SEND IN THE CLONES

In the first instalment of this series, we took a preliminary look at Vu-Calc, a simple spreadsheet modelling package for the Sinclair Spectrum and BBC Micro. Here we discuss how to use Vu-Calc's features to carry out calculations of mortgage repayments or bank loans.

The great strength of spreadsheet programs — even a simple package like Psion's Vu-Calc — is the way in which they allow complex formulae to be applied to data. As we shall see, it is possible to build up some interesting and useful models with Vu-Calc, despite the fact that this particular program supplies almost nothing in the way of built-in formulae. In fact, Vu-Calc's sole built-in formula is its ability to 'sum' (i.e. add together the contents of) blocks of cells; this is indicated by prefixing the cell address with an @ sign.

More advanced spreadsheets contain very elaborate built-in formulae, which the user can call up by name. The advantage of such a system is that you don't actually have to know the mechanics of how these formulae work. If you want to use a mortgage formula with Multiplan, for example, to calculate expected repayments on a house purchase over different repayment periods — say 15, 20 and 25 years — you simply call up the formula and enter the relevant data. Multiplan then works out all the answers.

With Vu-Calc, the same calculations take considerably more time and effort. You must construct the required formulae yourself, and then key them into the machine. Vu-Calc also imposes a number of constraints on the user. It has a maximum of 28 columns, so the largest model you can build, with each column representing one month, will cover a period of just over two years. Accuracy can also be a problem — Vu-Calc works with integer (whole number) values only, and simply ignores the figures after a decimal point, so 99.9 would be considered as 99. Vu-Calc does allow values and arithmetical operations to be entered at any point on the model. For example, if the cursor is located at a blank cell (H5, for example), you can enter 500*2 in the command line. Pressing the Enter key will display the result — 1,000 — in cell H5.

Another irritating Vu-Calc feature is the way in which formulae are edited. A 'smart' package such as Lotus 1-2-3 uses a function key for editing. Pressing the key automatically puts the contents of the cell containing the cursor into the command line. Vu-Calc has an EDIT command (#E) that is used if a formula needs to be changed, but the formula must be retyped each time the edit facility is used. If you are working on a long formula and realise that you have forgotten to enter a bracket, there is no way of simply inserting it — instead the entire line must be retyped. All the EDIT command does is to tell the program to erase the old formula from a cell and then insert the new one.

However, using the REPLICATE command (#R) with a formula allows some fairly complex modelling to be done. Let's suppose that you want to extend the household budget example (see page 692) to anticipate inflationary increases in the household grocery budget, assuming a steady inflation rate of 0.5 per cent per month. Performing the necessary calculations with pencil and paper would clearly be a time-consuming task. With Vu-Calc it can be done quickly by using a formula and the REPLICATE command.

To carry out the desired operation, you must tell Vu-Calc to 'grow' your initial monthly budget (say £200) by 0.5 per cent. More sophisticated spreadsheets make this easy by using a GROW BY command, but Vu-Calc requires that the user enters the arithmetical operations that must be carried out. In order for Vu-Calc to recognise a formula that contains cell addresses, the formula must be prefixed with either $ or %. These are two arbitrarily chosen symbols that have nothing to do with dollars or percentages, but tell the program that cell addresses are significant in the formula under consideration, and these addresses are either relative (%) or absolute ($). An absolute cell reference tells Vu-Calc to look for and act on the value in a specific cell, regardless of that cell's position.

To see what a 'relative address' does, let us return to our example. The formula for 'growing' the budget by 0.5 per cent is %B3*100.5/100, where % indicates a relative cell address and B3 is the address of the cell containing the value representing the monthly food budget. Having keyed this formula into cell B4, we then need to copy the formula to get the result for the full year. B4 will display the numeric result of the formula — the formula itself appears at the bottom of the worksheet when the cursor is at B4. The REPLICATE command #R,B4,B5:B14 gives us the desired result (B4 contains the formula, B5:B14 defines the range of cells across which replication occurs). The results are shown almost instantly, and our spreadsheet model will look like this:

	1	2	3	4	5
A			JAN	FEB	MAR
B	Food budget		200	201	202