1` **RETURN**

Change the window definition to:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
1     A1    A19   R0    60    7     FRM   TSC
```

When you press **RETURN** you should find yourself looking at something like the following. (If it does not work, re-edit the windows, comparing the definitions carefully with those given above.)



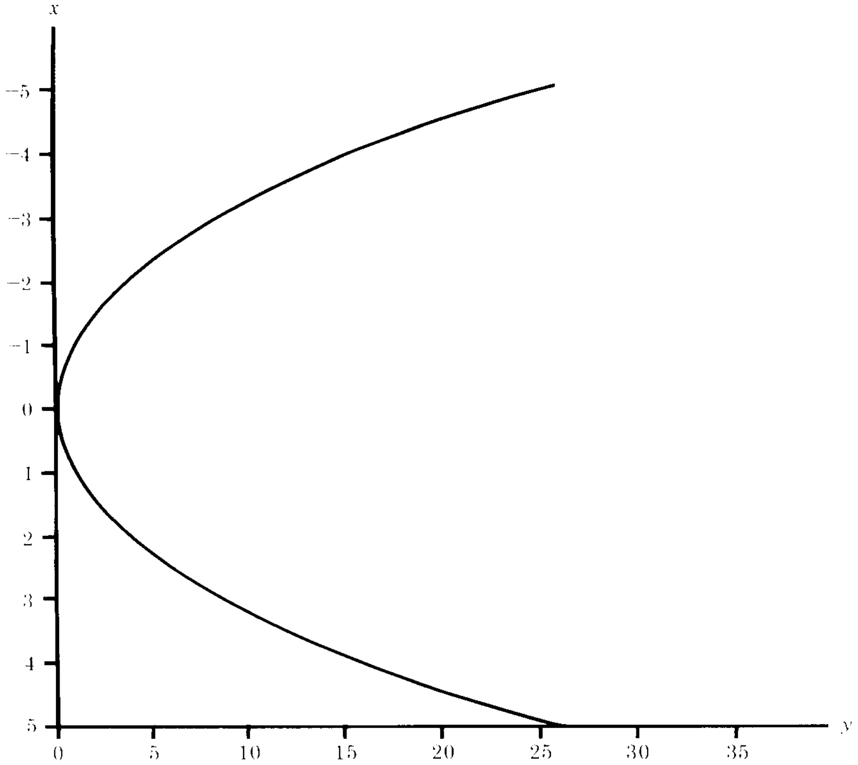
```
W0  SLOT=A1
CONTENTS=14


      14 **************
      23 ***********************
      21 *********************
      24 ************************
      32 ********************************
      40 ****************************************
      20 ********************
      11 ***********
      19 *******************
      29 *****************************
```

Since both the numbers and the asterisks represent the same column of numbers on the sheet, you can show the effect of the bar chart facility by moving the cursor and changing some of the numbers. The sheet will recalculate and the line of asterisks will change accordingly.

The following demonstration is of a slightly more mathematical kind, but even if you know little mathematics, the demonstration itself should make all clear.

The curve on the graph below is a parabola. The formula for such a curve is usually expressed as:

$y = ax^2$



(Conventionally the *x*-axis is the horizontal one. We have turned the graph on its side so that it will match the ViewSheet bar chart.)

The graph is plotted by first writing down the values for x along one axis, and those for y along the other. You then take each of the x values and fit it into the formula, so finding a value for y for each value of x. Each of these sets of values of x and y together represent a point. When each point is joined to the next the result is a line: a parabola.

The missing quantity in the formula is a. In fact the value of a controls the size of the parabola, as you will see if you follow the instructions below.

The aim is to construct a bar chart in which the ends of the bars describe a parabola, but which is instantly adjustable by changing the value of a. The result is a kind of dynamic bar chart which shows a good deal more about the relationships it displays than is ever possible with charts drawn on paper.

First clear the sheet and set up the following values for x in column A. Start at A3 and go on to A13.

A3	–5
A4	–4
A5	–3
A6	–2
A7	–1
A8	0
A9	1
A10	2
A11	3
A12	4
A13	5

Now move the cursor to slot A1 and enter the value 5. This will represent a in the formula.

Now to enter the first formula in slot B3
The formula is:

```
A3^2*A1/4+1
```

Note that ^ means 'to the power'.

The idea is to create the y values in column B. The formula squares the number in A3 and multiplies it by the number in A1 (the a value). To keep the numbers small enough for us to use in a ViewSheet bar chart the result is then divided by four and to prevent the chart from going down to zero (and so showing no asterisks at all) we add one.

The value shown at B3 should be `32.25`. If it is anything else you have made a mistake. Check the figures in column A and if they are all correct, check the formula by placing the cursor on B3 and looking at the contents line. If it is wrong, edit it again by pressing **COPY** .

Now to replicate the formula through to slot B13.

Press:	REPLICATE (**f0**)
Reply:	`From - To?`
Type:	`B3-B4B13` **RETURN**

Reply:	`R)elative, N)o change?`
	`A3^2*A1/4+1`

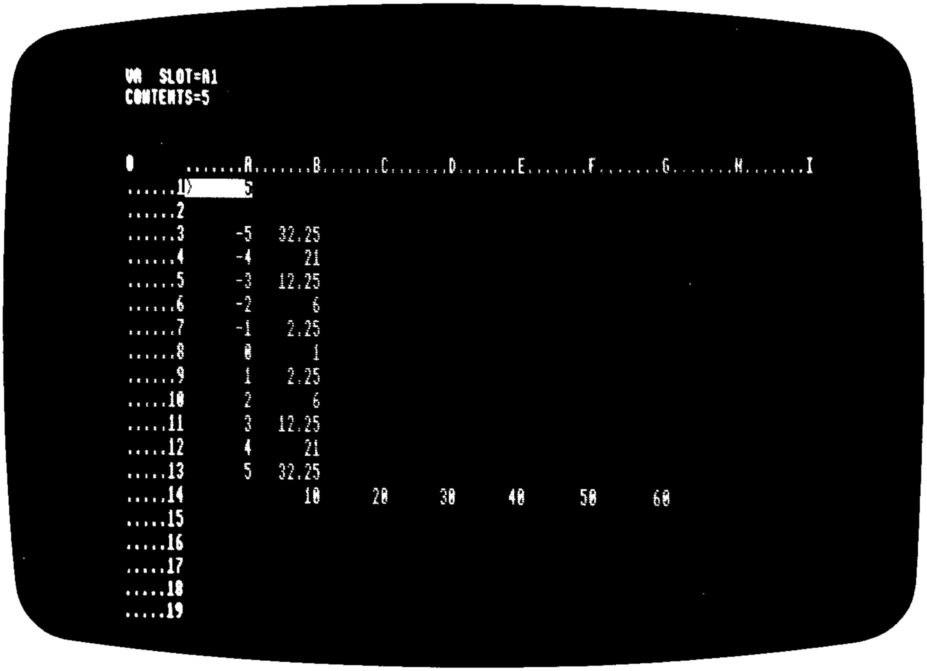
Press:	`R`

Reply:	`R)elative, N)o change?`
	`A1/4+1`

Press:	`N`

(It is `N` the second time since we wish to multiply all numbers by the value in A1.)

You should now be looking at columns A and B as shown below:



```
VM  SLOT=A1
CONTENTS=5


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1)      5
......2
......3      -5   32.25
......4      -4      21
......5      -3   12.25
......6      -2       6
......7      -1    2.25
......8       0       1
......9       1    2.25
.....10       2       6
.....11       3   12.25
.....12       4      21
.....13       5   32.25
.....14              10      20      30      40      50      60
.....15
.....16
.....17
.....18
.....19
```

The numbers in row 14 are to be the *y*-axis of the chart; they are not strictly necessary, but if you enter them in slots B14 to G14 you can use them to check the accuracy of the bar chart.

We now have three distinct parts of the sheet which we can combine using the window facility into a screen display:

- The value in A1 and the *x*-axis numbers in A3 to A13.
- The *y*-axis numbers in row B14 to G14.
- The numbers in B3 to B13 which are to be converted into asterisks.

Now set up the three windows as follows:

Window 0:

```
Wi     TopL   BotR   Pos    Cw     Bw     Fmt    Opt
0      A1     A13           7      0      FRM    TS
```

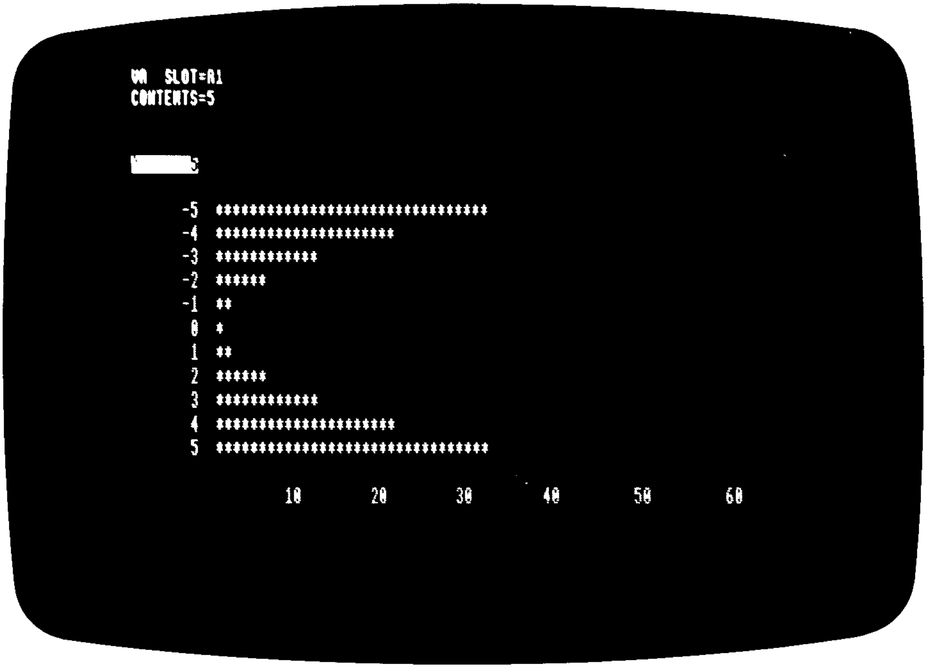
Window 1:

```
Wi     TopL   BotR   Pos    Cw     Bw     Fmt    Opt
1      B1     B13    R0     60     0      FRM    TSC
```

Window 2:

```
Wi     TopL   BotR   Pos    Cw     Bw     Fmt    Opt
2      A14    G14    B0     9      0      FRM    TS
```

If you have entered everything correctly you should now be looking at this:



```
VA  SLOT=A1
CONTENTS=5


     5

     -5 ********************************
     -4 *********************
     -3 ************
     -2 ******
     -1 **
      0 *
      1 **
      2 ******
      3 ************
      4 *********************
      5 ********************************

                 10        20        30        40        50        60
```

The shape described by the ends of the bars is recognisably a parabola. To show the effect of different *a* values on the shape, change the figure 5 at the top to any number from 0 to 10.

If you understand what is happening in this display you are ready to go ahead and make up your own sheets and displays. If not you should look at each part of the sheet again and see how the numbers and formulae relate to each other.

9 Conditions and lookup

So far in this book the sheets given as examples have used relatively simple relations between numbers – adding them together, adding ranges, combining them with numbers in simple formulae, and so on.

ViewSheet also allows you to do such things as find a number which you know exists somewhere in a range; take the average, maximum or minimum of a list of slots or a range; or cause the contents of a slot to be determined by whether a number is greater than another or not.

There are several operators of this kind, which are concerned with whether numbers are greater or less than, or equal to others. These are the *conditional operators*:

=	equal to
<>	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

These conditional operators can be used to test conditions by combining them with the `IF` statement.

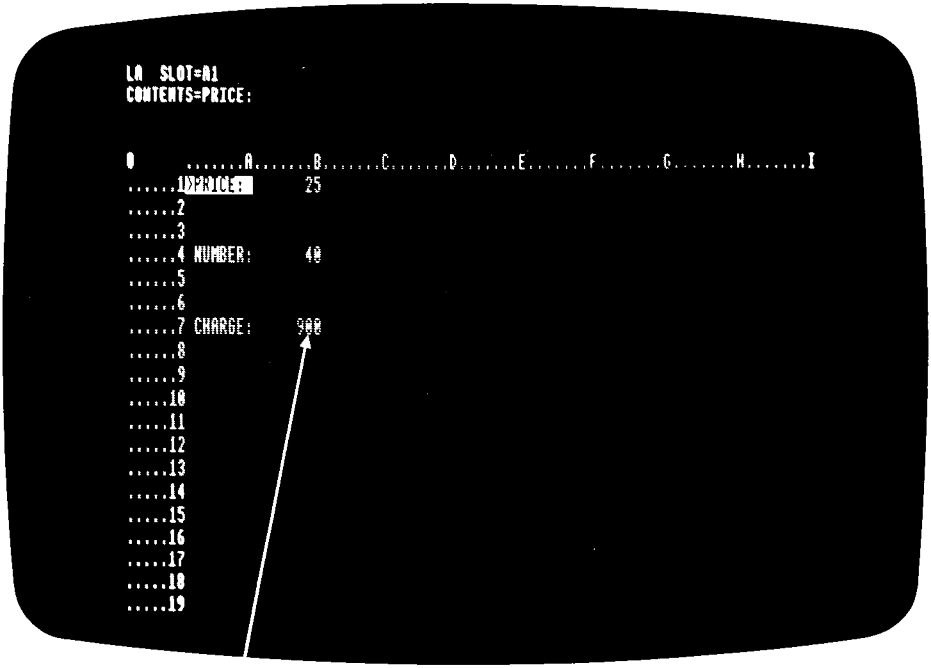
For example, suppose we are entering orders for products on a sheet, and we have a rule that all quantities over 20 are subject to a discount of 10 per cent. Enter the following on your sheet to demonstrate the calculation involved:

Slot	**Content**
A1	`PRICE:`
B1	`25`
A4	`NUMBER:`
B4	`40`
A7	`CHARGE:`
B7	`IF(B4>20,B1*0.9*B4,B1*B4)`

The idea is that the price is shown in B1; the quantity sold in B4 and the total charge to be made for that quantity in B7. The `IF` statement in B7 is in three parts:

```
IF(condition,result1,result2)
```

If the condition is true the first result is shown; if false the second. The condition is that B4 should be more than 20 – ie the order is for more than 20. If this is not true, then the total charge must be the price multiplied by the number ordered: `B1*B4`. If it is true, then the price must be reduced by 10 per cent, ie `B1*0.9*B4`.



```
LA  SLOT=A1
CONTENTS=PRICE:


0       .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1>PRICE:        25
......2
......3
......4 NUMBER:       40
......5
......6
......7 CHARGE:      900
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

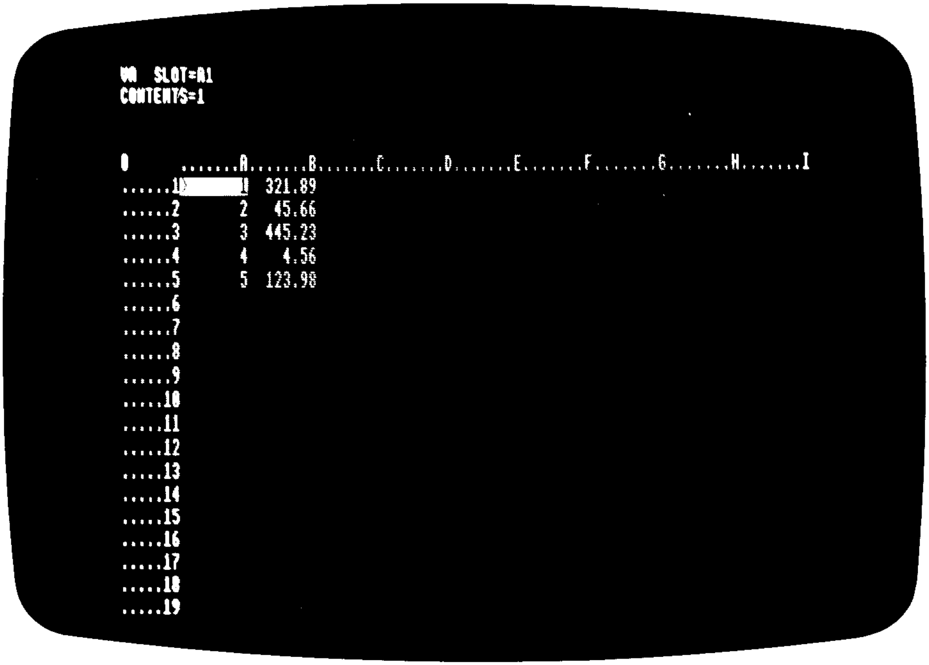
```
IF(B4>20,B1*0.9*B4,B1*B4)
```

You can test the sheet by typing different values into B4.

LOOKUP

Another ViewSheet facility for relating numbers is `LOOKUP`. What it does is to compare a specific value with the values in a range (the lookup range). When it finds an equal value it locates the corresponding value in another named range (the result range), which becomes the value of the `LOOKUP` function.

Put like this it may not seem very useful. In fact it can be used as a source of data in a sheet. For example, suppose we have a list of numbers like this:

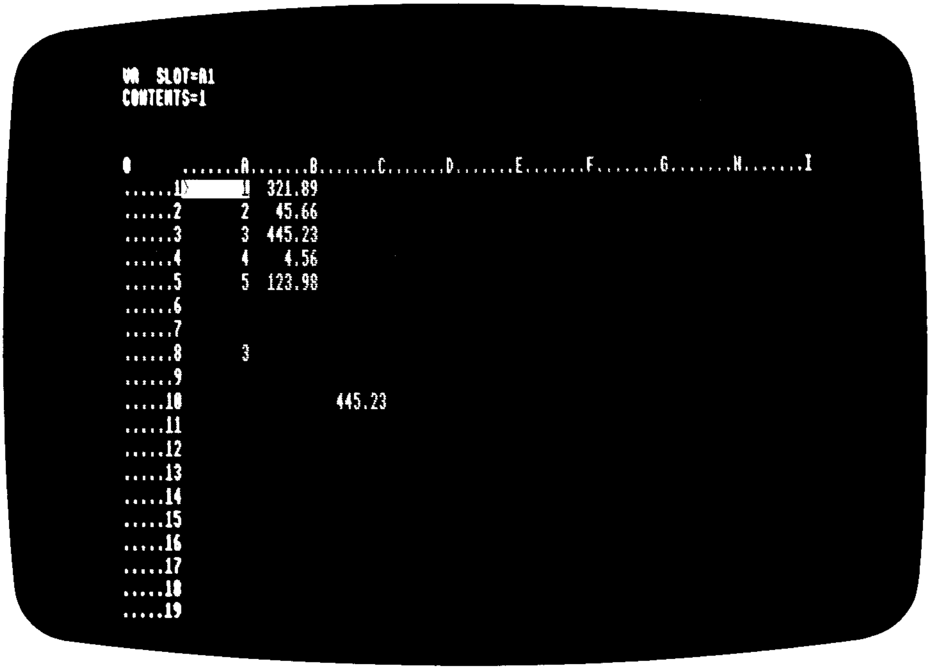


```
VA  SLOT=A1
CONTENTS=1

0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1        1  321.89
......2        2   45.66
......3        3  445.23
......4        4    4.56
......5        5  123.98
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

You can use the `LOOKUP` function to specify which of the numbers in the list you will use in a calculation, using the numbers 1 to 5 as a kind of code. For example, enter in slot C10 the following:

```
LOOKUP(A8,A1A5,B1B5)
```

Then enter any number from 1 to 5 in slot A8 and the corresponding number in the range in column B will appear in slot C10.

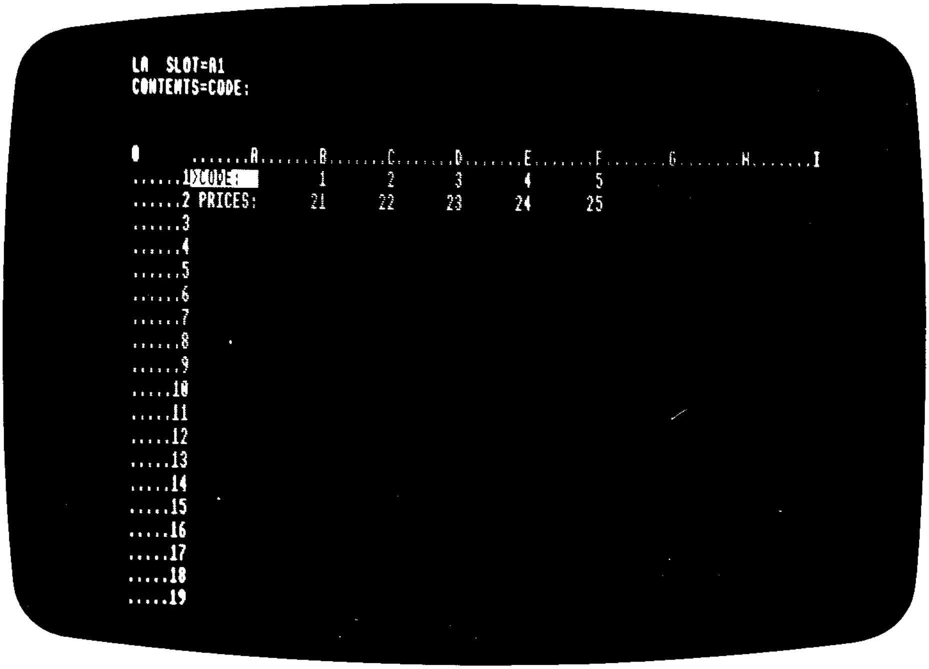


```
VR  SLOT=A1
CONTENTS=1


0       .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1        1  321.89
......2        2   45.66
......3        3  445.23
......4        4    4.56
......5        5  123.98
......6
......7
......8        3
......9
.....10                445.23
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

The standard layout for the `LOOKUP` formula is:

```
LOOKUP(value,lookup range,result range)
```

By way of a simple example we could modify the previous sheet involving a discount price. Instead of a 10 per cent discount we could have a price structure, with five prices, any one of which we could select for a given customer, like this:



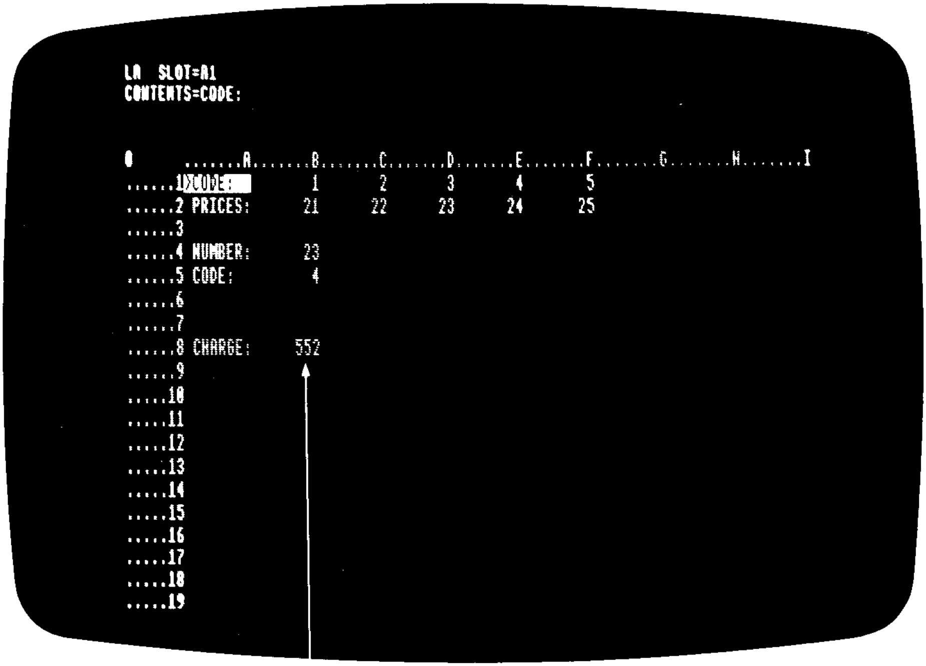
```
LA  SLOT=A1
CONTENTS=CODE:


0     .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1 CODE:        1       2       3       4       5
......2 PRICES:     21      22      23      24      25
......3
......4
......5
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

This gives us a list of codes and prices. We can specify the quantity in another line as before, and in another line still the price code which applies to that deal. Finally the `CHARGE` slot has to look up the price according to the code and multiply this by the number ordered.

Type in entries as follows:

Slot	**Contents**
A4	`NUMBER:`
A5	`CODE:`
A8	`CHARGE`
B8	`LOOKUP(B5,B1F1,B2F2)*B4`

Enter values for the number ordered (in B4) and the price code (in B5) and the sheet should look something like this:



```
LA  SLOT=A1
CONTENTS=CODE:

0     .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1 CODE:        1       2       3       4       5
......2 PRICES:     21      22      23      24      25
......3
......4 NUMBER:     23
......5 CODE:        4
......6
......7
......8 CHARGE:    552
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

LOOKUP(B5,B1F1,B2F2)*B4

Example – orders and statistics

The following more extensive example shows how many of the facilities described in this chapter can be combined to provide 'instant statistics'.

Suppose we have a company selling computer games to dealers. Each product has its code and price, and we need to know how many games of each kind are bought by each dealer. We also want to total our sales and analyse them according to how many of each game are sold, and what the average and maximum orders are.

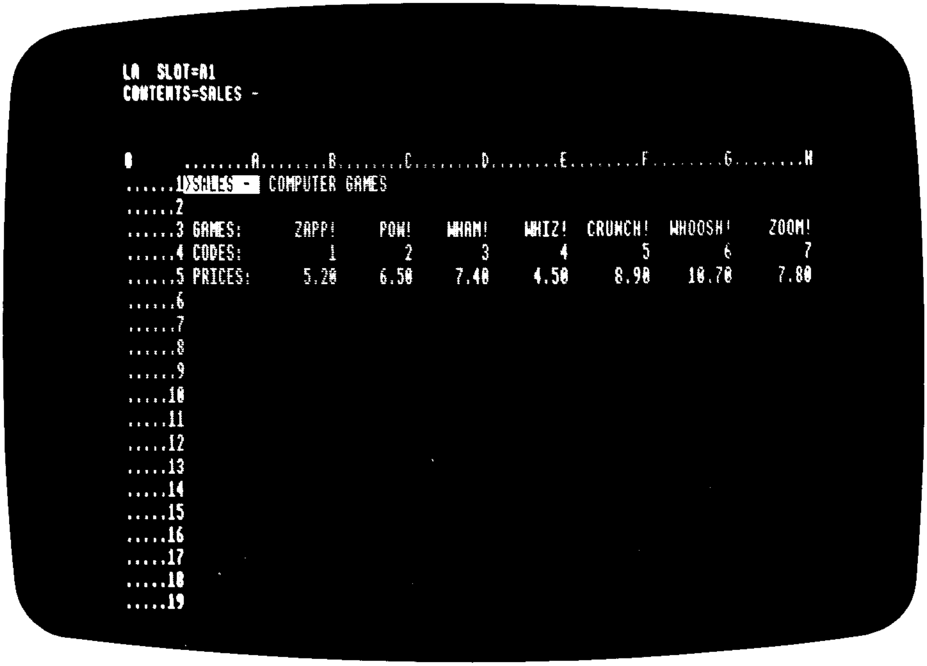
Products and prices

We can start by setting up our product prices and codes, much as in the last example. We are placing it at the top of the sheet so that when the sheet recalculates the basic information is read first and used below.

Before doing so, you will need to edit window 0 to increase the column width to eight character spaces, to accommodate words such as `COMPUTER`:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
0     A1    I19         8     7     FRM
```

You can then fill in the names of the games like this:



```
LA  SLOT=A1
CONTENTS=SALES -

0      ........A........B........C........D........E........F........G........H
......1>SALES - COMPUTER GAMES
......2
......3 GAMES:     ZAPP!     POW!    WHAM!    WHIZ!  CRUNCH!  WHOOSH!    ZOOM!
......4 CODES:         1        2        3        4        5        6        7
......5 PRICES:     5.20     6.50     7.40     4.50     8.90    10.70     7.80
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

You should adjust all the games titles to justify right if you want your display to look like the one above, ie with the codes and prices aligned with the right hand end of the title. To justify a label to the right place the cursor in the slot involved and press JUSTIFY LABEL (SHIFT f8).

The codes can conveniently be entered by using the `COL` function.

Move the sheet cursor to B4
Type: `COL-1` **RETURN**
The value 1 should now show in slot B4

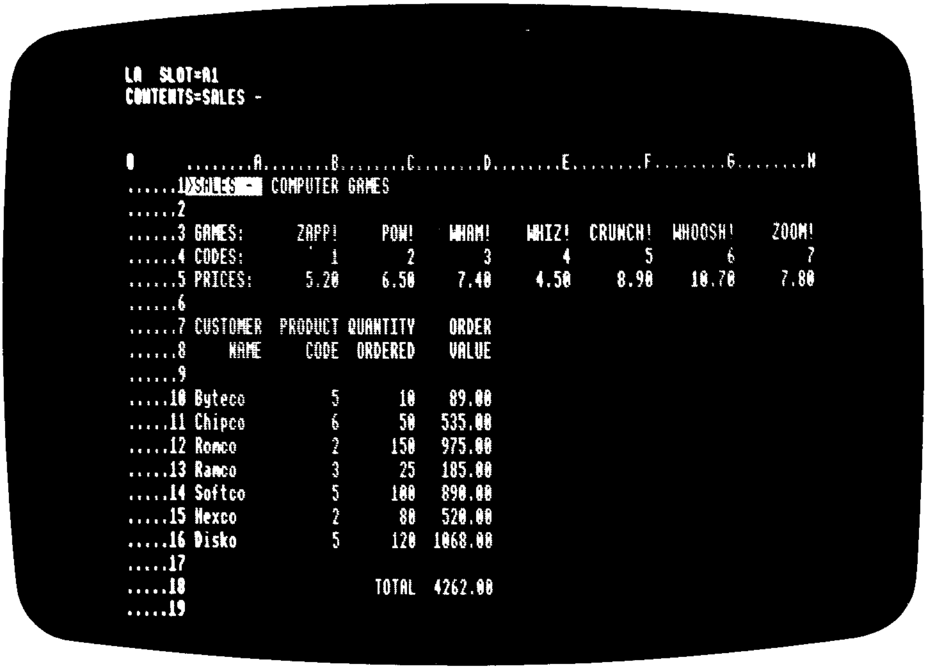
Press: REPLICATE (f0)
Reply: `From - To?`
Type: `B4-C4H4` **RETURN**

The values 1 to 7 should now show in columns B to H

If you attempt to enter the prices as shown above you will find that the computer misses out the final zeros. To display two decimal places you should edit the slot format for each of the seven slots involved. Press EDIT SLOT FORMAT (f6) and change `FRM` to `D2RM`.

Customers and orders

The next step is to set up columns for customers, the codes of the products they order, the quantities they order, and the value of the order, as shown below:



```
LA  SLOT=A1
CONTENTS=SALES -


█      ........A........B........C........D........E........F........G........H
......1)SALES -  COMPUTER GAMES
......2
......3 GAMES:        ZAPP!      POW!     WHAM!    WHIZ!  CRUNCH!  WHOOSH!    ZOOM!
......4 CODES:            1         2         3        4        5        6        7
......5 PRICES:        5.20      6.50      7.40     4.50     8.90    10.70     7.80
......6
......7 CUSTOMER  PRODUCT QUANTITY    ORDER
......8     NAME     CODE  ORDERED    VALUE
......9
.....10 Byteco          5       10    89.00
.....11 Chipco          6       50   535.00
.....12 Romco           2      150   975.00
.....13 Ramco           3       25   185.00
.....14 Softco          5      100   890.00
.....15 Hexco           2       80   520.00
.....16 Disko           5      120  1068.00
.....17
.....18                      TOTAL  4262.00
.....19
```

You should enter slots as follows:

Slot	**Content**
A7	`CUSTOMER`
A8	`NAME` (right justified)
B7	`PRODUCT` (right justified)
B8	`CODE` (right justified)
C7	`QUANTITY`
C8	`ORDERED` (right justified)
D7	`ORDER` (right justified)
D8	`VALUE` (right justified)
A10 to A16	Customer names
B10 to B16	Numbers from 1 to 7 representing product codes – these can be left blank for the moment
C10 to C16	Numbers to represent the quantities ordered – these can be left blank for the moment
D10	`C10*LOOKUP(B10,B4Z4,B5Z5)`

If you have not entered a product code in B10 you will find that an error is signalled in D10. This is because ViewSheet has interpreted the blank in C10 as zero and has tried to find a zero in the lookup range and failed.

Notice that we have set the lookup range as B4Z4 even though we have only entered products so far in columns A to H.

The next step is to replicate the contents of D10 into D11 to D16, but before doing so you should change the format of the slot to `D2RM`, so that when it is replicated the format is replicated too. To replicate:

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `D10-D11D16` **RETURN**
Reply: `R)elative, N)o change?`
`C10*LOOKUP(B10,B4Z4,B5Z5)`

Press `R` to replicate slot references C10 and B10 relatively, and `N` to replicate slot references B4, Z4, B5 and Z5 without change.

The result should be:

Slot	**Content**
D10	`C10*LOOKUP(B10,B4Z4,B5Z5)`
D11	`C11*LOOKUP(B11,B4Z4,B5Z5)`
D12	`C12*LOOKUP(B12,B4Z4,B5Z5)`
D13	`C13*LOOKUP(B13,B4Z4,B5Z5)`
D14	`C14*LOOKUP(B14,B4Z4,B5Z5)`
D15	`C15*LOOKUP(B15,B4Z4,B5Z5)`
D16	`C16*LOOKUP(B16,B4Z4,B5Z5)`

You can now add:

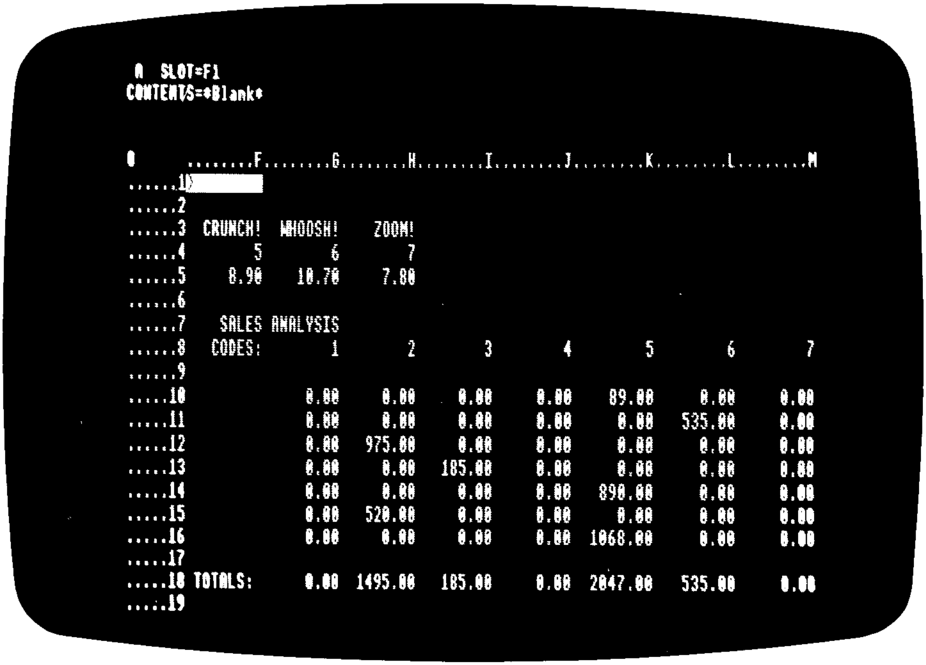
C18	`TOTAL` (right justified)
D18	`D10D16` (slot format `D2RM`)

If you now enter numbers in columns B and C ViewSheet will automatically look up the prices and work out the totals for the orders. The advantage of looking up the prices in this way instead of writing the prices into the sheet at every slot where they are required is that when you want to alter a price you have only to alter it on row 4 at the top and the effects of this will be recalculated throughout the rest of the sheet.

Analysis

Finally why not use ViewSheet's facilities to provide an analysis of sales for each product? To do this we need to enable ViewSheet to check whether a given product code has appeared in column B and, if it has, to enter the money in an analysis column which itself is totalled, so providing the total value of sales for each product.

A set of analysis columns can be set up to the right of the sales columns, like this:



```
 A  SLOT=F1
CONTENTS=*Blank*


0       ........F........G........H........I........J........K........L........M
......1>
......2
......3  CRUNCH!  WHOOSH!    ZOOM!
......4        5        6        7
......5     8.90    10.70     7.80
......6
......7    SALES ANALYSIS
......8   CODES:        1        2        3        4        5        6        7
......9
.....10              0.00     0.00     0.00     0.00    89.00     0.00     0.00
.....11              0.00     0.00     0.00     0.00     0.00   535.00     0.00
.....12              0.00   975.00     0.00     0.00     0.00     0.00     0.00
.....13              0.00     0.00   185.00     0.00     0.00     0.00     0.00
.....14              0.00     0.00     0.00     0.00   890.00     0.00     0.00
.....15              0.00   520.00     0.00     0.00     0.00     0.00     0.00
.....16              0.00     0.00     0.00     0.00  1068.00     0.00     0.00
.....17
.....18 TOTALS:      0.00  1495.00   185.00     0.00  2047.00   535.00     0.00
.....19
```

Enter slots as follows:

Slot	Content
F7	`SALES` (right justified)
G7	`ANALYSIS`
F8	`CODES:` (right justified)

You can now place the numbers 1 to 7 in slots G8 to M8 in the way we used before.

Place the sheet cursor in slot G8

Type: `COL-6` **RETURN**

The value 1 should show in slot G8

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `G8-H8M8` **RETURN**

Now move the sheet cursor to slot G10
Type: `IF(B10=G8,D10,0)` **RETURN**

This last means that if the product code in B10 is the same as the number at the top of this column (ie code 1 in slot G8) then the amount of the order should appear in slot G10. Clearly if we can replicate similar formulae into all the slots between G10 and M16 we shall have a complete analysis of the orders by product code.

Again before replicating you should adjust the slot format of slot G10 to D2RM. Now to replicate:

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `G10-G11G16` **RETURN**
Reply: `R)elative, N)o change?`
`B10=G8,D10,0)`
Press: `R`
Reply: `G8,D10,0)`
Press: `N`
Reply: `D10,0)`
Press: `R`

You should now have the formula replicated down column G.

Press: REPLICATE (f0)
Reply: `From - To?`
Type: `G10G16-H10M10`
Reply: `R)elative, N)o change?`

Each time the slot numbers are presented, type `R` for G8 and `N` for the rest.

All that remains is to set up totals for the analysis columns.

Slot	**Content**
F18	`TOTALS:`
G18	`G10G16`

You can then replicate this relatively across slots H18 to M18 – and the sheet is finished.

Part Two of this book is designed to provide you with the means of looking up ViewSheet's facilities quickly. The main thing is to be ready to experiment, to try out new methods and make new combinations of methods. In this way ViewSheet can become for you what it was designed to be: a tool for thinking.

Part Two
ViewSheet facilities

Contents

7 Recalculation 109

8 Replication 110

9 Value formats 115

10 Screen windows 117

11 Printing out 120

1 Entering ViewSheet

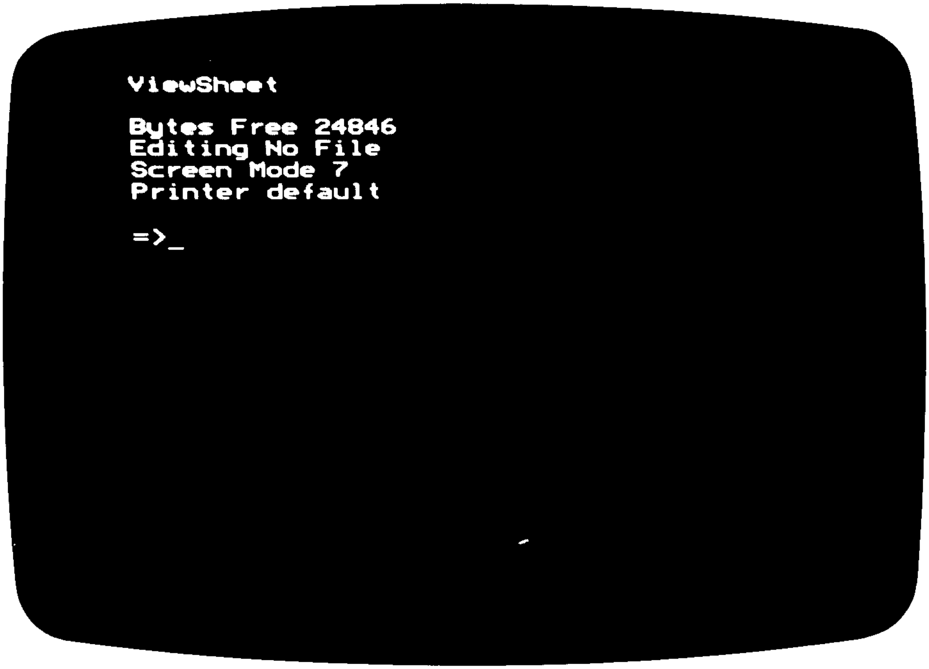
Before using ViewSheet place the function key card under the clear plastic strip at the top of the keyboard. Line it up with the function keys so that DELETE CHARACTER is opposite function key **f9**.

To enter ViewSheet from VIEW, BASIC or another language, type `*SHEET` **RETURN**.

Command mode

Command mode allows you to save and load using disc or tape, to catalogue files, lock files, delete files, compact discs, print out, and change screen modes.

At start-up the display shows:



```
ViewSheet

Bytes Free 24846
Editing No File
Screen Mode 7
Printer default

=>_
```

This means:

- That you are in ViewSheet.
- That you have 24846 bytes available.
- That no file has been loaded from disc or tape.
- That you are in `MODE 7`.
- That the system is using the default printer driver.

2 Commands used in command mode

The following commands can be typed after the prompt (=>). They may be in upper or lower case or any mixture of these.

Abbreviated commands (eg `S` for `SAVE`) must be followed by a space. So `S file` is valid, but `Sfile` is not.

Non-abbreviated commands followed by a number do not need a space. So `MODE 3` and `MODE3` are both valid.

`*ACCESS` `*A.`	Converts disc files to 'read only', ie so that it is impossible to delete or overwrite them. Type: `*ACCESS filename L` **RETURN** To convert files back to read/write, type: `*ACCESS filename` **RETURN**
`*CAT` `*.`	Provides a catalogue of files on the disc or cassette. Type: `*CAT` **RETURN** or `*.` **RETURN**
`CREATE` `C`	Creates a disc file which you can `WRITE` to or `READ` from within the sheet. Type: `CREATE n x y` to create a file V.VSn with dimensions x, y.
`*DELETE` `*DE.`	Deletes a named file from the disc. Type: `*DELETE filename` **RETURN**
ESCAPE	Pressing the **ESCAPE** key moves ViewSheet out of command mode and into sheet mode, or out of sheet mode and into command mode.
`HEADINGS` `H`	Indicates whether column or row headings are set to display the default letters and numbers, or user-defined forms. Reply: `Headings on`; `Headings off`

`HEADINGS OFF`
`H OF`
	Sets the sheet to display default row and column headings even if user-defined headings have been set up.

`HEADINGS ON`
`H ON`
	Reverses `HEADINGS OFF`, setting the sheet to display user-defined row and column headings again.

`LOAD` `L`	Loads a ViewSheet file from disc or cassette. Loading a new file destroys any material currently in the sheet. The `Bytes Free ...` message changes as bytes are used by the file that has been loaded. The message `Editing No File` changes to: `Editing filename` If ViewSheet is in the same mode as it was when the file was saved, all window definitions when saved will be active; if not ViewSheet will use the default windows. Type: `LOAD filename` **RETURN** or: `L filename` **RETURN**
`LW`	Loads a file containing a set of window definitions previously saved with `SW`. The screen mode is automatically reset to the mode that was used when the window definitions were saved. If there is not enough memory, ViewSheet will report the fact. Type: `LW filename` **RETURN**
`MODE` `M`	Changes the screen mode. For details of screen modes, see the *BBC Microcomputer System User Guide*. Different modes use different amounts of memory: `MODE 0` uses the most and `MODE 7` the least. `MODE 3` is probably the most convenient to use with ViewSheet since it displays a relatively large number of slots on the screen. Changing modes turns off all windows except 0 which is reset to the whole of the the screen. Type: `MODE X` **RETURN** where `X` is the mode number.
`NAME` `NA`	Gives a name to the sheet in memory, or renames it if named already. This name is shown in the line `Editing filename` and the sheet can be saved under that name by typing: `SAVE` **RETURN** Type: `NAME filename` **RETURN**

`NEW`	Creates a blank worksheet and sets the default window display for the current screen mode. `NEW` is used when the sheet has been corrupted, for example after using `*COMPACT`. Type: `NEW` **RETURN**
`PC`	Prints out the contents of every occupied slot on the sheet, proceeding from left to right and top to bottom of the sheet. Each slot content is on a new line, preceded by its coordinates. Type: `PC` **RETURN** To stop the printing, press **ESCAPE**
`PRINT` `P`	Prints out the sheet in memory, according to the current printer window definitions. Type: `PRINT` **RETURN**
`PRINTER` `PRINTE`	Loads a printer driver. Type: `PRINTER filename` **RETURN**
`PROTECT` `PRO`	Indicates whether protection of rows and columns is enabled. Type: `PROTECT` **RETURN** Reply: `Protection on`; `Protection off`

`PROTECT OFF`
`PRO OF`
	Disables any protection you may have specified for rows or columns, and the default protection against deleting rows and columns.
	Type: `PROTECT OFF` **RETURN**

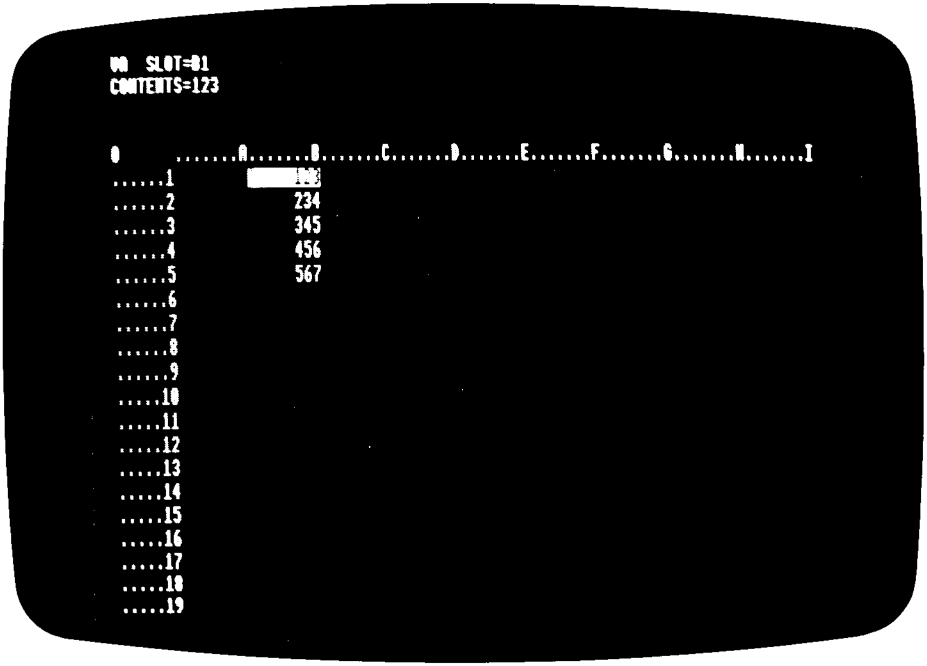
`PROTECT ON`
`PRO ON`
	Reverses the effect of `PROTECT OFF`.
	Type: `PROTECT ON` **RETURN**

`SAVE` `S`	Saves the whole sheet onto disc or cassette, including slot contents, row and column headings, protection of rows and columns, formats and values, window definitions, and the current window. Type: `SAVE filename` **RETURN** or `S filename` **RETURN** `SAVE` overwrites any file of the same name on the disc.

	If you have previously loaded a file into ViewSheet, or named the sheet using the `NAME` command, so that the name of this file is displayed in command mode, you can save an updated version of the file by typing `SAVE` **RETURN** without the filename.
`SCREEN` `SC`	Displays the sheet on the screen according to the current print windows. Type: `SCREEN` **RETURN**
`SW`	Saves the current screen and printer window definitions to a named file. Type: `SW filename` **RETURN**
`*TV`	To prevent the screen from flickering, type: `*TV0,1` **RETURN** and `MODE 3` **RETURN** To lower the image on the screen, type: `*TV255` **RETURN** and `MODE 3` **RETURN** To do both, type: `*TV255,1` **RETURN** and `MODE 3` **RETURN**

3 Sheet mode

The sheet is 255 slots wide by 255 slots deep. The default column headings are A to IU, and the row headings 1 to 255. You may vary the width of row headings, and of columns and their headings (see chapter 10, 'Screen windows'). Sheet mode layout is as follows:



```
VA  SLOT=A1
CONTENTS=123


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1           123
......2           234
......3           345
......4           456
......5           567
......6
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

V – This is known as a 'flag'. It indicates that the cursor is in a slot containing a value. If the slot contains a label, the flag is `L`.

A – This is another flag indicating 'automatic recalculation', ie that the sheet will recalculate every time a new value is entered. For manual recalculation the flag is `M`.

`SLOT=` shows the coordinates of the current slot, ie the slot where the cursor is.

`CONTENTS=` shows the contents of the slot. If the contents are a label or a number, the label or number is shown. If the contents are a formula, the formula is shown, the number which results from applying that formula being shown in the slot itself.

The `CONTENTS=` line shows the contents in full, whereas the slot shows only as many characters as will fit into the displayed slot. Thus if your slot width is at the default value of seven characters, and you enter a label of 25 characters, only the first seven characters appear in the slot, but all 25 will appear in the `CONTENTS=` line. See chapter 5, 'Values and labels' for more discussion of contents.

The sheet cursor

The sheet cursor varies according to the screen mode. In `MODE 7` it is a prompt; in other modes it is a white rectangle.

It can be moved across the sheet in any direction by using the four arrow keys or by using the GO TO SLOT command.

Using the arrow keys

Holding down **SHIFT** as you press an arrow key moves the cursor across the sheet by the whole width of the window horizontally, or by a whole window depth vertically.

Using the GO TO SLOT command

Press: Function key **f7**
Reply: `Slot?`
Type: Slot reference in the form `K95` and press **RETURN**

The cursor will move to the slot named.

When the cursor is moved past the edge of the screen, the window moves across the sheet and the row and column headings change to show the current cursor position.

4 Columns and rows

User-defined headings

You can make up your own headings to replace the default row and column headings using the commands COLUMN HEADING and ROW HEADING.

SHIFT **f4** and **SHIFT** **f3** are used for this as follows.

Row headings

Place the cursor in the row whose heading you wish to change
Press: **SHIFT** **f4**
Reply: `Row heading?`
Type your new heading and press **RETURN**

Column headings

Place the cursor in the column whose heading you wish to change
Press: **SHIFT** **f3**
Reply: `Column heading?`
Type your new heading and press **RETURN**

Headings must be of not less than three characters and not more than 15, and they must not start with a digit. How many characters are displayed in the sheet border depends on the border width you have set.

If you have defined a column or row heading of your own and wish to edit it, press COLUMN HEADING or ROW HEADING and ViewSheet displays it on the editing line, where you can use all the editing facilities to change it.

Use of user-defined headings in formulae

Having defined a heading of your own, you may use these new names in formulae. So for example if you replace column A with `SALES` and row 10 with `CAMBRIDGESHIRE` you may use a formula such as `("SALES"CAMBRIDGESHIRE"*.15)+B2`

Note the use of double quotes in the slot references.

`"SALES"16`	for slot A16
`B"CAMBRIDGESHIRE"`	for slot B10
`"SALES"CAMBRIDGESHIRE"`	for slot A10

The double quotes are in fact merely separators, which is why only one set of quotes appears between `SALES` and `CAMBRIDGESHIRE`. However, ViewSheet will not reject two sets of quotes in this position.

Even though you have defined column and row headings of your own ViewSheet still allows you to use the default headings. To save you the trouble of remembering what these are you can use the **SHIFT** **COPY** facility. Just move the cursor to the slot you wish to reference, hold down **SHIFT** and press **COPY** – and the slot reference will appear in the editing line.

Turning off user-defined headings

You may turn off user-defined headings temporarily by changing to command mode and typing:

`HEADINGS OFF` **RETURN**

To turn them on again type:

`HEADINGS ON` **RETURN**

Protecting columns and rows

Unless a slot is protected, its contents can be deleted or overwritten at any time. Slots cannot be protected individually; only columns and rows can be protected. A slot is protected if it is in a protected column or row.

Protection prevents the following from operating in the column or row concerned:

- DELETE SLOT.
- Slot editing.
- Typing over the slot.
- Replicating over the slot.
- Deleting columns or rows.

If you attempt to carry out any of these operations in a protected column or row, the computer will bleep and the message: `Protected` will appear at top left.

Protection of a column or row is indicated by changed characters in the heading. In `MODE 3` the dots are replaced by underline characters, and in `MODE 7` by dashes.

```
       ..........A..........B..........C__________D..........E
......1                                 This is a
______2 This is a protected row         protected
......3                                 column
......4
......5
......6
......7
......8
......9
.....10
```

To protect a column or row place the cursor in the row or column concerned.

Press: **SHIFT** **f5** (PROTECT COLUMN)
or **SHIFT** **f6** (PROTECT ROW)

The dots in the column or row heading are then replaced by a continuous line.

To remove protection press the same key again.

To disable protection temporarily:
Switch to command mode by pressing **ESCAPE**
Type: `PROTECT OFF` **RETURN**

To enable protection when disabled:
Switch to command mode by pressing **ESCAPE**
Type: `PROTECT ON` **RETURN**

To check whether protection is on or off:
Switch to command mode by pressing **ESCAPE**
Type: `PROTECT` **RETURN**

Inserting and deleting columns and rows

To insert a column or row move the cursor to the place where you want the new column or row to appear and press **SHIFT** **f1** (INSERT COLUMN) or **SHIFT** **f2** (INSERT ROW).

After an interval the new column or row will appear. When it does, all the slot references and all the formulae involved will have been adjusted so that the sheet works as before. Naturally the processing involved is complex and in a large sheet may take an appreciable time. While the adjustment is going on the processing signal (moving dots) at top left will be seen.

Deleting columns and rows is a special case, in that ViewSheet provides additional protection in such cases. You may only delete a column or row if you specifically disable protection first, even though no protection is shown in the column or row heading.

To delete a column or row first disable protection by switching to command mode and typing: `PROTECT OFF` **RETURN**

Then place the cursor in the row or column concerned and press **CTRL** **f_1** (DELETE COLUMN) or **CTRL** **f_2** (DELETE ROW). The column or row will be deleted and the other columns or rows will close up. Again all slot references in related formulae will be adjusted automatically.

Remember to enable the protection again when you have finished.

Limits of insertion

If any values or labels are entered in row 255, it is impossible to insert a row anywhere since this would push row 255 off the edge of the sheet. In such cases you must first delete a row somewhere else before inserting.

Note that it is not sufficient to delete slot contents in row 255. A whole row must be deleted using the DELETE ROW command. The same applies to columns when entries have been made in column IU.

5 Values and labels

Slot entries are either values or labels.

In effect ViewSheet examines the characters you are entering and decides whether they make up a value or a label. If they can be interpreted as a value, ViewSheet enters them as such. Otherwise they are entered as a label. If you wish to use numbers (eg 1984) as labels, you should place them between single quotes or any other characters which prevent their being interpreted as values.

When the cursor is in a slot containing a value, the V flag is displayed at top left. If the slot contains a label, the L flag is displayed.

Entering

The simplest items which can be entered are numbers and labels.

To enter a number or label:

- Place the sheet cursor in the slot required.
- Type the number or label, which will appear on the editing line at top left.
- Press **RETURN**, and the number or label appears in the slot, and V or L at the top left of the screen.

Once **RETURN** is pressed the entry is complete.

If L is displayed when you thought you were entering a value, look carefully at what you have entered – a mistake can cause a number to be incomprehensible to ViewSheet, which will then enter it as a label.

Pressing **ESCAPE** at any stage aborts the entry.

For entering formulae and commands involving ranges of slots, see chapter 6.

Right and left justification

In default condition, values are right justified in slots, and labels left. Justification of values may be changed by editing slot format (see chapter 9) or changing the format in the whole window by adjusting the window definition (see chapter 10).

Justification of labels can be changed by placing the cursor in the slot concerned and pressing SHIFT f8 (JUSTIFY LABEL). This reverses the current justification, ie left to right, or right to left.

Deleting slots

To delete the contents of a slot:

Place the cursor in the slot concerned.
Press: DELETE SLOT (SHIFT f9)

Editing entries

In the simplest kind of editing, you type in the characters concerned and press RETURN. At a more complex level you may revise what you have typed before pressing RETURN; or move to a slot, copy the contents of it into the editing line, modify the contents, and copy them back into the slot again.

Such operations involve the use of ViewSheet's editing facilities.

When typing in or editing an entry you are in effect using two cursors at the same time:

- The sheet cursor which is in the slot.
- The line cursor on the editing line.

Each of these cursors can be controlled independently, using the arrow keys.

To move the line cursor, hold down CTRL and press the forward or backward arrow key. To move the sheet cursor press the arrow keys alone.

You may also move the sheet cursor to another slot during the editing of an entry. Wherever you move it, the entry goes into the slot which the cursor was in when you started editing the entry. This facility for moving the cursor during editing is particularly useful when entering formulae which involve slot references: you can check on the slots your formula will refer to as you make the formula up.

SHIFT COPY

This is another feature of ViewSheet which can be used along with the ability to move the cursor during slot entry.

Move the cursor to a selected slot, hold down **SHIFT** and press **COPY** – and the slot reference is copied into the editing line.

Once you have started your slot entry, therefore, you can move to the slots whose references you need, and copy them into the editing line as part of your formula. If your formula starts with a reference to another slot, you can start the editing process by typing a space: this will be ignored when the formula is entered.

Other editing facilities

BEGINNING OF LINE (**f4**) moves the line cursor to the beginning of the line.

END OF LINE (**f5**) moves the line cursor to the end of the line.

DELETE (the black key at bottom right) deletes the character to the left of the line cursor.

DELETE END OF LINE (**f3**) deletes the character which the cursor is on and all characters to the right of it on that line.

INSERT CHARACTER (**f8**) opens up a space in the text at the line cursor, leaving the cursor there so that you may type in another character at that point.

DELETE CHARACTER (**f9**) deletes the character at the cursor, closing up the text from the right.

You should practise using the line editor commands, since they save a great deal of time when typing in complex formulae or long labels.

Note that the editing line is capable of holding up to 240 characters. Since all 240 cannot be displayed, the line characters can be scrolled right and left by using the arrow keys with **CTRL**.

Re-editing slot contents

You may copy the contents of a slot into the line editor for modification using the commands listed above, and put the corrected version back in the slot.

To do this, place the sheet cursor in the slot concerned and press **COPY** (at bottom right on the keyboard).

The contents of the slot appear in the editing line at top left, and all the commands listed above can be used.

When editing is complete, press **RETURN**, and the line is entered in the slot.

AUTO ENTRY

This saves time when you wish to enter a series of numbers in the same row or column. When AUTO ENTRY is set, the sheet cursor moves on to the next slot automatically as soon as you press **RETURN**, without your having to touch the arrow keys.

Press: **CTRL** ***f0***

The letter `R` appears on the flag line. Every time **RETURN** is pressed to enter a value or a label, the sheet cursor moves automatically to the right.

To change to movement down a column, press **CTRL** ***f0*** again. The letter `D` appears on the flag line. Pressing **RETURN** to enter a label or value moves the sheet cursor down to the next slot in that column.

To cancel AUTO ENTRY, press **CTRL** ***f0*** again, and neither `R` nor `D` is shown.

Errors

Errors resulting from expressions are signalled by the word `?Error` in the slot concerned. Division by zero, or an absurdly high number will result in such an error.

Percent sign in the slot signals that the number is too wide for the current slot format. For example if the window format is five characters with two decimal places it is impossible to display `1234.56`. However ViewSheet will display the number in scientific notation if possible. If the window format is changed to six characters with two decimal places ViewSheet can display `1234.56` as `1.23E3` (scientific notation meaning 1.23 × 1000). If the window format is changed to seven characters ViewSheet can display `1234.56`.

Errors resulting from editing are signalled by a bleep, the text cursor indicates where the error is, and an error message appears on the prompt line above the editing line. No entry is made in the slot.

At this point you have the choice of pressing **ESCAPE** to abort the entry, or using the editing facilities to amend the entry.

A list of error messages and their meanings is given in chapter 13.

6 The contents of slots

Formulae

Formulae may be composed of slot references, numbers, functions, operators and constants. For example a formula may consist simply of an instruction in slot C1 to multiply the contents of slot A1 by two, or an arithmetic expression such as `3.7*1234/0.06`. It may also be very long and complex, involving many slot references.

If a formula is entered in a slot, the number which results from that formula is displayed in the slot. If you wish to check the formula that produced the number, place the cursor in the slot and the number will be shown in the `CONTENTS=` line.

For example, if the number `123` is entered in slot A1 and the formula `A1*2` is entered in slot C1, then the number `246` will appear in slot C1 and when the cursor is in that slot the formula `A1*2` will appear in the `CONTENTS=` line.

Slot references

Columns must always be given before rows; for example: `A5`

When using the default letters for columns, you may use lower-case characters (eg `a5`) which ViewSheet converts to upper-case automatically.

If column or row headings have been user-defined, you may, of course, change the case of characters in these. You may also use the default headings rather than the user-defined headings if you wish.

If you include user-defined headings in formulae, you must use separators:

```
"SALES"23
C"JULY"
"SALES"JULY"
```

For more information on user-defined headings see chapter 4, 'Columns and rows'.

Constants

Constants are numbers in decimal form; or pi, which has the value 3.141592653.

Ranges

A range is a row or a column of slots bounded by two slot references.

In formulae a range is shown by placing the two references together:

```
A1G1
"SALES"JANUARY""SALES"DECEMBER"
AA120AA131
```

If this last were entered in a slot, the slot would show the sum of all the values in the range of twelve slots bounded by AA120 and AA131.

When slots without values are left in ranges, ViewSheet assumes they are set to zero. Note that labels occurring within a range are counted as zero.

Operators

The following operators may be used in formulae:

+	addition
-	subtraction or unary minus
*	multiplication
/	division
^	raise to power
( )	brackets

Conditional operators

Conditional operators used with ViewSheet are:

=	equal to
<>	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

These work as normal binary operators and result in the value 0 for false, and 1 for true.

This can be demonstrated by entering the following:

Slot	**Contents**
A1	45
A3	11
A10	A1>A3
A12	A1<A3

Clearly the statement in A10 is true and the statement in A12 is false. The sheet will therefore show 1 (for true) in A10, and 0 (for false) in A12. All conditional operators work in this way. They are of particular value when used with the `IF` function (see below).

Precedence of operators

The order of precedence for all the operators mentioned above is:

Group 1	`-`	unary minus
	`()`	brackets
Group 2	`^`	raise to power
Group 3	`*`	multiply
	`/`	divide
Group 4	`+`	plus
	`-`	minus
Group 5	`=`	equals
	`<>`	not equal to
	`<`	less than
	`>`	greater than
	`<=`	not greater than
	`>=`	not less than

For example `A1-B2*C3^D4` is equal to `A1-(B2*(C3^D4))`

Functions

The number 'n' in parentheses following some of the functions below is the 'argument', ie the number which the function uses. For example, `INT` is a function which strips any digits after the decimal point from a number, leaving only the integers. Suppose that the number `123.456` is in slot A1. If we enter the formula: `INT(A1)` in slot C1 then the number `123` alone will appear in C1.

Note that *ranges* of slots may be specified as arguments to some functions, a range being part of a row or part of a column of slots. In such cases the range is used as a list of arguments, rather than as the sum of the values in it.

`ABS(n)`	Absolute value of n
`ACS(n)`	Arccosine of n (the value returned is in radians)
`ASN(n)`	Arcsine of n (the value returned is in radians)
`ATN(n)`	Arctangent of n (the value returned is in radians)

`AVERAGE(list)`	Works out the average of values in the list. The list may consist of numbers, slot references, a range, and other functions, separated by commas. For example:
	`AVERAGE(23,A4,B15-0.2,A4-B4)`
	`AVERAGE` becomes considerably more powerful if used with a range as its argument.
	For example, if you want to average 50 numbers entered in slots A1 to A50 it is convenient to enter the formula:
	`AVERAGE(A1A50)`
	When using `AVERAGE` in this way, you should be quite clear as to how it works. In the last example ViewSheet adds up the values in all 50 slots and divides by 50. It will therefore produce an average which includes zeros for all slots into which you have not yet entered amounts – *not* an average of the amounts actually entered. So when using `AVERAGE` make sure your range covers exactly the slots you wish to include in the calculation.
	Note also that a range may be included as an item in the list:
	`AVERAGE(A1,B1B10,C3)`

	In this case the average is of all the slots named in the list, and all the slots included in the range.
`CHOOSE(n,list)`	Selects the nth value in the list. This is most useful when n is a slot reference, so that the selection is governed by another calculation. Such an example would be: `CHOOSE(D1,12,13,INT(B1),25)`
`COL`	Gives the number of the column which contains the slot
`COS(n)`	Cosine of n (n is in radians)
`DEG(n)`	Converts n from radians to degrees
`EXP(n)`	Exponent of n

`IF`	`IF` is used in the form:
	`IF(value1,value2,value3)`
	If value1 is zero, the value of the `IF` function becomes that of value3. If value1 is other than zero, the value of the `IF` function becomes that of value2.
	For example:
	`IF(10>9,100,200)`
	`IF(10>123,100,200)`
	In the first case, the value of the `IF` function is 100; in the second it is 200. The expression 10>9 is true, so it evaluates to 1; whereas 10>123 is false, so it evaluates to 0. In practice many `IF` statements consist largely of slot references:
	`IF(MAX(A1A100)>=B12,C1G1,D2)*1.2`
	Note in the last example that the value of the `IF` function has been multiplied by 1.2.
	Note also that `IF` statements can be nested:

```
IF(A1A3=B2,IF(C1>=D1,C1,D1),0)
```

This means that if the total of the range A1 to A3 is equal to the value in slot B2, then the value to be shown is the greater of C1 or D1; otherwise the slot should show zero.

For a discussion and examples of the use of the `IF` function see Part One chapter 9.

`INT(n)`

Truncate to integer. Note that this does not behave exactly like the BASIC function `INT`. In ViewSheet, `INT (-2.5)` returns `-2`. In BASIC, `INT (-2.5)` returns `-3`.

`LN(n)`

Natural logarithm of n

`LOG(n)`

Logarithm of n to base 10

`LOOKUP`

The function `LOOKUP` compares a specific value with the values in a range (the lookup range). If it finds an equal value it locates the corresponding value in another range (the result range). This number then becomes the value of the `LOOKUP` function. See Part One chapter 9.

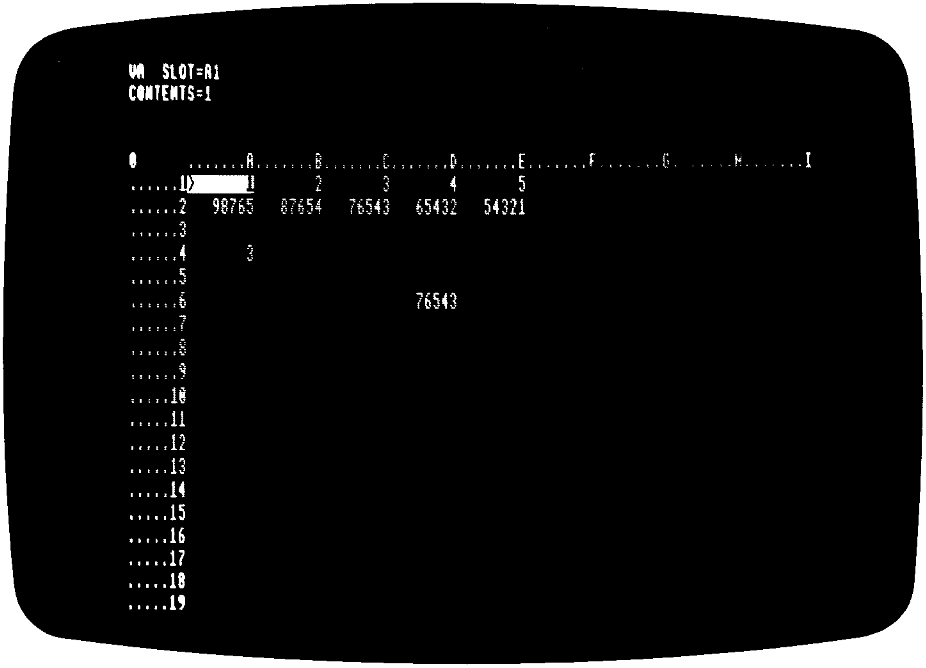
The form used is:

```
LOOKUP(value,lookup range,result
range)
```

In the example below the lookup range is A1 to E1 and the result range A2 to E2; the value used in the `LOOKUP` function is in A4. The `LOOKUP` formula entered in D6 is:

```
LOOKUP(A4,A1E1,A2E2)
```

Since the value in A3 is 3, the value of the `LOOKUP` function is 76543, being the value in the result range corresponding to 3 in the lookup range.



```
VA  SLOT=A1
CONTENTS=1


0      .......A.......B.......C.......D.......E.......F.......G.......H.......I
......1>      1         2         3         4         5
......2   98765     87654     76543     65432     54321
......3
......4       3
......5
......6                                 76543
......7
......8
......9
.....10
.....11
.....12
.....13
.....14
.....15
.....16
.....17
.....18
.....19
```

`MAX(list)`	Selects the maximum value in the list. The list may consist of numbers, slot references, a range or other functions, separated by commas. For example:
	`MAX(W3,D4D25,SIN(A1/10))`
	Note that a range may be included as an item in the list:
	`MAX(2,3,A1A10,B5)`
	In this case all the values specified in the list, and all the slots included in the range are searched for the maximum value.

`MIN(list)`	Selects the minimum value in the list. Used in the same way as **MAX**.
`PI`	Gives the value of pi (3.141592653)

`RAD(n)`	Converts n from degrees to radians
`READ(n,x,y)`	Reads from position x,y of disc file V.VSn
`ROW`	Gives the number of the row which contains the slot
`SGN(n)`	Sign of n. If `n` is negative, `-1` is returned. If `n` is `0`, `0` is returned. If `n` is positive, `1` is returned.
`SIN(n)`	Sine of n (n is in radians).
`SQR(n)`	Square root
`TAN(n)`	Tangent of n (n is in radians)
`WRITE(n,x,y,z)`	Writes z to position x,y in disc file V.VSn

Further information on functions and operators, and on error messages which result from incorrect use of them can be found in the *BBC Microcomputer System User Guide.*

7 Recalculation

On or off

Unless you instruct it otherwise, ViewSheet recalculates the whole sheet after each entry, ie after you have pressed RETURN. This may become inconvenient when you have a large number of slots filled and have continually to wait for the recalculation to be completed. It is particularly time-wasting when you are reading from and writing to disc files within the sheet; unless you turn off recalculation these are accessed frequently as you work on the sheet.

To turn recalculation off and on again, press SHIFT f0 (RECALCULATE MODE).

When recalculation is turned off, the flag `A` at top left changes to `M`. On entering a formula in this condition, the formula is evaluated and appears in the slot but the sheet is not recalculated.

You can cause ViewSheet to recalculate the sheet at any time by pressing SHIFT f7 or TAB .

Direction of recalculation

ViewSheet recalculates from left to right on each row, starting at row 1 and progressing down the sheet.

This fact is worth remembering when you make references between slots. A formula which refers to a slot lower down the sheet or to the right is perfectly valid, but may produce unwanted effects since it may refer to a non-recalculated value.

On entering a formula into a slot ViewSheet evaluates that formula and then recalculates the sheet starting at A1. So if another formula references the slot the new value of the slot is used. Note that the slot is calculated twice.

8 Replication

The REPLICATE command allows you to copy the contents of any of the following to any other part of the sheet:

- Single slots.
- Rows of slots.
- Columns of slots.

The REPLICATE command also allows you to copy all these in the following two ways:

- *Unchanged,* ie reproducing exactly the same slot contents elsewhere.
- *Relatively,* ie changing the slot references in formulae so that they relate to the new environment.

General procedure

Press: REPLICATE (f0)
Reply: `From - To?`

Replication can take place:

- *From one slot to another.*
- *From a slot to a range;* for example when replicating the number or formula in one slot into all the slots of a range.
- *From a range to a slot.* This is used to replicate the contents of a column range into another column range, or of a row range into another row range. The range to be replicated is given in full, but only the first slot of the range to be replicated into is given.
- *From a range to a range.* This is used for replicating a column or row range into a whole block of slots. If a column is to be replicated, the column range is given before the dash and the row after.

A dash (-) is placed between the slot or range to be replicated, and the slot or range into which the replication is to take place.

For example:

`A1-B1`	Replicate A1 contents into B1
`A1-B1H1`	Replicate A1 contents into the range B1 to H1
`A1A15-G1`	Replicate the contents of the range A1 to A15 into the range G1 to G15
`A1A15-B1H1`	Replicate the contents of the range A1 to A15 into all the columns along the row range B1 to H1

Type the slots or ranges as above and press **RETURN**. If there is a slot reference in the range to be replicated, ViewSheet will reply

```
R)elative, N)o change?
```

and the first slot references in the formula will appear in reverse video on the editing line, followed by the rest of the expression in order to help you locate the reference. If you want to transfer a reference unchanged, press `N`; if you want a reference to be adapted to the range into which the slot contents are to be replicated, press `R`.

All slot references in formulae are treated in this way. When you have decided on each one, the replication takes place.

In a complex sheet, replication can take a noticeable time. While it is going on, the processing signal appears at top left (a moving dot), and the progress of the replication is shown by a slot reference on the `SLOT=` line, which continually changes as replication proceeds.

Replication: single slot to single slot

Press: REPLICATE (**f0**)
Reply: `From - To?`
Type (for example): `K1-T3` **RETURN**
If the first slot contains a number only, the slot contents will be replicated immediately. If it contains slot references, ViewSheet replies

```
R)elative, N)o change?
```

and the slot references in the formula appear one by one in reverse video on the editing line.
Press: `R` or `N`
If there are several slot references in a formula you will have to decide on each with `R` or `N` before replication is carried out.

Replication: single slot to range

Press: REPLICATE (f0)
Reply: `From - To?`
Type (for example): `E1-E2E30` RETURN

If the slot contains a number only, the slot contents will be replicated immediately. If it contains slot references, ViewSheet replies

```
R)elative, N)o change?
```

and the slot references in the formula appear one by one in reverse video on the editing line.

Press: `R` or `N`

If there are several slot references in a formula you will have to decide on each with `R` or `N` before replication is carried out.

Example: A1 is to be added to C1 and displayed in E1 and the same procedure should be followed on all rows up to row 30. Numbers are entered in columns A and C.

Move the cursor to E1
Type: `A1+C1` RETURN
Press: REPLICATE (f0)
Reply: `From - To?`
Type: `E1-E2E30` RETURN
Reply: `R)elative, N)o change?`
`A1+C1`
Press: `R`
Reply: `R)elative, N)o change?`
`C1`
Press: `R`

The contents of slot E1 are then replicated relatively into E2 to E30 so that slot E2 contains the formula: A2+C2 and so on for succeeding rows. Enter numbers in columns A and C to test the replication.

Replication: range to single slot

The purpose of this is to replicate the contents of a column range into another column range, or of a row range into another row range.

Press: REPLICATE (f0)
Reply: `From - To?`
Type (for example): `A1 A20-H1` **RETURN**
If the range contains numbers only, the contents of the column A1 to A20 will be replicated into column H1 to H20.
If the range contains slot references ViewSheet replies

```
R)elative, N)o change?
```

and the slot references in the formulae appear one by one in reverse video on the editing line.
Press: `R` or `N`
If there are several slot references in a formula you will have to decide on each with `R` or `N` before replication is carried out.

The procedure for copying rows is the same. For example, to replicate row `A1` to `H1` into row 15 the command would be: `A1 H1-A15`

Replication: range to range

The purpose of this is to replicate the contents of a column into a range of columns, or of a row into a range of rows. Essentially this is the same procedure as in the last example, except that instead of replicating into a single range we are replicating into a series of ranges, but we are still naming the slots at the start of the ranges in order to do so.

When using this procedure it is worth remembering that if the range to be replicated is a column, then the range to be replicated into is a row, and vice versa.
Press: REPLICATE (f0)
Reply: `From - To?`
Type (for example): `A1 A20-B1 J1` **RETURN**
If the range contains numbers only, the contents of column A1 to A20 will be replicated into columns B to J.
If it contains formulae ViewSheet replies

```
R)elative, N)o change?
```

and the slot references in the formulae appear one by one in reverse video on the editing line.

Press: `R` or `N`

If there are several slot references in a formula you will have to decide on each with `R` or `N` before replication is carried out.

Since the replication involves more than one column you will also have to decide on each column with `R` or `N` before the replication is carried out.

The procedure for replicating the contents of a row into a range of rows is the same. For example to replicate the contents of the row A1 to J1 into rows 2 to 10 the command would be: `A1 J1 - A2 A10`

If you replicate off the edge of the sheet ViewSheet bleeps and replies `Edge`.

Protection

If rows or columns have been protected, ViewSheet will not change the slots. If you wish to change them you should switch to command mode by pressing **ESCAPE** and type:

```
PROTECT OFF RETURN
```

To restore protection type (in command mode):

```
PROTECT ON RETURN
```

Deleting blocks of slots

The REPLICATE command may be used to delete parts of columns or rows. Simply copy a block of empty slots into the area to be cleared.

To delete whole rows or columns use the DELETE ROW or DELETE COLUMN command.

9 Value formats

ViewSheet allows you to adjust the way in which values are displayed in slots. You may set:

- The number of decimal places to be displayed.
- Rounding to the next whole number.
- Right or left justification.
- Minus sign or brackets for negative numbers.

You may set a particular format for an individual slot, or for a whole screen or printer window.

Setting formats for individual slots

Press: EDIT SLOT FORMAT (f6)
Reply: `Format?`
`FRM`

`FRM` is the format set at switch-on. The actual format shown after pressing function key f6 will be the format of the window the cursor is in.

To change the format, type in the alternative code letters as shown below and press RETURN. Note that it is only possible to reformat a slot which has a value in it.

F

`F` stands for 'floating point' and indicates that there is no fixed number of decimal places to be displayed. Unless instructed to the contrary ViewSheet ranges all numbers right in the slot, which is entirely suitable for integers, but with decimal numbers may produce effects such as:

```
1.23456
12345.6
123.456
```

To set a fixed number of decimal places to be displayed, replace `F` with `D` followed by the number of decimal places required. The maximum is nine, although of course you would need to adjust the column width (see chapter 10, 'Screen windows') to make use of that number.

For example, `D2RM` specifies values displayed with two decimal places, right justified and with minus signs for negative numbers.

Rounding to the next whole number can be achieved through setting the format to `D0`.

R

`R` stands for 'right justification', ie numbers are lined up according to their extreme right-hand digit.
To justify left, replace the `R` with `L`.

For example, `D2LM` specifies values displayed with two decimal places, left justified and with minus signs for negative numbers.

M

`M` stands for 'minus', ie negative numbers are displayed with a minus sign in front of them.
To show negative numbers with brackets instead of a minus sign, replace the `M` with `B`.

For example, `D2LB` specifies values displayed with two decimal places, left justified and with brackets for negative numbers.

Range of numbers in ViewSheet

ViewSheet will accept values up to approximately 2×10^{38} and down to approximately 2×10^{-39}. If values above these limits occur, the word `?Error` will appear in the slot. A value below this limit will be displayed as `0` in the slot.

If a number is too large or too small to display in the column width set, it is shown in exponential format. If a `D` format has been set, exponential numbers will have the correct number of decimal places.

If the number is too long to display even in exponential format, this is signalled by a single percent sign in the slot, but the number can be seen in the `CONTENTS=` line above.

10 Screen windows

ViewSheet allows you to select up to ten rectangular areas of the sheet and to place them on the screen as windows. The effect is as if you were looking through each window onto a specified area of the sheet. In this way the most significant parts of the sheet may be assembled together to make a single display on the screen.

Windows may be scrolled independently or together. Slot formats and borders for each window may be individually specified. A window may be of any size from the total area of the screen (less the top four lines and the bottom line) to a single slot.

When ViewSheet is entered or when the screen mode is changed, window 0 covers the total area of the screen. How much of the sheet this covers depends on which screen mode has been selected. In `MODE 3` window 0 covers columns A to I and rows 1 to 19. In `MODE 7` it covers columns A to D and rows 1 to 19.

Window definitions

Windows are numbered 0 to 9 and are controlled by setting window definitions.

As mentioned above, when switched on ViewSheet's default condition is for window 0 to cover the whole screen. Before you can get other windows onto the screen, therefore, you must first make room for them by making window 0 smaller.

The general method of resetting window definitions is as follows.

Press: EDIT WINDOW (f_1)
Reply: `Window?`
Press (for example): `0`
Reply:

```
Wi    TopL  BotR  Pos   Cw    Bw    Fmt   Opt
0     A1    I19         7     7     FRM
```

`Wi`	The window number.
`TopL`	The top left slot of the window.

`BotR`	The bottom right slot of the window. If the window is a single slot, the top left and bottom right slot references are the same.
`Pos`	The position of the window on the screen. This is given in relation to another window in the form `B0` (meaning below window 0) or `R1` (meaning to the right of window 1). The window referred to must always be numbered lower in the window sequence than the window being defined. It is impossible to define the position of a window in relation to a window with a greater number. Therefore no entry is needed for window 0 which is always in the top left of the screen. A blank line or a blank column is left between windows.
`Cw`	Column width: the number of characters which can be placed in the column. When calculating the number of columns which will appear on the screen, remember that an additional character space is used to separate columns. The minimum column width is three characters and the upper limit is whatever will fit into the screen window in the screen mode you are using.
`Bw`	Border width, ie the width of the row headings. The maximum is 15 characters, and the minimum is two.
`Fmt`	Value format, ie the number of decimal places displayed in the slot; rounding; right or left justification; minus sign or brackets with negative numbers. The rules are the same as when setting formats for individual slots. See chapter 9, 'Value formats'.
`Opt`	Options. One or more of the codes below may be placed under `Opt`.
`H`	Horizontal scroll. When a window with option `H` set is scrolled, all other windows with `H` set scroll with it.
`V`	Vertical scroll. When a window with option `V` set is scrolled, all other windows with `V` set scroll with it.
`S`	Side border off. Row headings are not displayed.
`T`	Top border off. Column headings are not displayed.
`C`	Substitute asterisks for the value, one asterisk for each unit of the value. This is the bar chart facility. Negative values are shown as exclamation marks.

0 Window off. It may be convenient to turn windows off temporarily to make adjustments to the sheet, without losing all the other settings.

Note that options `H` and `V` can only be used with screen windows.

Editing window definitions

Window definitions can be edited just like any other line. Hold down CTRL and use the arrow keys to move the cursor along. Other editing commands such as INSERT CHARACTER can also be used.

An error in setting a window parameter results in a bleep as soon as RETURN is pressed, and the cursor is placed on the offending parameter.

If you wish to set print windows the same as your screen windows, you have only to use EDIT WINDOW and change the window number by inserting a `P` in front of it.

Displaying windows on the screen

Once windows have been defined, ViewSheet will display all that it can in order from 0 to 9. If it cannot fit a window in as defined, it will first try to reduce its size and so fit in as much of it as possible. If there is not room even for the reduced window, ViewSheet will turn it off.

Saving and loading windows

The `SW` and `LW` commands are described in chapter 2. To save a set of windows type `SW` followed by your chosen filename and RETURN. To load the windows again type `LW` followed by the filename and RETURN.

NEXT WINDOW command

The cursor may be moved between windows as follows:
Press: NEXT WINDOW (f_2)

Screen window scrolling

Although screen windows are set to cover a specific area of the sheet, they are not immovably fixed in that area. By placing the cursor in the window concerned and using the arrow keys the window may be moved in any direction and for any distance over the sheet.

11 Printing out

General method

Printing in ViewSheet is controlled by printer windows. The system allows you to print out a standard area of maximum size 255 characters wide by 255 lines deep. In practice of course the capability of your printer will probably modify the width measurement and you will have to bear this in mind when setting up ViewSheet to print.

ViewSheet allows you to place up to ten printer windows anywhere on the printing area, just as you place screen windows on the screen. Definitions for printer windows P0 to P9 are set in exactly the same way as those for screen windows.

(For a more detailed discussion of printer windows see Part One chapter 5.)

Setting up the printer

How this is done depends on what kind of printer you have and whether you need anything more sophisticated than simply printing out the sheet.

Connecting leads for the printer should be supplied by your dealer, although the *BBC Microcomputer System User Guide* gives some advice.

Printer drivers are programs which control printing. They enable you to print all labels and values in a given window in bold or underlined type, for example, provided that your printer is capable of this. ViewSheet itself contains a printer driver which operates by default if no other is loaded. For other drivers see Acornsoft sales leaflets.

Printing and baud rate codes are as in the *BBC Microcomputer System User Guide*. They are typed in command mode.

`*FX5,1`	Selects output to a parallel (Centronics® type) printer.
`*FX5,2`	Selects output to a serial (RS423) printer.
`*FX5,0`	Selects output to the screen.

`*FX8`	Selects the baud rate, as follows:

`*FX8,1`	75 baud
`*FX8,2`	150 baud
`*FX8,3`	300 baud
`*FX8,4`	1200 baud
`*FX8,5`	2400 baud
`*FX8,6`	4800 baud
`*FX8,7`	9600 baud
`*FX8,8`	19200 baud

Printing out the default window

The rule is that ViewSheet prints out all printer windows that are active.

Window P0 is active by default, ie unless you turn it off, and so can conveniently be used to print whatever is in the top left corner of the sheet. Unless adjusted, window P0 corresponds to the default screen window 0, which in turn depends on the mode you start in. For example if you start in `MODE 3`, screen window 0 and therefore printer window P0 will be from A1 to I19.

It is therefore possible to print whatever happens to be in the top left corner of the sheet with a minimum of trouble: just set up the printer as described above; switch to command mode; then type `PRINT` and press **RETURN**.

Setting printer windows

Press: EDIT WINDOW (**f1**)
Reply: `Window?`
Press (for example): `P0`
Reply:

```
Wi    TopL  BotR  Pos    Cw    Bw    Fmt   Opt
P0    A1    I19          7     7     FRM
```

`Wi`	The window number.
`TopL`	The top left slot of the window.
`BotR`	The bottom right slot of the window. If the window is a single slot the top left and bottom right slot references are the same.
`Pos`	The position of the window on the printing area of 255 characters wide (maximum) by 255 lines deep. A window position is given in relation to another window in the form `B0` (meaning below window 0) or `R1` (meaning to the right of window 1). The window referred

to must always be numbered lower in the window sequence than the window being defined. It is impossible to define the position of a window in relation to a window with a greater number. Therefore no entry is needed for printer window P0 which is always in the top left of the print area. A blank line or a blank column is left between windows.

`Cw`	Column width: the width of the slots in characters within that window. Remember that ViewSheet adds an extra character to separate columns. The minimum column width is three characters and the upper limit is 253 characters.
`Bw`	Border width, ie the width of the row headings. The maximum is 15 characters, and the minimum is two.
`Fmt`	Value format, ie the number of decimal places displayed in the slot; rounding; right or left justification; minus sign or brackets with negative numbers. The rules are the same as when setting formats for individual slots. See chapter 9, 'Value formats'.
`Opt`	Options. One or more of the codes below may be placed under `Opt` to control row and column headings.
`S`	Side border off. Row headings are not displayed.
`T`	Top border off. Column headings are not displayed.
`C`	Substitute asterisks for the value, one asterisk for each unit of the value. This is the bar chart facility. Negative values are shown with exclamation marks.
`1`	Highlight 1
`2`	Highlight 2 The whole window is printed with the highlight facility active. The effect of this depends on the printer driver which is loaded at the time. For example, highlight 1 may give underlined type and highlight 2 bold type. (See the instructions with your printer driver program.)
`0`	Window off.

Note that options 1 and 2 cannot be used with screen windows.

Saving and loading windows

The `SW` and `LW` commands are described in chapter 2. To save a set of windows type `SW` followed by your chosen filename and **RETURN**. To load the windows again type `LW` followed by the filename and **RETURN**.

Loading printer drivers

Type: `PRINTER` followed by the name of the printer driver and **RETURN**.
Printer driver names are usually the name of the printer itself.
For example: `PRINTER RICOH` **RETURN**
Always load the printer driver before using the `*FX` codes to introduce printing and set the baud rate.

Printing out

To print out all print windows which are currently active, type: `PRINT` **RETURN**.

Whatever printer driver has been loaded will be used automatically, and if none has been loaded the default printer driver will be used.

Loading spreadsheets into VIEW

The `SCREEN` command allows you to save parts of the current sheet on disc or cassette in such a way that they can subsequently be loaded into the VIEW word processor. The procedure is as follows.

Load or compose a sheet.
Decide on a filename for the VIEW version of the sheet.
Set any *printer* windows you need – the `SCREEN` command will only work with printer windows.
Type: `*SPOOL filename` **RETURN**
Type: `SCREEN` **RETURN**
The sheet is displayed on the screen as it is loaded into the file.
Type: `*SPOOL` **RETURN**

The new file can now be loaded into VIEW using the `READ` command, and edited or incorporated into another document.

See the *BBC Microcomputer System User Guide* for information on the `*SPOOL` command and the *VIEW Guide* for information on the `READ` command.

SCREEN

The command `SCREEN` displays the current sheet on the screen with the current printer windows active. It can therefore be used as a rough check on the placing of the windows.

12 Using discs within the sheet

ViewSheet has the facility to `WRITE` information onto disc files in response to instructions entered in slots within the sheet itself, and to `READ` back information from those files. Every time the sheet is recalculated the disc files are accessed.

The information is written into two-dimensional array files which exist independently of files created by `SAVE` and `SW` commands.

Up to 10710 numbers may be stored in each of these array files and up to five files may be used in each sheet.

(Disc files used within sheets are discussed at greater length in Part One chapter 7.)

Creating disc files

Before a disc file can be used it must be created. This establishes its name, which follows a strict convention, and its format.

Type: `CREATE n x y` **RETURN**

`n` is the file number which is used in naming the file. Any number from 1 to 255 can be used. Up to five disc files may be used in each sheet.

`x y` are the maximum horizontal and vertical coordinates to be used with the array file. A file may not contain more than 10710 numbers.

On receiving the `CREATE` command ViewSheet creates a file named `V.VSn`. Thus files used within the sheet are instantly recognisable in the disc directory.

Example: `CREATE 7 18 8` creates the file `V.VS7` with dimensions 18 by 8.

Writing to disc files

The `WRITE` function puts a number in a specific place in a specific disc file. So the file, the coordinates and the number must be given in each `WRITE` command.

Position the cursor on the slot where you want the `WRITE` command.
Type: `WRITE (n,x,y,value)` **RETURN**

`n` is the name of the file set up by the `CREATE` command.

`x,y` are coordinates of the element in the array file where the number is to be recorded.

`value` is the value to be recorded. This may be a number, a slot reference, or a formula.

Example: `WRITE (7,2,10,A5F5/2.22)` writes into element 2,10 of a disc file called `V.VS7` the sum of slots A5 to F5 divided by 2.22.

When specifying disc file coordinates, it can save a good deal of time if you use the functions `ROW` and `COL` instead of numbers. So for example if you are writing from slots C11 to C20 into file coordinates 1,1 to 1,10 you can use commands such as `WRITE (7,1,ROW-10,A5F5/2.22)`. When this command is replicated, the coordinates will automatically be updated.

Reading from disc files

To read from a disc file the system must be told which file to read from and which element in that file.

Position the cursor on the slot where you want the `READ` command.
Type: `READ (n,x,y)` **RETURN**

`n` is the file into which numbers have been entered with the `WRITE` command.

`x,y` are coordinates of the element in the array file where the numbers are stored.

Example: `READ (7,2,10)` reads back the information entered in the example above.

Disc file efficiency

When a sheet includes `READ` and `WRITE` commands ViewSheet accesses the disc every time the sheet recalculates. This can become very tedious, but the tedium can be minimised as follows:

- If possible restrict the size of your files to 41 entries. The reason is that entries are six bytes long and 42 entries fit into a disc sector (41 in the first sector). The disc filing system keeps one sector in memory, so after the first disc access in the sheet no further access is needed, and speed is greatly improved.
- If the disc file spans more than one sector refer to the elements in the disc file in order from within the sheet, remembering that ViewSheet recalculates from left to right on each row starting at the top.
- If you cannot keep your disc file small and must have random access from all over the sheet, turn off automatic recalculation.

Running out of memory

You can run out of memory when replicating, entering a slot, or when ViewSheet is evaluating an expression.

In such cases the computer bleeps and the message `Memory` appears. If ViewSheet is recalculating a sheet, `?Error` messages will appear in the slots. This indicates stack overflow.

You may avoid the problem in two ways:

- Reduce the complexity of whatever expression is causing the problem.
- Delete parts of the sheet. Deleting whole rows and columns is more efficient for saving memory than deleting slots, since it reclaims all the memory used.

Disc file format

The following is for those who wish to pass values to ViewSheet disc files from another language or program system.

ViewSheet disc files are made up of six-byte entries. At the beginning of the file is a six-byte 'identification record' with the following format:

Byte	Contents
0	Any
1	Any
2	Any
3	Maximum y coordinate of file
4	Maximum x coordinate of file
5	&AA (identification byte)

File entries are stored from byte 6 onwards, the first entry being file slot 1,1, the second 2,1 followed by 3,1 and so on. In a 2 by 2 file the order of entries would be: 1,1 2,1 1,2 2,2.

The first byte of each entry is &FF. Then the five bytes of the floating point number are stored in reverse order. This value entry format is compatible with BBC BASIC.

The following BBC BASIC program prints out all the values of a 10 by 12 ViewSheet file:

```
 10 Z=OPENIN("V.VS1")
 20 FOR Y=1 TO 12
 30   FOR X=1 TO 10
 40     PTR#Z=((Y-1)*10+X-1)*6+6
 50     WW=0
 60     IF BGET#Z<>255 THEN GOTO 90
 70     PTR#Z=((Y-1)*10+X-1)*6+6
 80     INPUT #Z,WW
 90     PRINT WW;
100   NEXT X
110   PRINT
120 NEXT Y
130 CLOSE #Z
```

This file would be created using the command `CREATE 1 10 12`.

13 RETURN – and what happens when you press it

Pressing **RETURN**, as anyone who has had anything to do with computers will know, is the moment of truth.

In ViewSheet a great deal is happening in that moment and it can have many different effects. Sometimes a value is entered, sometimes a label, sometimes a label when you wanted a value, occasionally a value when you wanted a label; sometimes the computer bleeps and tells you something is wrong; sometimes it signals an error in the slot. All this and more can happen very quickly, and you need to be aware of what it all means.

The first thing the computer has to decide when you press **RETURN** is whether what you have entered is acceptable according to the rules. For example, if you type the number `1` and press **RETURN**, ViewSheet first decides whether it is a label or a value. Since it is a value ViewSheet goes on to display it, provided the slot involved is not protected.

On the other hand if you type the number `-12345.6789` and press **RETURN** the result may be different. Certainly it is a value so ViewSheet tries to display it. However, suppose the slot is seven characters wide and formatted `D2RB` (two places of decimals and brackets for negative numbers).

In such a case the number cannot be displayed – it would require a slot of ten character spaces to do that. But it is fully acceptable as an entry, so ViewSheet displays a percent sign in the slot and the full number in the `CONTENTS=` line above.

A different reaction still occurs if you type a formula such as `123/0` and press **RETURN**. ViewSheet first tests whether it is a value or a label. In a literal sense it is certainly a value since it consists only of numbers and an operator. So it passes the first test, but fails the next because such a ‘value’ cannot be computed, so ViewSheet signals an error. The word `?Error` appears in the slot and the message `Divide by 0` appears at the top left of the sheet.

It is important to realise that these last two errors are of a quite different kind. Dividing by zero is a mathematical operation that cannot be evaluated. If you type the letter `O` instead of `0` (zero), or the letter `l` instead of the number `1` you will probably find yourself entering a label. ViewSheet can warn you about bad arithmetic, but it cannot know that you really meant to type something else.

Pressing RETURN after typing in a reply to a prompt

The following are examples of the way in which ViewSheet checks what you have typed and acts on it. For a complete list of error messages see later in this chapter.

Window definitions

ViewSheet checks for:

- Correct numbering of window.
- Acceptable numbers in `TopL` and `BotR` (eg that the slots exist).
- Acceptable `Opt` characters.
- Acceptable column width.
- Acceptable border width.
- Acceptable format.

If all is correct the window is defined and used. If not, the computer bleeps and the text cursor is positioned at the incorrect part of the definition.

If the window can be fitted in as defined, it is used as it is. If not ViewSheet reduces it in size. If it still will not fit, ViewSheet turns it off.

Replication

ViewSheet checks for:

- Acceptable slots and ranges – eg that you have not specified a column range to be replicated to another specified column range, such as A1A10-B1B10.
- Protected slots.
- Correctly specified ranges – slots must be in the same column or row.
- Complete instruction to replicate, including the dash.

Error messages

Some error messages in command mode occur as a result of incorrect operation of the disc operating system (eg `Bad command`). You should look these up in the *Disc Filing System User Guide*.

`Bad file`	You have named a file which has the wrong format.
`Bad heading`	You have specified a row or column heading which is not valid, possibly because it already exists.
`Brackets`	There is an unequal number of opening and closing brackets in the expression.
`Command?`	ViewSheet does not understand what you have typed in.
`Divide by 0`	There is a division by zero within the expression.
`Edge`	You are trying to replicate beyond the edge of the sheet.
`File not found`	ViewSheet cannot find the file you have specified in command mode.
`FS error`	ViewSheet cannot `WRITE` to or `READ` from a disc in sheet mode because of a filing system error.
`LOG range`	The number specified as an argument to `LOG LN` and ^ is outside the valid range.
`Lookup`	There is no equal value in the lookup range, or no corresponding value in the result range.
`Memory`	A command has been used which requires more memory for the sheet than is available. This can occur during replication.
`No file`	The data file referred to in your `READ` or `WRITE` command does not exist on the disc.
`Not enough memory`	There is not enough memory to change screen modes or to load a sheet file. Can you reduce the size of the sheet?

No sheet	You have used a command which will work only if a valid sheet is in memory, but there is no sheet there, or the current sheet has been corrupted. Type: `NEW` **RETURN** to initialise the sheet.
Out of range	A function whose argument is limited to 255 has been given an argument greater than this; or an attempt has been made with the `READ` or `WRITE` command to access an element outside the size of a random access disc file.
Overflow	Stack overflow has occurred. Either you are using an excessively complex expression or you are running out of memory.
Propagated	A reference is made to a slot containing an error.
Protected	You have tried to edit or delete a protected slot, or to delete a protected row or column.
Range	A non-valid range has been specified, eg descending rather than ascending.
Syntax?	You have used a valid command, but the information you have typed in with it is incorrect.
Too big	ViewSheet has encountered a number which is too big for it.
Too few arguments	You have used `CHOOSE`, `LOOKUP`, `READ`, `WRITE` or `IF` and have omitted one or more of the arguments which should follow.
Too long	The expression you have just entered is too long.
Too many files	Your sheet is trying to use more than five `READ`/`WRITE` files at one time.
-ve root	There is an attempt within the expression to calculate the root of a negative number. This may occur with invalid arguments to some trigonometric functions.

14 Additional notes

The *FX calls

We have mentioned many of these already.

`*FX5`	To specify a serial or parallel printer. See chapter 11.
`*FX6`	To determine the character that will be suppressed by the printer driver routines. The default setting is `*FX6,10` – which suppresses line feed characters. If you wish line feed characters to reach the printer you should use this command to cause the computer to suppress another character you do not need – `*FX6,255` for example will suppress character 255, and so allow the line feeds through.
`*FX8`	Sets the transmission rate to the printer. See chapter 11.
`*FX11`	Sets the delay for the auto-repeat facility. If you hold down a key the character auto-repeats on the screen, after a short delay. Similarly function key commands auto-repeat after a short delay. You may reset this delay by typing: `*FX11,X` **RETURN** where X is the delay in hundredths of a second.
`*FX12`	Sets the auto-repeat rate. If you hold down a key the character auto-repeats on the screen after a short delay. Function key commands auto-repeat similarly. You may reset the auto-repeat rate by typing: `*FX12,X` **RETURN** where X is the delay between successive characters in hundredths of a second. `*FX12,0` **RETURN** resets both the delay and the repeat rate to the default values.

The `*FX` calls are more fully described in the *BBC Microcomputer System User Guide*.

Colours

If you use a colour monitor or a television set you may wish to take advantage of alternative screen colours.

Both text and background colours can be changed in all screen modes except `MODE 7`. The colours available are:

0 – black
1 – red
2 – green
3 – yellow
4 – blue
5 – magenta
6 – cyan
7 – white

These may be used as follows in command mode.

For a green screen effect:
Hold down **CTRL** and press `S`
Press: `7 2 0 0 0`

For white text on a blue background:
Hold down **CTRL** and press `S`
Press: `0 4 0 0 0`

For black text on a white background:
Hold down **CTRL** and press `S`
Press: `0 7 0 0 0`
Hold down **CTRL** and press `S`

Press: `7 0 0 0 0`

The first digit indicates whether it is text or background colour that is to be changed. 0 means background and 7 means text. The second digit indicates the colour, as in the table above.

The Second Processor

As you work with ViewSheet you will probably come to use more and more information in your spreadsheets. If you finally decide you need more memory you should consider the Acorn 6502 Second Processor. This provides 30K of memory in any screen mode, so if you use `MODE 3` the memory available will increase by about three times.

Index

Notes

Notes

ViewSheet was designed and developed by Protechnic Computers Limited, Cambridge, England.

The ViewSheet User Guide was written by Information Transfer Limited, Cambridge, England.

SBD15

ISBN 0 907876 82 X

Sheet mode flags

L	Label
V	Value
A	Automatic recalculation
M	Manual recalculation
R	Auto entry – right
D	Auto entry – down

Window definitions

W1	Window number
TopL	Top left slot of window
BotR	Bottom right slot of window
Pos	Position in relation to other windows: use R (right), B (below)
Cw	Column width
Bw	Border width
Fmt	Format: F (floating point) D (decimal places) R or L (justification) M (minus) or B (brackets)
Opt	Options: H (horizontal scroll) V (vertical scroll) S (side border off) T (top border off) C (bar chart) 0 (window turned off) 1 and 2 (print highlights)

Note that options H and V apply to screen windows only and 1 and 2 to printer windows only.

To save and load window definitions only:

SW filename	Saves all window definitions set
LW filename	Loads the definitions and applies them to the sheet in memory

Replication

Slot to slot
For example A1-B3 replicates slot A1 to the slot B3

Slot to range
For example A1-B3B10 replicates slot A1 to the range B3 to B10

Range to slot (to replicate column to column or row to row)
For example A1A26-B1 replicates the range A1 to A26 to the range beginning B1

Column range to row range (to replicate column to range of columns)
For example A1A26-B1I1 replicates the column A1 to 26 to the ranges beginning B1 to I1

Row range to column range (to replicate row to range of rows)
For example A1I1-A2A9 replicates row A1I1 to the rows beginning A2 to A9

Function keys

	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9
CTRL ▶	AUTO ENTRY	DELETE COLUMN	DELETE ROW							
SHIFT ▶	RECALCULATE MODE	INSERT COLUMN	INSERT ROW	COLUMN HEADING	ROW HEADING	PROTECT COLUMN	PROTECT ROW	RECALCULATE	JUSTIFY LABEL	DELETE SLOT
ViewSheet	REPLICATE	EDIT WINDOW	NEXT WINDOW	DELETE END OF LINE	BEGINNING OF LINE	END OF LINE	EDIT SLOT FORMAT	GO TO SLOT	INSERT CHARACTER	DELETE CHARACTER

CTRL f0 Causes the cursor to move to the next slot after entry
SHIFT f0 Switches automatic recalculation off and on
f0 Replicates specified slots or ranges

CTRL f1 Deletes the column that the cursor is in
SHIFT f1 Inserts a column to the left of the sheet cursor
f1 Displays the definition for the specified window

CTRL f2 Deletes the row that the cursor is in
SHIFT f2 Inserts a row above the sheet cursor
f2 Moves the cursor to the next screen window

SHIFT f3 Replaces the column heading with a user-defined heading
f3 Deletes all characters to the right of the cursor on the editing line

SHIFT f4 Replaces the row heading with a user-defined heading
f4 Moves the cursor to the beginning of the editing line

SHIFT f5 Turns on/off protection for the column the cursor is in
f5 Moves the cursor to the end of the editing line

SHIFT f6 Turns on/off protection for the row the cursor is in
f6 Sets the value display format for the current slot

SHIFT f7 Causes the sheet to recalculate
f7 Moves the cursor to the specified slot

SHIFT f8 Justifies a label to the right/left
f8 Inserts a character space in the editing line

SHIFT f9 Deletes the contents of the current slot
f9 Deletes a character in the editing line

Other sheet mode key commands

Arrow keys	The four arrow keys move the cursor a slot at a time
SHIFT with arrow keys	Moves the cursor a whole window width or depth
CTRL with arrow keys	Moves the cursor along the editing line
COPY	Copies the contents of the current slot into the editing line
SHIFT COPY	Copies the coordinates of the slot where the cursor is into the editing line
DELETE	Deletes the character to the left of the cursor on the editing line
ESCAPE	Escapes from the editing line – when the editing line is not in use, switches between sheet mode and command mode

Loading spreadsheets into VIEW

Set any printer windows you need.

In ViewSheet type:

`*SPOOL filename` RETURN

Type:

`SCREEN` RETURN

The sheet is saved to the file.

Type:

`*SPOOL` RETURN

To use the file in the VIEW word processor switch into VIEW and use the `READ` command.

ACORNSOFT

VIEWSHEET

An electronic spreadsheet
for the BBC Microcomputer

Reference Card

Starting ViewSheet

To enter ViewSheet from VIEW, BASIC or another language, type `*SHEET` and press **RETURN**.

Command mode

(minimum abbreviations shown in brackets)

`*ACCESS filename (*A.)`	Unlocks specified disc file
`*ACCESS filename L`	Locks specified disc file
`*BASIC (*B.)`	Switches computer to BASIC
`*CAT (*.)`	Displays contents of disc or cassette
`CREATE n x y (C)`	Creates a disc file V.VSn with dimensions x and y for `READ` and `WRITE` within the sheet
`*DELETE filename (*DE.)`	Deletes the specified file from the disc
ESCAPE	Switches between command mode and sheet mode
`HEADINGS (H)`	Indicates if user-defined headings are set
`HEADINGS OFF (H OF)`	Switches off user-defined headings
`HEADINGS ON (H ON)`	Switches on user-defined headings
`LOAD filename (L)`	Loads the specified file from disc or cassette
`LW filename`	Loads a file of window definitions from disc or cassette
`MODE number (M)`	Sets the screen mode specified
`NAME filename (NA)`	Names (or renames) the sheet in memory
`NEW`	Creates a blank worksheet
`PC`	Prints out the contents of every occupied slot, with coordinates
`PRINT (P)`	Prints out the sheet in memory
`PRINTER filename (PRINTE)`	Loads a printer driver
`PROTECT (PRO)`	Indicates if protection is enabled or disabled
`PROTECT OFF (PRO OF)`	Disables protection
`PROTECT ON (PRO ON)`	Enables protection
`SAVE filename (S)`	Saves the sheet in memory to disc or cassette
`SCREEN (SC)`	Displays the sheet in memory with current print windows
`SW filename`	Saves a file of window definitions
`*WORD`	Switches computer to VIEW

Note that disc filing system commands such as `*COPY` and `*OPT` can all be used in command mode.

Operators – order of precedence

Group 1	-	Unary minus		
	( )	Brackets		
Group 2	^	Raise to power		
Group 3	*	Multiply		
	/	Divide		
Group 4	+	Plus		
	-	Minus		
Group 5	=	Equals	>	Greater than
	<>	Not equal to	<=	Less than or equal to
	<	Less than	>=	Greater than or equal to

For example: `A1+B2*C3^D4` is equivalent to `A1+(B2*(C3^D4))`

Functions

`ABS(n)`	Absolute value of n
`ACS(n)`	Arccosine of n (radians)
`ASN(n)`	Arcsine of n (radians)
`ATN(n)`	Arctangent of n (radians)
`AVERAGE(list)`	Average value in list
`CHOOSE(n,list)`	Selects nth value in list
`COL`	Value equal to column number
`COS(n)`	Cosine of n (radians)
`DEG(n)`	Converts n from radians to degrees
`EXP(n)`	Exponent of n
`IF(X,Y,Z)`	If X is true then Y is the value of the function; if not then Z is the value
`INT(n)`	Integral value of n
`LN(n)`	Natural logarithm of n
`LOG(n)`	Logarithm to base 10 of n
`LOOKUP(X,Y,Z)`	If value X occurs in range Y the function's value is the corresponding value in the range Z
`MAX(list)`	Maximum value in the list
`MIN(list)`	Minimum value in the list
`PI`	3.141592653
`RAD(n)`	Converts n from degrees to radians
`READ(n,x,y)`	Reads from position x,y of disc file V.VSn
`ROW`	Value equal to row number
`SGN(n)`	Sign of n
`SIN(n)`	Sine of n (radians)
`SQR(n)`	Square root of n
`TAN(n)`	Tangent of n (radians)
`WRITE(n,x,y,z)`	Writes z to position x,y in the disc file V.VSn