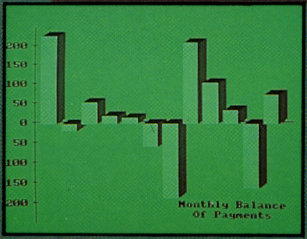
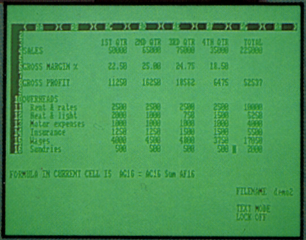
DOING BUSINESS WITH YOUR

AMSTRAD

CLIVE GIFFORD



```
(+Pitch)Dear Sirs
The AMSTRAD PCW8256 has revolutionised the way
prospective purchasers can compare prices betw
specification wordprocessor and an electric ty

With 256K of RAM, built-in disc drive, monitor
wordprocessing software, and complete CP/M Plus computer
power, Amstrad will speed up the way in which the
microcomputer takes away the drudgery of office routine.
Simply, efficiently, completely.

Open the box.  Fit on a single mains plug.  Plug in the
keyboard and printer to the screen unit.  Plug in.  Switch on.
Put in the paper.  Insert the master disc.  You are now ready
to start wordprocessing, and Amstrad has already exploded the
first myth.

With the 'user friendly' software of the PCW8256,
wordprocessing need be no more complicated than an electric
typewriter - although as you absorb more of the fundamentals
and skills, then you can progress further to take advantage of
some of the most powerful wordprocessing software available
anywhere.

For AMSTRAD CONSUMER ELECTRONICS plc
```


```
                     1ST QTR  2ND QTR  3RD QTR  4TH QTR    TOTAL
SALES                  50000    65000    75000    35000   225000

GROSS MARGIN %         22.50    25.00    24.75    18.50

GROSS PROFIT           11250    16250    18562     6475    52537

OVERHEADS
  Rent & rates          2500     2500     2500     2500    10000
  Heat & light          2000     1000      750     1500     5250
  Motor expenses        1000     1000     1000     1000     4000
  Insurance             1250     1250     1500     1500     5500
  Wages                 4000     4500     4800     3750    17050
  Sundries               500      500      500      500     2000

FORMULA IN CURRENT CELL IS  AG16 = AC16 Sum AF16

                                                  FILENAME  demo2

                                                  TEXT MODE
                                                  LOCK OFF
```

COMPATIBLE CPC 464, CPC 664,
CPC 6128 and PCW 8256

DOING BUSINESS WITH YOUR AMSTRAD

CLIVE GIFFORD

Best Wishes,

C. Gifford

Virgin

First published in Great Britain in 1986 by Virgin Books,
328 Kensal Road, London W10 5XJ

ISBN 0 86369 146 3

Printed and bound in Great Britain by Richard Clay Ltd
(The Chaucer Press), Suffolk

Typeset by Keyline Graphics, London NW6

Cover and design by Sue Walliker

Distributed by Arrow Books

CONTENTS

Acknowledgements

Thanks, first and foremost, to Scott Vincent without whom the book would be a mere shadow of its present self; and to the people at Gemini, Cumana, Tasman, Camsoft and Michael Joyce Consultants for their kind assistance.

On a more personal note, I would like to thank the members of the Ashford Hockey Club, especially John, Yazbi, Steven and Mark. Cheers to Keith, Andrew, Andrea, Gill, Don and Lynne, as well as everyone else at Southampton University.

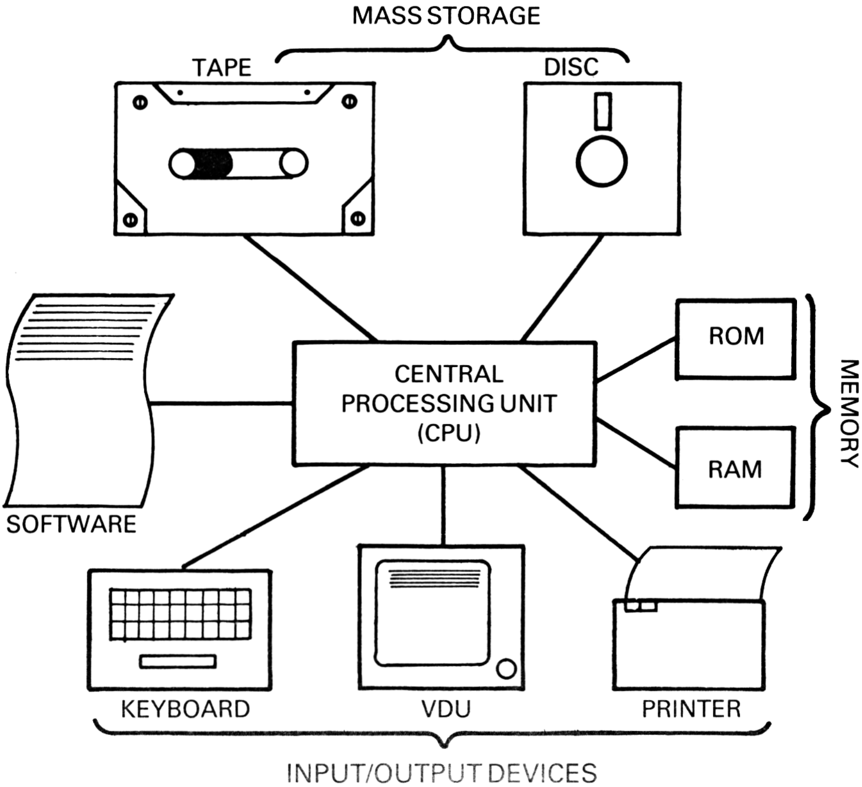
And a big thank you, as always, to Catherine, my family, and all at Virgin.

Bytes And Pieces

Those people with a reasonable knowledge of computers and computing may be best advised to miss out this chapter and start at the next. This chapter is essentially for beginners. They may already have an Amstrad computer or access to one. Alternatively, they may be considering buying one, but they will have one thing in common: the computing world will seem complex and confusing to them. ROM, RAM, bits, bytes and megabytes, floppies, hard copies, daisywheels and dot matrices all conspire to make life even more difficult and bewildering for those new to computing. The purpose of this chapter is, in part, to help the beginner by shedding light on some of the jargon and explaining what an Amstrad computer is and what it can do.

First let's look at the principal components of a computer. The following diagram (reproduced by courtesy of Interface Publications) shows the three essential components of a microcomputer such as the Amstrad: a Central Processing Unit; several kinds of memory; and a range of input and output devices.

The Central Processing Unit (or 'CPU') is the 'brains' of the computer. This is where all instructions are carried out. It consists of thousands upon thousands of miniature electronic circuits full of switches which respond to the instructions with YES or NO answers. Thus, for a complex operation to be carried out, hundreds of switches have to register a series of YES/NO answers and this creates a vast amount of circuitry.



THE PRINCIPAL COMPONENTS OF A COMPUTER

The binary system of numbering (every digit either a 1 or a 0) corresponds tidily to the workings of these YES/NO 'gates' and so is used by the processor as a means of representing the workings of the computer. One now has a system of programming a computer using certain combinations of ones and noughts to represent instructions and programs.

Communicating with a processor in ones and noughts is very tedious and slow. What is needed is some kind of interpreter to give a selection of commands that are quicker to input and more easily understood by the user. These commands can then be converted back into the original binary coding once they are entered into the machine.

Originally, an interpreter created what is known as a 'low-level language' or 'machine code'. This is certainly one step better than entering large numbers of binary figures, but is still difficult to learn, slow to program in and results in the computer crashing if one piece of data or one instruction is entered incorrectly.

'High-level languages' are a further stage up from binary. They do not suffer from the problems of machine code but are less versatile than a low-level language. Commands from the high level language have to be taken by the interpreter, converted into machine code and then converted back into binary. This obviously takes longer than the translation of machine code instructions into binary, and so high-level languages always run more slowly than machine code. However, the difference in speed may not be of any importance in many programs. Examples of high-level languages include FORTH, PASCAL, COBOL and the language that you are most likely to be familiar with as it is supplied with the Amstrad machines, Beginners All-purpose Symbolic Instruction Code, or BASIC.

The switches contained within a processor used to consist of electrical relays which consumed much space and money and resulted in a processor far slower and less powerful than today's models. Current processors are embedded in tiny slices of silicon holding many thousands of circuits within the area of a fingertip. Among the better known 'silicon chips' used with microcomputers today are the Zilog Z80, the Motorola 6809 and the Mos Technology 6502. All the Amstrad computers use the proven Z80 chip as their powerhouse.

Memory is vital to the functioning of a computer. It is necessary to store the code for the instructions and for data and program information. As a computer buyer, one of the most important criteria for choosing a micro is 'How much memory does the machine have?' The greater the memory of a computer, the more room it has for larger and more complex programs. It's not so long since hobbyists were quite happy with the 1K or 2K machines — now 32K, 64K and even 128K are considered minimum amounts of memory for new microcomputers.

An important point concerning memory is that it doesn't provide an absolute standard; more memory does not necessarily make machine X any better than machine Y. The Amstrad machines are well-equipped in memory terms with the two lower-range machines, the 464 and the now-discontinued 664, boasting 64K of RAM.

In the last two paragraphs we have already encountered a few classic pieces of computer jargon. What is a 'K'? It is, in fact, a term which, when used with memory storage, stands for the number 1024. Therefore, 32K means 32768 units of memory, each unit known as a byte. The term RAM is simply an acronym for Random Access Memory and is the type of memory that is left free for the user to write programs in. It can be filled with a program or with the data for a program to work on.

The other type of internal memory is ROM, or Read Only Memory. This memory is filled with the interpreter and programs that make up the operating system of the computer. This memory can only be looked at (or read) by the user via the computer and cannot be overwritten.

The need for external memory (that is, some sort of medium which can store programs or data outside of the computer for use at a later date) has been filled by a number of devices. In the early days, paper tape and punched cards were the best thing available but they have been superseded by faster and more versatile cassette tapes and discs.

Cassette storage of programs is undoubtedly the most popular form of external storage today among home computer users, but it is significant to note that three out of four of Amstrad's machines boast an integral disc drive. Cassettes have the distinction of being very cheap but they are not as fast as discs. Neither can they be read in any order other than sequential. Discs have the advantage of being faster and can be read in any order. This 'random access' is particularly useful for the handling of data, rather than for the loading of a certain program.

There are other storage media; bubble memories, ROM cartridges, EPROMs, Microdrives and Winchester discs are just some of the alternatives. The three that you will come into most contact with as an Amstrad owner or user are cassettes, floppy discs and ROMs.

Having looked at the CPU and the various types of memory, both internal and external, we now come to the third feature of a computer, the facility for the input and output of information or instructions. It is all very well having a machine that will perform calculations for you, but what if the machine doesn't provide a way for you to enter the data into the memory where it is to be acted upon? Furthermore, what happens if the machine performs the calculations but has no way of outputting them to the user? There would be no point in the computer making the calculations if it didn't share the answers with you. Obviously, then, some method of inputting information into the computer and receiving some reply from it is essential.

Of the various inputting devices available, the keyboard is the most important and the most common. Other devices include light pens, joysticks, digital tracers, optical readers and speech synthesisers. The standard output device for use with computers of all types is the visual display unit or 'VDU'. Many home computers make use of a television to display the information — Amstrad owners are lucky to have a monitor which has a better picture resolution than a television. Another output device which is rapidly becoming essential for all small business computing concerns is the printer. In a later chapter on hardware I will be discussing the various types of printer and the benefits that they can bring (see page 13).

To conclude, a computer consists of the CPU, the internal memory and the basic input and outputting functions. Other devices such as mass external storage and printers are looked upon as 'peripherals'.

There is one element that we have not mentioned. Yet without it, a computer is nothing more than a dormant electrical device. This vital component is software. Without programs, the computer is nothing; it needs to be *told* what to do before it will do it, and programs provide this all-important stimulus. There are various kinds of software which carry out many different tasks with varying degrees of success. These will be looked at in the chapter dealing specifically with software later in this book (see page 26).

What, then, can a computer do? Well, we already know that it has to be told what to do through a program which is made up of various commands drawn from the chosen computer language. This means that a computer is restricted both by the limitations of its own hardware (a computer without sound hardware is going to find it difficult to play Beethoven's Fifth), as well as by the limitations of its software. If the language being used does not take into account features that the computer is capable of producing, then these features cannot be accessed and exploited.

Computers depend on the software produced for them. For the businessman a whole host of useful and time-saving programs are available from databases to word-processors, from payroll packages to promotional displays. A computer is not a god, it is just another electronic device, yet used with sense and with suitable software, it can transform the working situation.

Questions, Questions

This chapter could be subtitled 'The whys and wherefores of introducing a computer into your business'. Firstly, let's look at why you intend to use a computer in your business. Here, a few hard decisions have to be made. Do you want a computer to speed up a particular area of your business operation and, if so, can that operation effectively be made less time-consuming using an Amstrad computer? Are you considering a computer to enhance your business image to customers or to other businesses in your field? If the answer to this is 'yes' then consider how the computer is to improve your image and whether thinking of buying a computer is merely obscuring the main problem of poor business performance with few real ideas of how to remedy it.

There are other 'wrong' reasons for using a computer in your business: 'Joe Bloggs has got one and Fred Smith has got one, so I *must* need one.' Rubbish! Their needs may be different and they may also be choosing a computer for the wrong reasons. Do not dismiss this as something that will never happen to you. So many worried small businessmen talk about needing a computer to improve their business, but often they are unsure as to exactly why they want a computer and how it will help them. Establish a reason for wanting a computer, a reason that is true, valid and watertight, and that can in some way be fulfilled by the use of an Amstrad computer and software.

There are other reasons for wanting to use a computer in business. These are not specifically inappropriate, but need to be looked at seriously as they could prove more damaging than beneficial to your business.

'I want to learn more about computers and by incorporating one in my working environment, doing a few chores, I will be able to understand it more while it helps my business.' A fine goal, admittedly, but does it hold water in the business situation? Well, if the person in question defines some genuinely important 'chores' that the computer can do and these chores can be simply and effectively performed by an existing piece of software that is straightforward to use, then this reason becomes very valid indeed. But to expect the computer, with the minimum of effort on your part, to suddenly leap to various tasks and perform them perfectly is a naive assumption. One of two things tends to happen in these cases: either there is an immediate cash-loss and a small amount of time wasted when the businessman realises that it isn't as simple as he thought and he gets rid of the system at a reduced price. Or, alternatively, there is a larger and more dangerous time-loss as the businessman ploughs on with the system, trying to get it to work and trying to understand the processes but all the time getting himself into more and more trouble with the result that he spends less and less time in the main business.

Another reason, rather different to the previous ones, can be put forward. Indeed, it may co-exist alongside one of the other reasons. The businessman may already have bought an Amstrad computer, for a hobby, or for his family, with the thought at the back of the mind that he could possibly use the computer in some form in his business. This is not very different from the previous reason, except that the computer and some software has already been bought and justified in non-business terms. The computer system does not need to be anywhere near as cost-effective as a specifically-bought system, as only the software for the new applications has been bought with the business's money. This is undoubtedly a fair point, but on the other hand a note of caution is needed. The same mistake of not using the computer effectively or to help in one particular field may still occur but now there may be a tendency to worry even less about the system's relative inactivity and possible misuse. 'Oh, it's not part of the business anyway,' may be the response which guarantees that no real effort be made to sort out teething troubles, resulting in a very short working life for the Amstrad.

We've looked at the most likely underlying reasons for wanting to use a computer in your business and, more specifically, an Amstrad computer. I don't feel that there is any need to dwell on the range of Amstrads and their differences (I have included brief comparisons at the back of the book). Nor do I feel there is any need to talk at length about the different types of hobbyist and small business. Throughout the book, the statements made will, as far as possible, apply to all the Amstrad range and to all business situations. There *will* be some exceptions, but these will be kept to a minimum.

What we do need to look at in definite terms are the particular uses the Amstrad can be put to in a business situation and the resultant benefits it can bring.

Tony Smithies, in an article in Prentice-Hall's *The Small Business Computer Guide* (pub. 1981), made some very salient points which I have reproduced here. One should seek at least some of the following:

1) **Improved productivity** through, for example: better production planning/scheduling; improved utilisation of materials; direct improvements in labour and/or equipment efficiency.

2) **Indirect cost savings or cost containment** through: reduction in inventories (raw materials, WIP, finished goods); improved product cost; earlier or more regular invoicing for sales; tighter control over creditors' payments.

3) **Improved cash flows** by, for example: earlier or more regular invoicing for sales; tighter control over debtors' payments.

4) **Reduction of bad debts** through improved credit control.

5) **Improved customer service** as a result of: prompt and accurate response to general orders and enquiries; accurate price quotations; helpful product-knowledge assistance; accurate production/ delivery estimates.

6) **Greater sales penetration** through use of improved product/ customer sales information. Combined with accurate product cost/profit contribution data, continuous product reviews and rationalisation may be possible.

7) **Improved overall financial and management control** as a result of responsive budgeting, forecasting and performance-assessment tools.

This is a very formal assessment of the potential benefits of using a computer in one's business, but it is nonetheless true for small business machines and much of it is valid for situations in which the Amstrad may find itself. But these benefits need to be qualified a little. The obvious ones are those that can be measured, often in financial terms. For example, having a word-processor may cut the cost of having specially prepared stationery or getting a typist to type out the same letter half a dozen times. To these *tangible* benefits can then be added the *intangible* benefits, such as having your data all in order and easily accessible.

We can see the overall benefits of obtaining a computer for use in a business, but what specific areas of the business can actually involve the use of a computer? This question is answered in two separate ways. Here we discuss situations where a computer might prove beneficial while on page 26 we look at the various types of programs around.

You will use a computer for business purposes in one of two situations:

1) Where a computer can do the job more effectively and efficiently (this usually involves the performance of a very tedious and repetitive job).
2) Where a computer can do things that people would find impossible or impractical to attempt themselves. This latter role includes complex financial accounting.

A situation that one should consider computerising or, at least, aiding with a computer, will include some or all of the following elements:

1) A thoroughly comprehensible and well-defined process. If it can't be described in detail in simple English and using simple instructions, then how will the computer understand?
2) A need for structured and analysed information, a feat which is very difficult or impossible to carry out by non-computer methods.
3) A significant volume of data records and/or documents with a similar/shared format or usage.
4) The need to use information in more than one way a number of times, particularly if this places a strain on the present method of operation.
5) A potential for growth generating more work in the future.
6) The benefit of obtaining one common record, centrally positioned and available to everyone that might be interested in the business, so that problems of consistency or clarity can be solved.
7) Frequent information changes involving possible additions and deletions and the maintenance of an up-to-date record.
8) Consistent and repeated application of similar defined routines.
9) Automatic reminders, based over time or over an event.
10) Collation of existing information or new information on a regular basis or in response to an enquiry.

Finally, a cautionary note. There is a great chance of failure if:

1) Neither you nor the supplier has done this with a computer before.
2) Nobody else in the industry has used this particular application before.
3) You have trouble defining your actual requirements, even with the help of the supplier.
4) It requires a proportionately 'large' project with many dependencies. The bigger an operation, the greater the risk.

Let's take a brief look at the varying types of package available to the businessman. The 'Big Three' should be mentioned first. The Big Three consist of word-processors, databases and spreadsheets. I assume that readers are familiar with. what a word-processor and a database are. A spreadsheet is a calculating device which allows you to manipulate formulae and data for financial, mathematical and scientific purposes. If you need to do any financial modelling then there is a very good chance that you will need to use a spreadsheet program. Some packages have been created which incorporate all three programs into one big package. This has a far greater benefit than the mere convenience of having all the programs immediately 'on tap'. With all the programs together, data from one can be transferred to another. Therefore a manager can look up a client's account from the database, juggle with it numerically using the spreadsheet, and then write a memo or a letter to the client on the word-processor, including an account of the financial manipulation that has taken place.

Despite the Big Three's dominance of computer programs for business, there are a number of other potentially useful general packages. There is a large range of financial packages which deal with company accounts, VAT returns, payroll, and budget planning. Programs enabling you to construct mailing lists with standard listings often appear as enhancement packages to either word-processors or databases, and there is an enormous range of more specialist business programs including draughtsmen's programs, stress calculation for civil engineers, insurance quotation tables, stocks and shares predictors and so on.

Should you trust a large part or all of your business to a large-scale computer operation? The answer is, if you're setting up the system yourself, no. Even if a competent dealer, supplier or consultant is helping you, I would still advise you to computerise just one aspect of your business to start with. By monitoring this one aspect carefully, you should be able to gauge the success of your project and then decide whether any other sectors of your business could benefit from being computerised. Looking pessimistically at the project, if the system breaks down (I do not mean this in the mechanical sense of the word — a computer system's physical parts are generally far more reliable than the software or the person using the system) then it will only affect one area of your business operation. Putting all your eggs in one basket is always a risk, particularly when the basket has not been tried and tested.

There is one application which is really an anomaly in the business field as it can be run as a straight replacement for another method of operation with relatively little change. This application is word-processing with, in this case, an Amstrad connected to a suitable printer (discussed in a later chapter — see page 21) in place of a more conventional typewriter. As I write this chapter, I am using a word-processor set-up, namely an Amstrad CPC 464 with disc drive, running Tasword 464-D, linked to a Canon PW1080A printer.

A word-processor can be set up, the user trained and the system put into operation in a very short time indeed. The cost of setting up a word-processing system is not much higher than a good electronic typewriter, and about the same if you choose the 8256 computer word-processing package available from Amstrad. Creating a system such as this relies on some, but not all, of the points made in this chapter and others. I have singled it out here because many of the points to come do not apply to a system intended to be used *just* as a word-processor.

The following three chapters need to be read together as they refer to each other frequently. Also, much of what is in one chapter applies to another.

Hardware Selection

The most fundamental piece of hardware that you need is obviously the computer itself. As an Amstrad owner or as someone who has decided to buy an Amstrad, your first hardware selection hurdle has been cleared — or has it?

You have an Amstrad machine in mind from Amstrad's range. Is it necessarily the right one for the job? Does it have the memory capacity you need? Will it run the software you want to use? Hopefully you will have considered these and other, similar questions before embarking on your purchase. If you did not, or if you bought the Amstrad not just for business purposes, do not worry. Independent companies, as well as Amstrad, are supplying peripherals that will turn your particular Amstrad into almost any one of the other machines. What you can be relieved about is that you bought an Amstrad machine, for that guarantees you a high level of company support for peripherals, software and service.

With your choice of computer out of the way, let's look at the various other pieces of hardware that you may need to purchase for your system. Essentially, these can be grouped into three major categories: outputting devices (printers, primarily); external storage and memory upgrades; and input devices.

Printers

The most essential peripheral that you, as a businessman, will consider. A printer will give you a 'hard copy' on paper of details from the screen. The range of specific uses that the printer can be put to is very large indeed and includes report generation, the presentation of statistics and graphs, multiple letter writing and mailing list compilation. Just as important is the facility to transfer information from the computer's memory to an easily portable form, paper. Printing options are available and necessary in most accounting, financial, database and spreadsheet programs, so, without a printer, you are not making use of the full potential of a business system.

You can obtain a new printer for your Amstrad machine from as little as £80. Look out for second-hand buys, but be warned, many printers are sold because the original buyer finds that the printer lacks the power that they had expected. There are more details of some good printer buys at the end of this chapter.

Let's look at the varying types of printer. I have split them up into three separate groups: dot matrix; daisywheel; and other printers. We'll take each in turn.

The dot matrix printers are the most popular type in the small microcomputer market. They form their characters on paper using a grid of pins or dots. This leads to a fast printing speed but poorer quality printing than you would find on a typewriter. You can see all the individual dots forming the characters on cheaper dot matrix printers. For the businessman writing reports or letters this is simply not suitable, but for someone printing out program listings, looking at copies of figures, dumping diagrams from the screen, or just preparing rough draft copies, a dot matrix printer is fine.

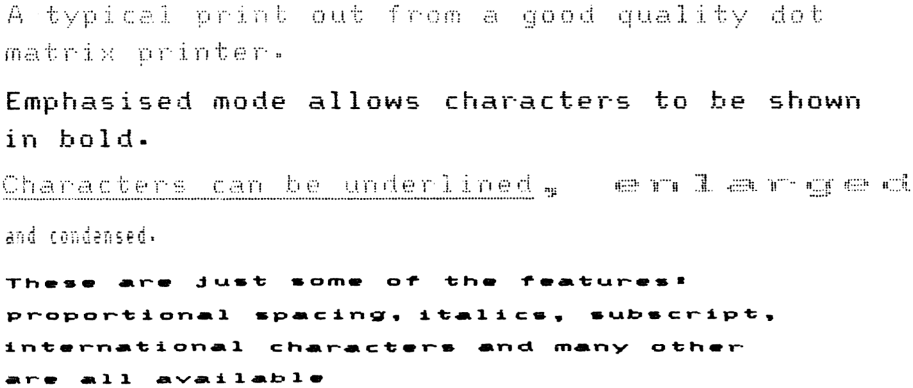
Dot matrix printer manufacturers have been only too aware that they were losing sales whenever people wanted a printer that would print letters to an acceptable standard. Their answer, the NLQ system, was a very positive one. NLQ stands for Near Letter Quality mode and uses a grid system of pins in the same way as the ordinary mode, except that the grid is much finer. Instead of a usual 8 by 8 grid (some are 9 by 7) NLQ mode puts the printer into a mode that uses a grid of more like 20 by 16 proportions. This leads to a much finer, better quality character being printed, but, of course, slows the printing speed right down to between 15 and 35 characters per second. The NLQ mode has really made a dot matrix printer the most suitable choice for the small business which may need to output some material roughly but quickly and also prepare high quality letters and other documents. The NLQ mode on a good printer is hard to differentiate from a daisywheel printer's output.

Daisywheel printers used to have a near monopoly in the word-processing market, but are now threatened by the NLQ dot matrix printers discussed above. A daisywheel printer uses an element shaped like a flower with one character at the end of each 'petal'. The daisywheel does not form its characters using a selection of dots, nor does it draw them. It prints them accurately and neatly in much the same way as a typewriter does. Daisywheels, then, give a high quality output and, furthermore, their typeface can be changed simply by inserting a new daisywheel into the printer. On the negative side, their printing speed is slow, usually under 30 characters per second for a daisywheel printer in the under £500 category — the category that I assume most readers will be thinking of. Daisywheel printers in this price range can have reliability problems, stemming particularly from their less robust manufacture. Nor can they draw diagrams or graphs. It all depends on the amount of work you intend your printer to do, and the model you go for. Again, my own choices are detailed at the end of this chapter, but do make the decision yourself.

The final group encompasses thermal printers, plotters, colour printers and all sorts of bizarre printing devices. Thermal printers have their place at the bottom end of the printer market offering a low cost alternative to the cheap dot matrix printers. Plotters can be found at all prices from a rival to the thermal printers at one end of the market to complex, sophisticated machines of high accuracy at the other. Colour printers tend to start at around £350-£400 and go up and up. Before deciding on any one of these more specialised printers, you must really take a long, hard look at your printing needs. Do you need colour printing? Will you actually use plotting facilities a greal deal? It may be cheaper to buy a good dot matrix printer that has a dot graphics printing mode which, with the right software, will enable high quality graphs and diagrams to be drawn.

Below are examples of the print quality from several different printers.

The following shows a printout from a PCW 8256 printer using its integral word-processor, Locoscript.



A typical print out from a good quality dot matrix printer.

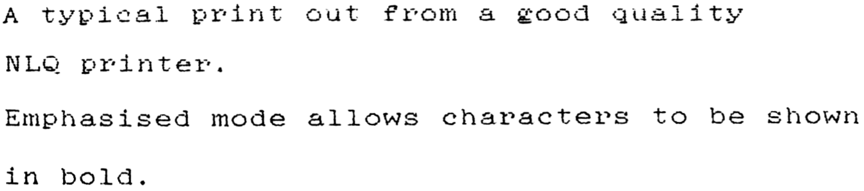
Emphasised mode allows characters to be shown in bold.

Characters can be underlined, enlarged

and condensed.

These are just some of the features: proportional spacing, italics, subscript, international characters and many other are all available

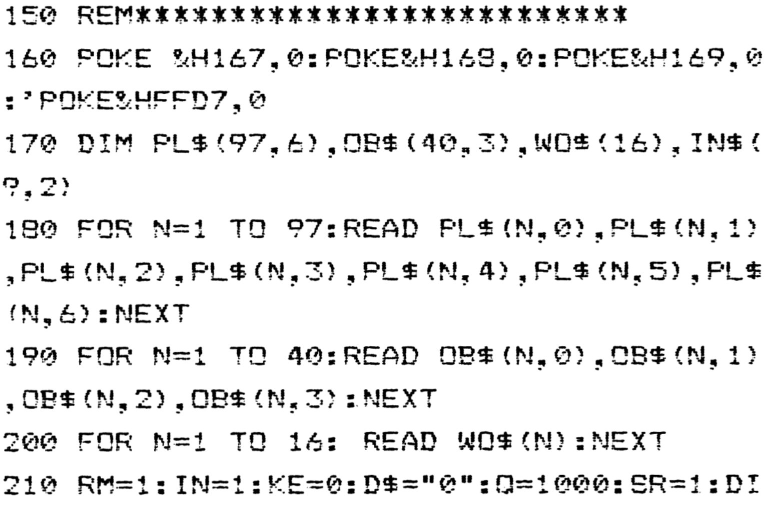
Now, with NLQ mode selected:



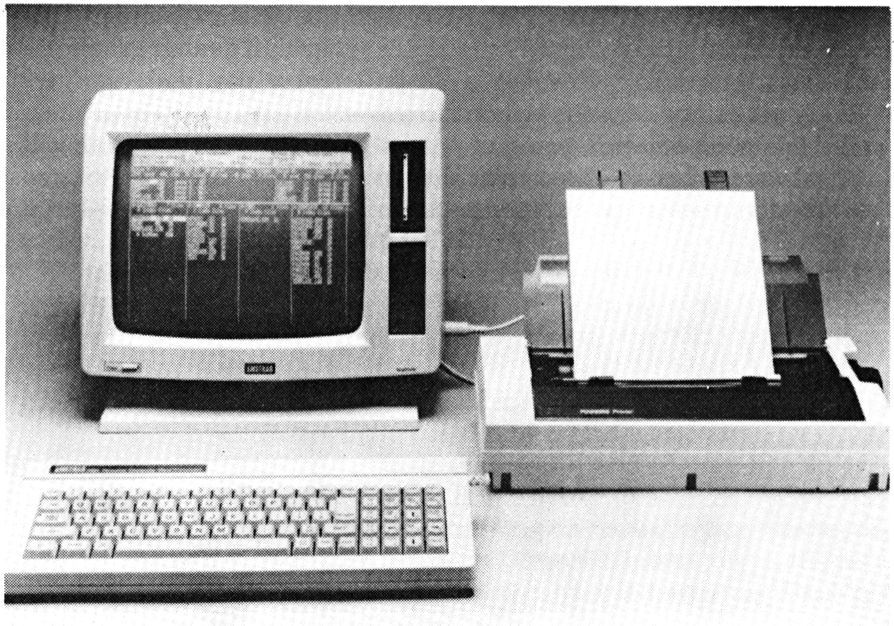
A typical print out from a good quality NLQ printer.

Emphasised mode allows characters to be shown in bold.

To contrast the above, we have a sample of a poorer dot matrix printout.



```
150 REM****************************
160 POKE &H167,0:POKE&H168,0:POKE&H169,0
:'POKE&HFFD7,0
170 DIM PL$(97,6),OB$(40,3),WO$(16),IN$(
9,2)
180 FOR N=1 TO 97:READ PL$(N,0),PL$(N,1)
,PL$(N,2),PL$(N,3),PL$(N,4),PL$(N,5),PL$
(N,6):NEXT
190 FOR N=1 TO 40:READ OB$(N,0),OB$(N,1)
,OB$(N,2),OB$(N,3):NEXT
200 FOR N=1 TO 16: READ WO$(N):NEXT
210 RM=1:IN=1:KE=0:D$="0":Q=1000:SR=1:DI
```



Amstrad PCW 8256 word-processor/personal computer

Readers may come across terms such as 'tractor feed' and 'friction feed'. These, like 'pin feed' and 'cut sheet feed', are all ways of manoeuvring the paper into the printer. It is necessary to make a distinction between them. Friction feed is similar to the style used on typewriters. It is capable of taking continuous or single sheets of paper. Tractor feed and pin feed are similar, both taking fan fed paper which has perforations between each sheet and holes down the sides. Tractor feed is a little more versatile as it can adjust to any width of this type of paper. Single sheet feed is a special, usually optional, device on a printer which feeds in single sheets of paper without manual intervention. This is quite expensive — expect to pay upwards of £200. You will probably need a printer with both friction and tractor feed.

External Storage

As far as storage media are concerned, you must choose between cassette, ROM or disc.

If you have a 664, 6128 or 8256 then your choice, in part, has already been made for you. You have an integral disc drive – my advice is, use it and use it exclusively. The floppy disc is, at present, without peer as a low-cost storage medium. It is fast, can handle reasonably large amounts of data (and access it swiftly) and is fully reusable. The 3-inch drive supplied by Amstrad is a good example of disc technology. It works well and could be easily recommended to 464 owners seeking to upgrade their machine.

Cassette is positively pedestrian in comparison, but as its adherents point out, cassette versions of programs can be a good deal cheaper when you are getting a similar product. The problem here, for disc owners, lies with the 3-inch disc format. These discs cost around £4 each; disc costs (high compared to cassettes — around 35p) are transferred to the consumer and often exaggerated by wholesalers' percentages and dealers' mark-ups. Discs can also be hard to get hold of, which can cause problems.

But these points are easily balanced by the straightforward increase in performance that a disc offers. For a business system based around an Amstrad, I would suggest that one disc is absolutely essential. To start with, a database run from cassette is of little value in time-saving terms and falls short in other areas as well.

Let's take a brief look at ROMs. These are extra 'chips' implanted into your Amstrad either within the actual computer casing or on the outside and connected through the expansion port. These ROMs have a program or series of programs recorded onto them which acts like firmware, making them instantly available when you switch on your machine. ROMs can turn your computer into a word-processor, database, or spreadsheet — in fact, most business packages are available in ROM form.

The disadvantages of a system such as this must be mentioned. There is less flexibility with ROMs. You need special equipment to record your own ROMs; they are really the preserve of commercial software rather like the old Atari games cassettes that were all the rage a few years ago. The programs available are, by and large, very good, but the range is limited. The price of setting up a ROM system with a couple of ROM programs is cheaper than a disc system, but it is also less versatile. I would tend to favour a disc system simply because of its versatility.

Input Devices

Your Amstrad has a keyboard, all Amstrads have that in common and, unlike a number of home computers that I could mention, there is no need to upgrade it. The main input devices that you may consider or come across are: 'mice'; OCRs; microwriters; and grafpads.

Mice are becoming more and more popular with small business computers, but not necessarily with businessmen. They are inputting devices which can be held in the hand or moved around a desk or flat surface, answering questions and making decisions. AMX offer the most popular mouse package for the Amstrad. The idea behind such a device is to take businessmen who are not used to computers, and who don't have time to get accustomed to them and their jargon, away from the complexities of the keyboard to a hand-held device that they can use confidently. This, in my opinion, is a fine philosophy for a computer such as the Apple Macintosh which is simplicity itself to use, but is not so well-suited to the Amstrad. A mouse can only be used in a limited number of situations; when it comes to typing in text then it is almost certainly necessary to revert to the keyboard. The small business or self-employed man will not have a brace of secretaries eager to do the typing; more likely than not, they will have to do it themselves, which means that they will have to get used to the keyboard sooner or later. Don't dismiss the mouse if you have artistic or graphic intentions as it can be of great use here, but otherwise I doubt if it will be of much real interest to you.

OCR stands for Optical Character Reader and is a device that I predict will become more and more popular as time goes on. At the time of writing, there are only a few models available at a price within the small computer user's budget, but it is predicted that prices will fall, reliability will increase and the range will widen quickly. An optical character reader 'reads' text on a printed page and translates that into digital information that can be inputted and handled by the computer. This device, then, can enter 20 pages of an author's manuscript straight into the computer which has an in-built word-processing program. Scanning a page with the OCR takes just a fraction of the time it would normally take to type the text into the computer using a keyboard, and, for situations where much textual information must be entered into a computer and then manipulated, it has great potential. As I write, the cheaper models (costing some £300), still have some recognition problems which cause data to be wrongly inputted into the computer. I hope very much that these peripherals overcome the problems they face at present and that there will soon be lower-cost models available for Amstrad users.

A microwriter is a small device run from a rechargeable battery which has half a dozen keys and a small LCD screen. It has a memory capable of holding about five pages of closely typed text and it is intended to be your portable article-, memo- or letter-writer which, once you reach the office or wherever your Amstrad resides, can be loaded into your machine, manipulated and then printed out. As well as a the microwriter you would need an RS232 interface such as the one supplied by Amsoft for £50. With so few keys, a new style of typing characters is obviously required. In the accompanying manual, you are shown how to use the microwriter very clearly, so that a couple of hours will set you on the road and you will improve every time you use it. It could still become a viable alternative to the QWERTY keyboard that everyone is so familiar with.

The grafpad is the most prominent member of a group of inputting devices called 'drawing tablets'. They are flat boards divided into grids which when touched with the provided 'pen' register a signal which is sent to the computer. Obviously, these devices have artistic applications and can be used in similar ways to mice, but there is also potential for diagram design and output. The grafpad costs around £60.

I have not included information about modems and on-line computing. If you are interested in the possibilities of going on-line with your computer then you could do worse than buy Interface Publications' guide to the subject, entitled *Going On-Line With Your Computer.*

The Selection Process

With most hardware purchases, a standard list of questions can and should be asked before parting with any hard cash. Hunches and impressions may sway you, but don't let them rule you. If you are unsure about any aspect of a purchase, don't make it. Question further and think it over.

A busy businessman is unlikely to be able to devote more than a small amount of his time to a hardware purchase. Nevertheless, make sure that you at least find out some basic points and facts on the items. The list below, plus a couple of your own specific questions, may suffice. This is important. If you make a bad purchase, you haven't just lost the item's purchase price but also the time spent in installing it and possibly, causing still more damage, created problems for your business as a result of its having started to depend on the computer system.

1) Does the item of hardware satisfy your minimum requirements for: construction (robust, if necessary); speed of operation; ease of operation; any other requirements of this type?

2) Will the item of hardware expand and adapt to meet foreseeable changes in your system?

3) Can it be used at the same time with other peripherals and hardware?

4) Can you see the hardware fully demonstrated? (This is vital — if you can't, go elsewhere).

5) Does the price include absolutely everything that you need in order to use it, such as leads, power packs, and special software?

6) How complete and comprehensible is the documentation?

7) Do you actually have confidence in the equipment, what it can and what it cannot do?

8) Do you have confidence in the supplier you're dealing with? (There's a suppliers' checklist in the software appraisal section — see page 25 — and much of what is there applies here to hardware purchases as well.)

9) How many of these items have been sold and used? Can you obtain the name, address and phone number of one client who has actually bought and uses one of these items? (Don't always expect an answer here, even from products that have sold thousands and thousands. The company may refuse to give out names and addresses like this.)

10) What are the guarantee and service conditions attached to this hardware?

Hardware Reviews

The two main areas of interest to the Amstrad owner are the purchasing of a printer and of external memory equipment. Both have already been looked at, but detailed below are a few products that I can particularly recommend as a result of my discussions with others as well as through my own use.

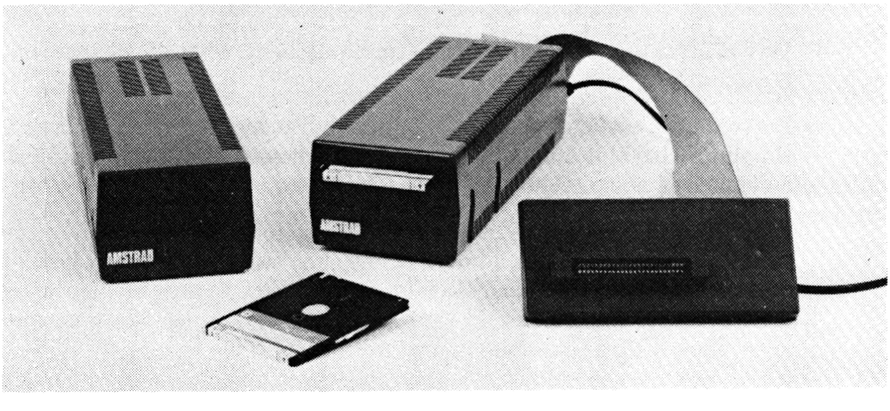
External Memory And Storage

The CPC 464's cassette deck works far more reliably than a lot of home computers' tape interfaces. Yet the speed and ease of discs make a disc drive an essential peripheral for the serious Amstrad user. For 664, 6128 and 8256 users, one disc is built in, but even they should consider the purchase of a second disc for back-up and for running certain software.

The **Amstrad DD1 Disc Drive** currently retails for £159.99 with a disc interface and £99.95 for a second drive without the interface. This drive is very, very reliable, has a reasonable access speed and all in all is an excellent buy. My only complaint is not actually aimed at the drive but rather at the discs themselves. The 3-inch disc format is not the most common of sizes; previously the 5¼-inch and the 3½-inch discs reigned supreme. At the time of writing, there is a fairly severe disc shortage, making them most difficult to get hold of. Even when they are in good supply, the price, a shade under £5 (or closer to £4 if you buy them in tens) is rather steep. Hopefully they will come down in price.

There is another quibble about the disc size. Nearly all CPM software is to be found on 5¼-inch disc which means that Amstrad owners must get specially recorded copies of CPM and other programs on 3-inch disc from suppliers. This leads to a narrowing of the software range and an increase in some prices. A solution may be the addition of a 5¼-inch disc drive.

Cumana do a 5¼-inch disc drive with an internal power supply unit and connecting cables for just £140. This is a very reasonable price for a robustly constructed disc drive which is fully compatible with the Amstrad disc operating system. The drive comes complete with a rough-textured case that won't altogether match the rest of your Amstrad system. But aesthetic considerations aside, the unit is a very good one and is well constructed internally.



Amstrad DDI-1 disc drive and interface

Printers

Those owning a PCW 8256 have nothing to grumble about in this department. They seem to have a solid, high quality device which has an excellent NLQ mode and offers all the features that one has grown used to seeing on Epson-compatible printers. For those looking to buy a dot matrix printer with a good NLQ mode and guaranteed reliability, then I suggest the **Canon PW1080A.** This machine is not the cheapest NLQ printer on the market at around £300, but it is a sturdy one, capable of high speed printing (160 characters per second in draft mode) and with an excellent NLQ mode (27 characters per second in this mode). It takes all types of paper and is a versatile machine. Print quality is high too.

If you don't need an NLQ mode then you are thrown into a tough, competitive tussle between the major printer manufacturers for the best model at the fairest price. The various **Epson** machines are the best quality models of the bunch and have the advantage of being an industry standard, therefore all software with printer options always cater for the Epson. These machines tend to be a little more expensive than the immediate competition.

It may then be worth looking at the **Amstrad DMP 2000,** if you want good quality dot matrix printing. This model does have an NLQ mode though it is arguably not as good as the Canon. It does have a whole host of other features, of which the most impressive is its price, a mere £160.

You may wish to own a printer purely as a means of recording details from the screen onto paper; the quality may not matter. In this case, a thermal printer such as the model supplied by Epson may be your choice. The **Epson P40** is an excellent machine for £100. Its thermal printing style does mean that you will be paying more for special rolls, but the print quality is very high. It uses paper a mere 11.5 cm wide, but can still handle 80 characters across the paper width in condensed mode. What is more, it handles most of the printing modes discussed earlier in the hardware chapter.

If £100 is too much, then why not consider the **Micro-Peripheral CPP-40** which has a selling price of just under £80. It is a print and plotting type of printer using four different colour pens. Its print quality is a little poorer than the Epson but it does have plotting facilities. Whereas you will have to buy thermal paper from a specialist shop for the Epson, you will have to buy pen refills for the Micro-Peripheral model.

At the other end of the printing scale, you may decide that you want a daisywheel printer. The two best models under £500 must be the **Juki 6000** and the **Brother HR15.** Both are high quality units with a range of daisywheels and good customer support. I would favour the Brother over the Juki for its more robust construction, but it is rather a matter of opinion.

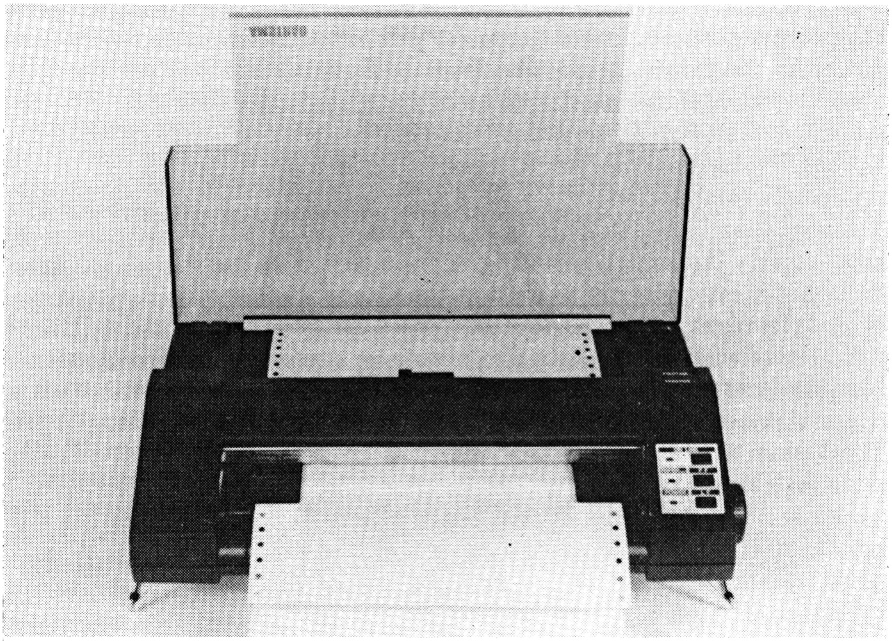
Whatever printer you are thinking of buying, go and see it in operation. Get a sample of the machine's print quality which you can compare with others.

To conclude this chapter, here are some wise words from Alastair de Watteville and Andrew Schulkins, authors of an excellent pamphlet entitled 'An Introduction To The Selection And Use Of Microcomputers'.

1) There is no protection from technological obsolescence. Effort should be devoted to trying to ensure that its effects do not lead to early functional or operational difficulties.

2) The field of selection is so wide, and the pace of change so hectic that a comprehensive survey may not be cost-effective. [This may apply to printers, but not to many other hardware devices that you are likely to use.]

3) An uncomplicated growth path to larger internal memory and to additional or bigger peripheral devices is most desirable. [Your Amstrad does have good expansion possibilities, but a peripheral may not.]



The Amstrad DMP 2000 printer. This high-quality printer takes both tractor and sheet feed paper and has 144 combinations of type style

Software Selection

With your hardware problems hopefully solved, let's now consider the thorny problem of software. Software is probably more crucial to the running of your computer for business than the actual hardware. Without some form of program, the computer will be as idle and as useless as if it were switched off. The range of software available is far greater and more bewildering than hardware, and there are less chances of rectifying errors in your choice of software. Another sobering thought is that selling software second-hand will lose you a good deal of money. You would be lucky if you offered it for more than half its cost price. Clearly, it is necessary to make your software purchases very carefully.

There are three ways of obtaining software to use with your business: you can purchase a ready-made software package; you can commission programmers and designers to write you a specially-tailored package; or you can write your own package. Each approach has its benefits and drawbacks. Writing your own programs is dealt with in a later chapter (see page 33), but let's look at the other two methods in more detail.

Commissioning Packages

Your business may be too specialised to be covered by one of the available packages. Your database or word-processing requirements may be so idiosyncratic that an off-the-shelf word-processor or database would not fulfil your needs. Alternatively, the purpose for which you wish to use a computer may be sufficiently unique to guarantee that no suitable packages have been written. Whatever your reasons for considering this option, have a look at the list of characteristics below which are the preserve of the commissioned program.

1) The specification, delivery date and cost can be negotiated and altered. These details must be carefully decided upon and preferably put into writing before a contract is drawn up.

2) Penalty clauses for late delivery are not common in this field so you may find yourself unsure about the delivery date, just as the program writers may not know when they will be able to complete the program.

3) Some form of constant monitoring should be adopted; regular meetings will ensure that constant checks on progress and problems are made.

4) The programming and documentation standards, although one can try and alter them, are almost certain to be those of the programmer/designer and not those of the customer.

5) A long-term relationship with the programmer/designer will prove beneficial, particularly to the customer. It will allow the package to be maintained and extended, if needs be, by the package authors.

6) No additional skills will be learnt by the customer (as they would be if the customer was writing his own program) but difficult recruitment problems as well as the obvious problems attached to writing the program will be avoided.

7) The price will be higher than buying a ready-made package — in some cases, astronomically higher.

8) One may consider the contracting of part-time programmers to alter an existing commercial program. This situation will have many of the attributes of program commissioning but with less cost and less program flexibility.

Commercial Packages

The range of software available for the Amstrad is extremely varied and covers not only the stock business programs of accountancy, word-processing and database management, but also a large number of more specialised applications and uses.

Before we go on to look at suppliers and package evaluation, take a look at the list below — a comparable one to the list dealing with commissioned packages.

1) The specification may not correspond to the exact requirements. Options may be missing that are considered important by the potential customer. In the same way, there may be many useless options and features that simply waste memory space, take time to learn and add to the cost.

2) Modification of the package may be next to impossible or certainly not at all cost-effective.

3) The delivery period will be either immediate or within a short time, depending on what type of supplier is used and whether a copy of the program required is readily available. There is little uncertainty here.

4) The exact cost will be known though there will be little chance of negotiating figures.

5) The performance of the program package can be tested before purchase. This is a very important point, often underestimated.

6) The documentation can be looked at before purchasing, although little can be done if it's not up to the customer's standard.

7) There may be a useful user group or product newsletter. If there is, it would be a very good idea to join.

8) Subsequent maintenance, improvements and extensions are likely to have to be covered by another agreement and may not be guaranteed.

9) As with commissioned programs, no additional skills will be learnt, but many of the problems will stay outside of the customer's domain.

10) Commercial packages are often cheaper in money terms than commissioned programs and *much* cheaper in terms of time than writing your own programs.

A few points can be made about both commissioned and ready-bought programs. Firstly, it *can* be an advantage to purchase a relatively untried program. The advantages may include a reduced price and extra documentation as well as a greater response to queries than you would normally get. This may amount to semi-collaboration and could be extremely valuable for all concerned.

Secondly, the Amstrad's role as a popular home computer means that there are many home users who have become proficient at programming their machines. Those with more entrepreneurial instincts may advertise their services, charging rates for creating different types of programs, either by the hour, by the amount of memory that the program takes or at a set rate.

Software Package Appraisal — Twenty Questions

This is intended to be a rough reckoner of the capability and usefulness of a given package. To this list you may wish to add further questions. Similarly, you may wish to discard a number of the questions as irrelevant to your particular business or requirements.

1) Does the package match your functional requirements?

2) What limitations are there on the data that the package can accept?

3) What options are available at the time of ordering and what predicted for the future?

4) Will the package be capable of handling your load of data in the required time?

5) When will the package be delivered?

6) Does having an Amstrad computer affect the performance of the package detrimentally? (Here we are dealing with machine-specific packages.)

7) What, if any, training is provided by the vendor? This includes the quality of the documentation as well.

8) What exactly is provided? Extra copies of the user guide or of the disc itself, for example.

9) Has any provision been made for maintenance? Is the guarantee satisfactory? Can you try before you buy?

10) How many copies of the package have been delivered to paying customers. Can you have the address of several of them?

11) Is there a user group or product newsletter?

12) How has the price changed since the initial launch of the package?

13) Has the supplier mentioned the existence of new versions of the package?

14) If the selling company is not based in your own country, then what is known about the stability of both the parent company and the British supplier?

15) What is the seller's general reputation?

16) Is the supplier willing to demonstrate the product to you on his premises?

17) Is the supplier willing to demonstrate the product to you on *your* premises?

18) Will he demonstrate using some of your data as an example?

19) The obvious one: how much more or less does the package cost when supplied by this supplier?

20) If there is any difference, can the supplier give you reasons for the difference in price?

Software Reviews

A small selection of products are reviewed below. They have been chosen for their popularity and quality. They are already well-established, and so my reviews are just to give a little more idea about these packages.

Gemini Database

The database is one of the most popular programs implemented on small business computers, and with the Amstrad this is no exception. There are many databases available offering varying features at equally varying prices, but for me, this is one of the better value databases. Simply named Database, it offers a typical range of features at a reasonable cost: £19.95 for the cassette version and £24.95 for the disc-based variant. What impresses me most is the simplicity with which the newcomer can load in the database and start creating files within minutes. This is undoubtedly aided by the slim manual. Written clearly and very concisely, a few minutes reading tells you all that you need to know about the database and how to use it.

The usual data entry and manipulation modes are offered. I found the entry of some of the longer fields a little slow, but that was just a minor grumble. The ability to format your individual 'cards' of data to present them on-screen and, if required, to printer, I found to be a very useful option.

Gemini's database allows relatively complex searching to be implemented using the various inequality signs and the linking commands AND NOT and OR. A slightly rarer feature on databases is the facility for calculations using the data in the various database fields. This program will give you the total and average values of all the fields in a given data file as well as allowing you to perform more complex calculations on any fields in the file, placing the answer in another field. These calculations can use the usual mathematical operators and also any of the Amstrad's integral functions such as SIN, LOG or ABS. This could be most useful in certain circumstances.

In conclusion, a good value, easy-to-use database, backed by an efficient company prepared to give quality customer support. A buy that you are unlikely to regret.

Tasword

I will keep this review fairly brief as so many people know already about Tasword and its features and also because there is little need to explain what a word-processor is and does.

Tasword can be looked upon as the Model-T Ford of quality word-processing. In its earlier forms on the Spectrum and other computers, it gave serious word-processing capabilities to the masses. Tasword for the Spectrum cost under £10, less than a twentieth of the cost of Wordstar. Tasword for the Amstrad is a little more expensive, but the extra features more than compensate for this.

Tasword-464 is available for the cassette-based 464 Amstrad, while Tasword 464-D runs on the 464 with disc drive or on the 664 machine. Tasword 6128 obviously runs on the 128K Amstrad. There is not a great deal of difference between the three Taswords, simply that as you move up the scale, you are offered a few extra features and more space in RAM for the text. The basic model holds about eight A4 pages while the 464-D version holds closer to 12.

If you have the 464 model but now have a disc drive, it is well worth upgrading to the 464-D, as you are also offered facilities to create mailing lists.

Let's say a few things about Tasword in general. It is an excellent, very easy-to-use word-processor with many of the usual features found on far more expensive word-processors. More than any other program, it is responsible for bringing down the cost of word-processing. Naturally you have full control of page sizes, widths, printer controls, saving, loading and manipulation of files, plus justification, not to mention a line, character and (most useful for the budding journalist or report writer) word count.

The whole program runs around menus and will take a very short time to come to terms with. I found myself writing normally with Tasword far more quickly than with other word-processors. Another feature which is appealing is its transparency. You can get into the program simply by selecting the main menu option 'into BASIC'. From there, you can alter the letter heading, add extra embedded printer control codes or make an extra copy of the program for back-up purposes. Tasman advise you to use the back-up, keeping the master disc safe. This is very good advice.

The program is not without drawbacks. The justification process is slow, and if your material is in paragraphs then to re-justify the whole document one has to re-justify each paragraph, one at a time. If you want to change the size of the margins after you have written the text, or some of it, then you must exercise care. The other major criticism, in my view, is the text file size (8 pages and upwards, depending on which version you have) which makes it necessary to save the file every time it becomes full. Wordstar keeps on saving as you type, making it possible to have a single text file of many times larger than on Tasword.

However, criticisms aside, the program is a superb one and possibly the only word-processor that you need to consider. There are many excellent word-processors now available, but they are generally more expensive and they too have their faults. Examine the features offered by Tasword and if they satisfy your requirements and you don't need to write one long text file but can split it into a number of smaller files, then it must be the word-processor to consider.

Writing to Tasman Products with any queries is a profitable exercise; one tends to get a thorough and well thought out reply from this friendliest of companies.

To conclude: for normal word-processing tasks (report-writing, letters, documents, etc.) Tasword must be strongly considered. I know a number of people, myself included, who regularly use Tasword with the Amstrad.

Tas-Spell

This program, sold as a companion to Tasword, acts as a dictionary that can be manipulated and will check a piece of text written on Tasword for spelling errors. Many people will find themselves in one of two groups as regards a program like this. One group cries 'Marvellous, what a handy program!' while the other tends to murmur 'Clever, but what use is it?' Combining the two, equally valid, views gets something like 'A clever and *potentially* useful device, but what use is it to me in my situation?'

Before considering whether Tas-Spell *is* a useful addition to your program library, it is definitely worth looking at its features and ease of operation. You select the spelling option on the Tasword menu (note that it will only work with improved disc versions of the program) and then load in the Tas-Spell disc. The dictionary comprises some 20,000 words, which sounds a great deal but, in fact, creates a number of limitations — something which we'll look at in a moment.

The program makes about 20 passes over the text checking for words not conforming to its dictionary. You can select options that will skip single letters and numbers without reporting them as spelling errors. When it comes across an error, it will either highlight it on the screen or will send it to the printer, depending on which option you select. If you choose the output-to-printer option, then it also prints the line and column position of the misspelt word.

There are other differences between the printer and screen options. The printer option is ideal for those people who wish to leave the machine to get on with the error checking while they go and perform some other useful task. This is a good option to have, as the program can take some time to check through a piece of text. The screen version needs your interaction with the program. When a misspelt word is located, the program gives you three choices:

Ignore the word
Change the word
Learn the word

The first is pretty straightforward; in this case the computer skips over that word and it is printed as it is. Changing the word is also easily understood. You alter the misspelt word to the correct spelling. The final option is the most interesting one of the three. It allows the program to add the new word to its dictionary and so recognise it in the future. This feature makes the program an interesting and potentially very useful one.

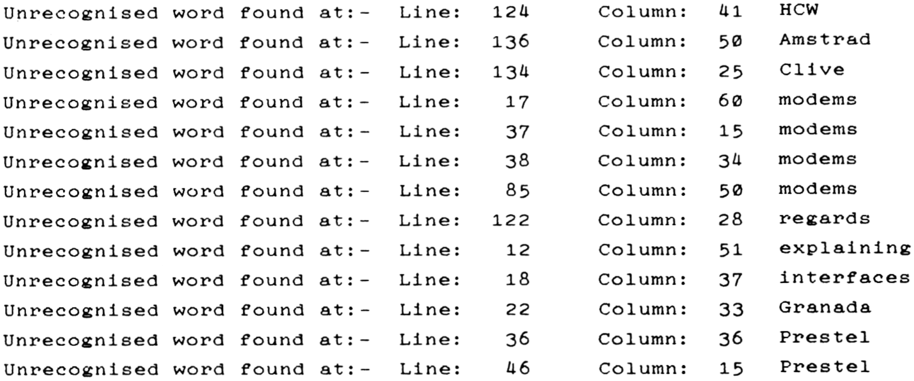
The words already carried do point to a number of quite glaring omissions. 'Explaining', 'regards' and 'incorporating' are all excluded, for example, though they can be easily added to the dictionary. Another point to mention is that there is another command which will delete a word from the dictionary.

Continuing with omissions, you will find that using a program such as this means you will have to make a large number of your own dictionary entries. Apart from customers' names, names of businesses, products and services and technical terms belonging to your area of business, you will have to add any words that utilise hyphens. The secret with this program is to create a text file made up of all words that you consider to be fairly specialised and so not in the dictionary. Running through the text file, all 'misspelt' words can be added as they are to the dictionary. Tasman, the software company behind the program, claim that there is some 50K (approx 10,000 more words) available on the disc. However, they do point out that no segment of the dictionary can grow to more than 10K and this limits the number of extra words. There is still room for more words than you will want to add and there is a provision for creating a new dictionary altogether if you really have many thousands of additional words that you want Tas-Spell to cater for. This might apply when using Tasword and Tas-Spell with a foreign language.

The program is a useful one for people writing large amounts of text but does have a limited application for most businesses. Its low cost, just over £17 at the time of writing, helps to compensate for the slow speed of spelling, checking and correction. When you think that 20,000 words or more are being checked against the text then it is not surprising that this takes time. I ran a couple of benchmarks and found that for a text file of some 140 lines, Tas-Spell took six and a half minutes to check through it.

Conclusion: very much a personal purchase; if you think it will be of good use then consider buying, but be aware of the speed and of the need to add your specialist words, which will take time.

A sample of Tas-Spell's error dumps to printer:



```
Unrecognised word found at:-  Line:  124     Column:  41  HCW
Unrecognised word found at:-  Line:  136     Column:  50  Amstrad
Unrecognised word found at:-  Line:  134     Column:  25  Clive
Unrecognised word found at:-  Line:   17     Column:  60  modems
Unrecognised word found at:-  Line:   37     Column:  15  modems
Unrecognised word found at:-  Line:   38     Column:  34  modems
Unrecognised word found at:-  Line:   85     Column:  50  modems
Unrecognised word found at:-  Line:  122     Column:  28  regards
Unrecognised word found at:-  Line:   12     Column:  51  explaining
Unrecognised word found at:-  Line:   18     Column:  37  interfaces
Unrecognised word found at:-  Line:   22     Column:  33  Granada
Unrecognised word found at:-  Line:   36     Column:  36  Prestel
Unrecognised word found at:-  Line:   46     Column:  15  Prestel
```

Gemini Report Generator

The idea behind this package is that it should act as a complement to the database program (see above), thus allowing you to create textual documents using data from the database files. What the name does *not* tell you is that it acts largely as a mailing list and standard letter formulator.

The document generation option can be used for less mundane things than standard letters. It allows any database information to be merged with the document file and then, with the document complete, it can be manipulated in the usual ways — printing, saving as a file, and so on.

Gemini should be applauded for the way they have made their package range interfaceable. The Report Generator uses the database's files, while data from the database files and also from the cashbook programs can be loaded into and manipulated by their recently introduced Graph Plot package. For the same price as the Database and the Report Generator, the Graph Plot provides the user with a complete statistical analysis package, including correlation coefficients and a large range of diagrammatic forms of presentation that can be dumped to most dot-matrix printers.

Back to the Report Generator, the program is simple to use and comes complete with another excellent, concise user instruction booklet. However, I feel that, at the same price as the original database program, Report Generator is too expensive.

Cashbook Accounting Including VAT File

Gemini's Cashbook package is well known to other computer users, particularly those with a BBC Micro. It has been around for some time now and is widely regarded as a high quality accounting package, complete in every way.

There are three parts to this package, each available separately. Cashbook itself, Final Accounts and VAT File. A sample of the Final Account's printout is given after this review.

To explain only some of the features of this package would take the space of much of this book. Suffice to say that all accounting principles are covered. With VAT file as well, the package covers all bookkeeping needs as well as providing accurate management information in the form of trial balances, profit and loss accounts, trading accounts and so on.

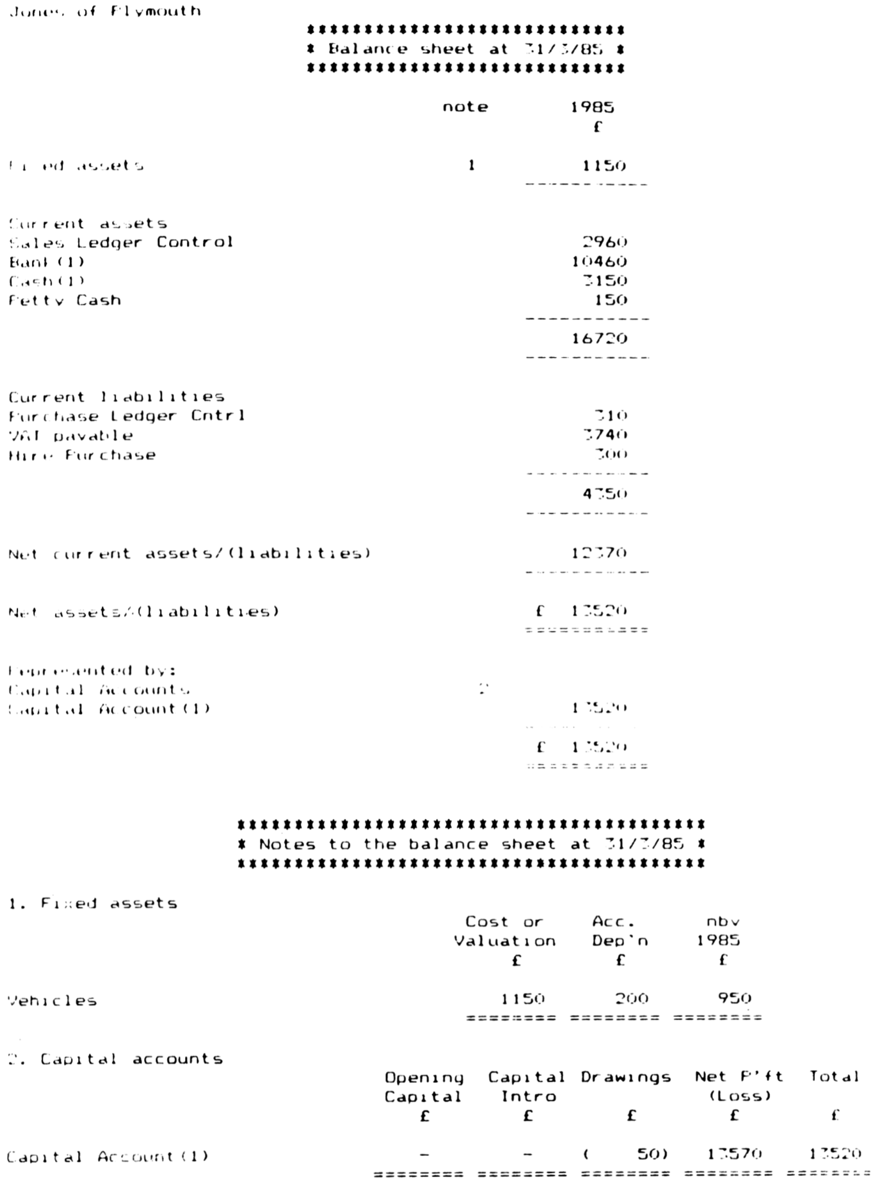
Cashbook costs £60 for the cassette and £65 for the disc version. Its partner, Final Accounts, costs the same. VAT File costs £20 for cassette and £25 for disc versions. However, you can buy all three together in one combination pack for £90 for cassette and £100 for disc. For a fully operable accounting system for the small businessman this is a small price to pay. I consider this a very worthwhile product indeed.

Other Programs Of Note

Campbell Systems' programs are generally of a very high standard and are fairly priced. Their Mastercalc program for the Amstrad, at a similar price to Gemini's Database program, is a marvellous piece of software engineering. It has been widely regarded by many software critics as one of the best spreadsheets available and, as such, is sold at a very reasonable price. Campbell Systems' products are largely sold through Amsoft and it's probably best to deal with Amsoft on most matters.

Camsoft offer a high quality range of financial software. Their Payroll package is the only one that I have actually seen, but it seemed quite impressive. Their documentation is good and all of their range is based around the same standard control program which they call CADO. This means that once you have learnt how to operate one package, subsequent packages become child's play to use.

Amsoft produce the largest range of business software, but some of it is not of the best quality nor of great use. Caxton's range is far more exclusive, only offering a few titles, but they are adaptations of some of the more popular packages available for small business computers. Cardbox is one of their offerings and is a superb file indexing package.



```
Jones of Plymouth
                          ****************************
                          * Balance sheet at 31/3/85 *
                          ****************************

                                      note       1985
                                                   £

Fixed assets                             1        1150
                                             -----------

Current assets
Sales Ledger Control                              2960
Bank(1)                                          10460
Cash(1)                                           3150
Petty Cash                                         150
                                             -----------
                                                 16720
                                             -----------

Current liabilities
Purchase Ledger Cntrl                              310
VAT payable                                       3740
Hire Purchase                                      300
                                             -----------
                                                  4350
                                             -----------

Net current assets/(liabilities)                 12370
                                             -----------

Net assets/(liabilities)                      £  13520
                                             ===========

Represented by:
Capital Accounts                         2
Capital Account(1)                               13520
                                             -----------
                                              £  13520
                                             ===========


                    *****************************************
                    * Notes to the balance sheet at 31/3/85 *
                    *****************************************

1. Fixed assets
                                        Cost or    Acc.     nbv
                                       Valuation   Dep'n    1985
                                            £          £       £

Vehicles                                   1150        200     950
                                        ======== ======== ========

2. Capital accounts
                                 Opening  Capital Drawings  Net P'ft  Total
                                 Capital   Intro            (Loss)
                                    £        £        £        £        £

Capital Account(1)                  -        -     (    50)  13570    13520
                                ======== ======== ======== ======== ========
```

An example of Cashbook Accounting printout

Suppliers

There are three basic ways of purchasing a program: mail order; trade fairs; and over-the-counter suppliers. Mail order is a 'last resort' option. If the product you want is unavailable locally and you need it relatively quickly, then you may have no other choice than to send off for a copy. A couple of points: firstly, beware, all the tricks that can be pulled in other markets apply equally in the computer software market. Secondly, though the price may be lower, you cannot test the product and will have to send it back or write to them if you have problems.

Trade fairs are an in-between option. You can see the product, test it out and it may be on offer at a promotional discount, but you will have to act in the same way as you would with mail order suppliers if you have problems with the product once you get it home. If you are considering using mail order then it may be worth dealing with the original program authors, if they are in a company and are resident in Britain. Tasman are a possible example. I've had some marvellous service from them with Tasword 464-D and Tas-Spell. It may cost you a fraction more (though not always) but it may also gain you better service in future.

Over-the-counter services are probably the best to deal with if you can. You should try to get the product that you're interested in fully demonstrated by them. Even for a program with a selling price of £10, you should be allowed to see how it works — if a shop does not allow you to, then definitely go elsewhere.

Trevor Housely and Joe Auer in Interface's *How To Computerise Your Business Successfully* give some general hints:

1) Make sure you know what is on the market. Almost always, the vendor you have just spoken to will seem the most impressive. Do not lose sight of the fact that there are other, competing suppliers. Be aware of the rival prices.

2) Fight your way through the fog of buzzwords and misleading statements. You may get lulled into a false sense of security because of misunderstandings about the meaning of a buzzword.

3) It does not hurt to have two or three demonstrations of the same system. It is easy to be dazzled by a feature of a system on your first visit. Be prepared to go back with a sober mind and more questions.

4) Watch out for last minute price rises, including the sudden inclusion of a tax or an optional extra.

5) Consider the compatibility of packages.

6) Watch out for unnecessary offerings. When the salesperson withdraws the unnecessary item, the price will come down and you may think that you are getting a good deal. Are you? Probably not, because you didn't need the extra feature in the first place.

7) Read the vendor's proposed contract very, very carefully.

Writing Your Own Programs

For those who have little or no programming knowledge or ability, writing their own programs cannot be seen as a realistic option. If, however, you have some programming skills and wish to improve them, you could, in some cases, try to write your own programs. I do stress ‘in some cases’, as there are programs which are very difficult to write from scratch and which will waste your time. A word-processor is one such program that is not worth bothering with. A good one will take you a considerable time to complete and there are many different models already on the market at prices as low as £6 or £7. The same can be said for a complex spreadsheet program and for a number of other programs.

From this we can derive the main reasons for writing your own programs. Firstly, as we’ve already said, it will undoubtedly improve your programming knowledge and skills. Secondly, it will save you the cost of purchasing commercial software to do the same. Thirdly, when you write your own programs, you will be able to add the enhancements that you want, and cut down on the time and memory-wasting options that you *don’t* want. Fourthly, you may write the program because you have yet to find a suitable alternative offered by commercial software houses. This was the case for a friend of mine who wrote his own program to keep track of articles written, payments received and so on. It wasn’t quite a database because it could also handle financial transactions such as calculating the payments made to him by one publishing group and plotting it as a relationship to another company’s payments or to the hours that he’d worked on each article.

Any one or a number of these benefits may be in your mind when you consider programming. There are many considerations to be taken into account before you definitely decide to go ahead and write your own software solution and, of course, many points to bear in mind *as* you write the program or programs.

I would first take a look at some of the programs in the programs chapter of this book (see page 37). Are they the sort of thing that you had in mind? You may be able to modify one of the programs there; they are all written simply and straightforwardly in BASIC (though the database and word-processor make good use of machine code). You may be able to use one of them as a basis for your own, more elaborate package.

If this is not the case then consider the programs for a moment. Would you be able to write comparable programs? Could you take a need, expressed in common English, and turn it into a programming problem that you can solve using the Amstrad’s BASIC (I’ve assumed BASIC as this is the language that most people know and it is the one supplied with all versions of the Amstrad computer). If you think that you can, read on. If not, it may be an idea to return to one of the two other available options, buying commercial software or hiring a software consultant to do the job.

You’ve now decided that, provisionally at least, you feel that you can write the program. This is an important decision, because you will be making a big commitment in terms of time. It is unlikely that your program will be such a short one that it will only take an hour or two to write.

If you have read through the preceding chapters, particularly the ones dealing with software acquisition and with defining your business problem, then you are part of the way there. You have a list, firstly of functions that the program must perform and secondly of the capabilities and features that you see as being necessary for inclusion within the program.

Look at these lists long and hard. Are there any features that you can drop easily? What features are the most important? By this stage you'll have a far better idea of what product you in fact want to get hold of. Have another look around for an affordable commercial package that may do the job. Remember the points made in the chapter on software choice and don't expect to write an enormous package in a week. This is the last time that you can re-think your software acquisition plan and make changes without losing valuable time and possibly capital.

You are now going ahead to program.

Good Programming

It may seem so obvious to everyone how to program well, but time and time again we can see poor programming leading to program failure, convoluted coding and an inefficient use of the computer's facilities.

If you are intending to write a fair-sized program and wish to improve your own programming skills at the same time, then I suggest the purchase of a book on structured programming. Peter Juliff's *Program Design* is one such title, though there are a number of good buys in this field.

Here are a few points to consider when writing your own programs:

1) Always plan your program on paper before you so much as switch on your Amstrad. You don't have to flow-chart the program, but a setting down on paper of the variables to be used, the procedures to be followed, the sub-routines to be created and so on will help you enormously as you get into the latter stages of writing the program.

2) Divide your program up into separate chunks or modules. Each module can be given a role such as displaying the information or handling the menu. In this way, each module on paper can become a sub-routine or group of sub-routines in the program. Modular programming is the most efficient method of programming and it is easy to see why. Each function is accorded its own sub-routine and one works on that sub-routine until it is complete. It can be tested in isolation from the rest of the program, debugged and modified and then entered into its place in the actual program. The next sub-routine can be taken, written, debugged and tested, then placed in its position in the program, and so on. Sub-routines allow you to work on small portions of the program, and with a proper framework you will always be able to come back to the project a few days or months later and find it easy to start where you left off. Your program structure may look something like this:

```
 5000 REM PRINTER DUMP ROUTINE
 6000 REM MENU DISPLAY
 7000 REM CALCULATE VALUES
 8000 REM ACCEPT USER'S INPUTS
 9000 REM DRAW GRAPH ON-SCREEN
 9500 REM PAUSE ROUTINE
12000 REM INITIALISATION
```

Plenty of room has been left in between each routine title.

3) Your program is bound to contain a number of places where the user will have to input some information, be it date, sales figures or the report's title. Make sure that you prompt the user correctly. Don't leave an input showing simply as a waiting question mark on the screen, add a line of text telling the user what they should be entering and the format in which they should be entering it. For example, don't leave an input as 'ENTER DATE': should they enter it numerically '4/7/86' or with the day, 'Friday 4th July, 1986'? Explain the format, such as: 'ENTER DATE: DAY, NUMBER, MONTH'.

4) Error-checking goes hand in hand with good inputting. If data was inputted correctly every time then there would be no case for having error-checking routines. However, this is not the case and, no matter how hard you try, if you enter large amounts of data, you are bound to get a couple of pieces wrong. Good error-checking should reject all values that are out of range. For instance, if you were inputting details about your clients and one of the questions was age, then an error-checking line in English would read, 'If the client's age is less than 14 or more than 100 then a mistake has been made'. This would translate into BASIC as 'IF AGE <14 OR AGE > 100 THEN GOTO [line dealing with mistaken data]'. In data entry programs such as databases and word-processors where much information is being entered and manipulated, you should always include an option to edit the data.

5) Always set out the display side of the program well. Do not make it confusing or cluttered; spread out the information. Remember, if you are using Locomotive BASIC, that you have two modes that you could use — MODE 1, with 40 characters across the screen, and MODE 2 with 80 characters. Whichever mode you use, make sure that the colours used for the foreground and the background are contrasting, but do not clash as this will have a bad effect on the eyesight after a couple of hours typing away.

6) If you have a printer then make sure that your printer option is carefully worked out to allow for control codes and for the various styles of printing that your program may require. The only way to do this is to test the option again and again. I'm afraid you will use a good deal of printer paper. Scott (who wrote the word-processor program) and I can both vouch for that!

7) Keep your sub-routines in a tidy order and make sure that your programming doesn't become too elongated and untidy. Again, a copy of a book on structured programming is useful here if you haven't programmed a great deal before.

8) Even if the program is going to be used only by yourself, it is worth including a 'help' option whereby, if a certain key is pressed, a summary of how the program works and what commands are available is displayed on-screen.

9) Once you think you have completed your masterpiece, and you have checked and re-checked it, sit someone familiar with your business (but not necessarily with the program) in front of your Amstrad and let them have a go at your program. There is a large chance that they will dig up a few errors or problems. It is very important to test your program fully

before letting it work in your business. It only takes one mistake to erase a lot of hard work, or, if the error is undetected, to wreak havoc with your business.

10) My final point is concerned with documentation. It is essential that you create careful, detailed documentation. You should keep a copy for yourself, and if anybody else is to use it then they too should have a copy.

To summarise:

Plan program out in advance.

Make program listing neat and logical.

Use structured programming techniques, controlling the program through a series of sub-routines.

Make output displays attractive and clear.

Ensure all user prompts are clear.

Add error-checks to input.

Document your program.

Test program thoroughly, and if possible get someone else to try it out.

Programs, Programs, Programs!

Here is a selection of programs that may aid you in your business. Some are demonstrations or examples, while some are genuinely useful programs in themselves. I have kept them all relatively short and fairly easy to follow. This was intended so that you could easily modify and expand any of these programs. Customising them to your own preferences and needs will help you in two ways, by providing you with a useful routine or program, and by improving your programming skills.

Table Format

Here is an example of a way of outputting data in a presentable fashion. This example deals with currency values, call it dollar:pound values. The 'A' value is the number of dollars to each pound (wishful thinking maybe?) while lines 40 and 50 set up the top of the display.

The program displays values between 0 and £9.90 and shows the equivalent dollar values. This could easily be altered to the current values, or to the values of other currencies or even to any two related sets of figures, such as the number of man hours worked to wages paid (allowing for overtime values).

The importance of programs like this is not so much their calculating powers, which in the example program is certainly minimal, but their ability to produce neat, tidy tables. For constantly changing figures and tables of data, a computer is the only logical choice — completing a table manually would take someone up to 40 times longer than it takes by computer.

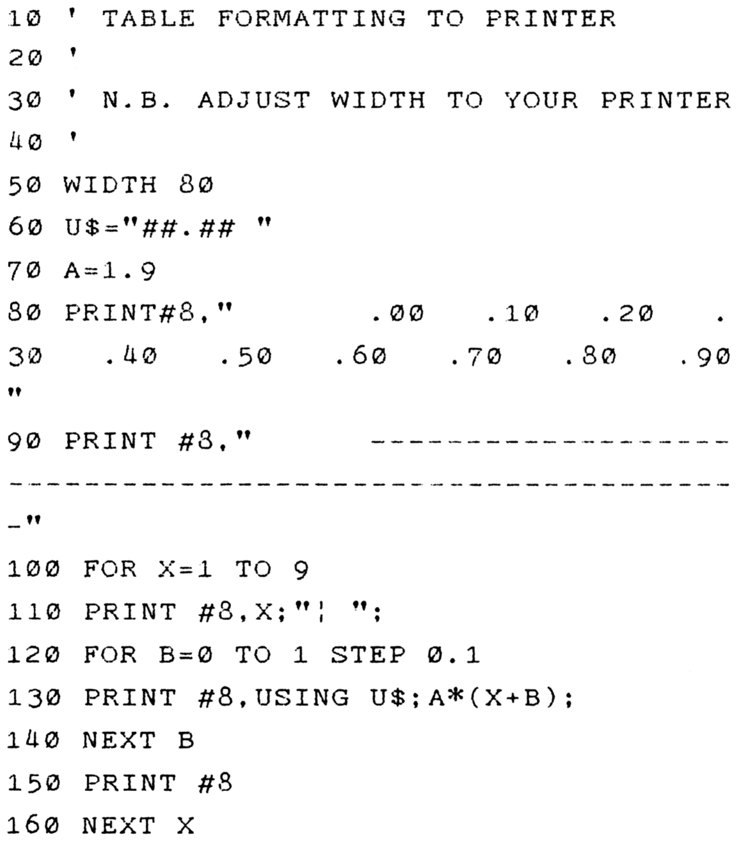
```
5 '        TABLE FORMATTING EXAMPLE
1Ø MODE 2
2Ø U$="##.## "
3Ø A=1.9
4Ø PRINT "       .ØØ    .1Ø    .2Ø    .3Ø
   .4Ø    .5Ø    .6Ø    .7Ø    .8Ø    .9Ø"
5Ø PRINT "      ----------------------
-------------------------------------"
6Ø FOR X=1 TO 9
7Ø PRINT X;"¦ ";
8Ø FOR B=Ø TO 1 STEP Ø.1
9Ø PRINT USING U$;A*(X+B);
1ØØ NEXT B
11Ø PRINT
12Ø NEXT X
```

Table Format To Printer Example

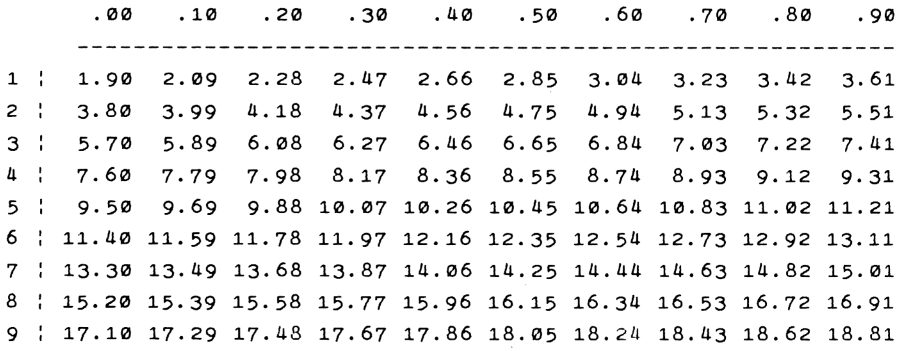
A hard copy of the table displayed on screen in the previous program is essential. With just a few alterations, the program now outputs to printer instead.

The U$ found in both programs is used with the PRINT USING command which formats displays. Full details can be found in the Amstrad Manual.

Here is a sample printout of the table.



```
10 ' TABLE FORMATTING TO PRINTER
20 '
30 ' N.B. ADJUST WIDTH TO YOUR PRINTER
40 '
50 WIDTH 80
60 U$="##.## "
70 A=1.9
80 PRINT#8,"          .00    .10    .20    .
30    .40    .50    .60    .70    .80    .90
"
90 PRINT #8,"          ------------------
--------------------------------------
_"
100 FOR X=1 TO 9
110 PRINT #8,X;"¦  ";
120 FOR B=0 TO 1 STEP 0.1
130 PRINT #8,USING U$;A*(X+B);
140 NEXT B
150 PRINT #8
160 NEXT X
```



```
       .00    .10    .20    .30    .40    .50    .60    .70    .80    .90
       ---------------------------------------------------------------------
1 ¦    1.90   2.09   2.28   2.47   2.66   2.85   3.04   3.23   3.42   3.61
2 ¦    3.80   3.99   4.18   4.37   4.56   4.75   4.94   5.13   5.32   5.51
3 ¦    5.70   5.89   6.08   6.27   6.46   6.65   6.84   7.03   7.22   7.41
4 ¦    7.60   7.79   7.98   8.17   8.36   8.55   8.74   8.93   9.12   9.31
5 ¦    9.50   9.69   9.88  10.07  10.26  10.45  10.64  10.83  11.02  11.21
6 ¦   11.40  11.59  11.78  11.97  12.16  12.35  12.54  12.73  12.92  13.11
7 ¦   13.30  13.49  13.68  13.87  14.06  14.25  14.44  14.63  14.82  15.01
8 ¦   15.20  15.39  15.58  15.77  15.96  16.15  16.34  16.53  16.72  16.91
9 ¦   17.10  17.29  17.48  17.67  17.86  18.05  18.24  18.43  18.62  18.81
```

By just altering the value of A, it is possible to construct a table of different currency values. The example below is supposed to be the exchange rates of the Austrian Schilling versus the pound. The adding of a title above the table improves the output greatly.

```
            AUSTRIAN SCHILLING-STERLING CURRENCY EXCHANGE
            =============================================
       .ØØ    .1Ø    .2Ø    .3Ø    .4Ø    .5Ø    .6Ø    .7Ø    .8Ø    .9Ø
     ---------------------------------------------------------------------
1 |  25.6Ø  28.16  3Ø.72  33.28  35.84  38.4Ø  4Ø.96  43.52  46.Ø8  48.64
2 |  51.2Ø  53.76  56.32  58.88  61.44  64.ØØ  66.56  69.12  71.68  74.24
3 |  76.8Ø  79.36  81.92  84.48  87.Ø4  89.6Ø  92.16  94.72  97.28  99.84
```

Table Creator

The logical extension of the earlier table-formatting programs is a more versatile and flexible version capable of altering its content and style of presentation.

The program is very straightforward to follow: all inputs required from the user are prompted and the essential ones are error-checked. The best way to create a table from this program is to plan out briefly on paper how you want it to look, set it up on the screen, and then, when you are happy with the result, send it out to the printer.

The printing formats help to show the power and use of the PRINT USING command.

```
1Ø '                TABLE CREATOR
2Ø '
3Ø ' N.B. Remember to set up printer t
o      your requirements and to plan o
ut        table size and formatting
4Ø '
5Ø '
6Ø GOSUB 6ØØ
7Ø CLS:PRINT "                TABLE CREAT
OR"
8Ø PRINT "                =============" 
9Ø INPUT "which mode to display in, 1
or 2";M
1ØØ MODE M
11Ø PRINT "enter top column display"
12Ø INPUT C$
13Ø IF LEN(C$)>8Ø THEN LOCATE 1,25:PRI
NT "TOO LARGE":SOUND 1,5ØØ:GOTO 11Ø
```

```
140 INPUT "Enter units across (e.g. 10
 or .1)";Z
150 INPUT "Enter starting value:";Z1
160 INPUT "Enter starting value of uni
ts down:";Y1
170 INPUT "Enter how many units down:"
;Y2
180 INPUT "Enter size of units down (e
.g. 10 or 1)";Y
190 INPUT "Enter relationship between
across and   down using format:  acros
s=X times down";K
200 INPUT "Select Pre-defined format o
r define     your own (Pre-defined = 1
):";H
210 IF H<>1 THEN 410
220 PRINT:PRINT
230 PRINT "      Formats Available"
240 PRINT:PRINT
250 PRINT " 6 numbers       press 1"
260 PRINT " 5 numbers       press 2"
270 PRINT " 4 numbers       press 3"
280 PRINT " 3 numbers       press 4"
290 PRINT " 4.4             press 5"
300 PRINT " 2.2 space       press 6"
310 PRINT " 3.2             press 7"
320 PRINT " 4.2 space       press 8"
330 PRINT " 2.4 space       press 9"
340 SOUND 1,200
350 WHILE VAL(INKEY$)>9 AND VAL(INKEY$
)<1:WEND:L$=INKEY$
360 L=VAL(L$):U$=M$(L)
370 IF L=0 THEN 350
380 SOUND 1,160,15,15
390 GOTO 440
400 PRINT:PRINT
410 INPUT "Enter format procedure (PRI
NT USING). If you're unsure as to the
formatting      enter XXX and the compu
ter will default to a 3 number-decimal
 point-2 number-    space format";U$
420 IF U$="XXX" THEN U$="###.## "
430 IF LEFT$(U$,1)<>"#" AND LEFT$(U$,1
)<>"+" AND LEFT$(U$,1)<>"$" THEN LOCAT
E 1,24:PRINT ">>>>>>>INVALID FORMAT -
TRY AGAIN<<<<<<<";:SOUND 1,200:SOUND 1
,300:WHILE INKEY$="":WEND:LOCATE 1,24:
PRINT SPC(41):LOCATE 1,20:GOTO 410
440 INPUT "Do you want output to print
er or screen? Printer=1, Screen=0";J
450 IF J=1 THEN J=8
460 LINE INPUT "Enter title of table
                  :";T$
470 CLS
480 IF J=8 THEN LOCATE 12,12:PRINT "PR
INTING NOW..."
490 PRINT #J,T$
500 PRINT #J,"-------------------------
---------------------------------------
------------------":PRINT #J
510 PRINT C$
520 FOR X=Y1 TO Y2*Y STEP Y
530 PRINT #J,Y;"¦ ";
540 FOR B=Z1 TO Z*10 STEP Z
550 PRINT #J,USING U$;K*(X+B);
560 NEXT B
570 PRINT #J
580 NEXT X
590 IF INKEY$="R" OR INKEY$="r" THEN 1
90 ELSE 590
600 RESTORE
610 DIM M$(9)
620 FOR T=1 TO 9:READ M$(T):NEXT T
630 RETURN
```

```
640 DATA "######","#####","####","###"
,"####.####","##.## ","###.##","####.#
# ","##.#### "
```

Company Growth Rates

This simply calculates the company growth rate from the data given. Like the previous program, it is quite possible to calculate the answer on a scientific calculator, but it is much simpler to use a computer program which handles and error-checks the inputted data and which can output the answer in whatever format you choose.

A number of these smaller calculating programs could be easily merged together with others to complete a useful maths and stats package for the businessman.

```
10 '      COMPANY GROWTH RATES
20 A=0
30 CLS:PRINT
40 INPUT "ENTER NUMBER OF DATA POINTS:
";C
50 PRINT:PRINT "ENTER DATA POINTS IN O
RDER"
60 PRINT:PRINT
70 FOR B=1 TO C
80 INPUT Z(B)
90 NEXT B
100 FOR B=1 TO C
110 IF Z(B)<=0 THEN PRINT "Data less t
han zero":GOTO 20
120 A=A+LOG(Z(B))
130 NEXT B
140 A=1/C*A
150 D=0:E=0
160 FOR B=1 TO C
170 D=D+(LOG(Z(B))-A)*(B-(C/2))
180 E=E+((B-(C/2))^2)
190 NEXT B
200 S=D/E
210 PRINT:PRINT "GROWTH RATE IS";INT((
100*S)+0.5);"%"
```

Eye-Catching Display

The title says it all. Here the computer is used as a form of advertising device, either in the shop window, in a display in the shop itself or at a trade fair, conference or exhibition. This program is not intended to be used exactly as it is by a business or an individual, but it does show what can be done with relatively little programming knowledge and ability. The original display hopefully catches enough attention to ensure an audience for the all-important second half which details the products, service, news or whatever information is to be displayed to others. The advantages of such a display are many. Firstly, it is easily altered and updated, unlike posters and leaflets. Secondly, there is the attraction of moving pictures and sounds which may attract where static advertising methods do not. Thirdly, there is always the novelty factor, that of attracting people simply because it is something new which will hold attention longer than a more conventional display technique.

If you do intend to use the program then you will have to alter lines 260 and 270 as well as 360 to 390 in order to enter your own messages. Making your display last less or more time is simply a matter of adding more or less information to be displayed. Another option could be to offer the customer hard copies of the information on the screen that the computer will print.

```
10 '           EYE-CATCHING DISPLAY
20 '
30 MODE 0
40 BORDER 0
50 CLG
60 CLS
70 INK 0,0
80 INK 2,24,6
90 FOR T=1 TO 180
100 PLOT 320,200
110 DRAW RND*640,RND*400,RND*14
120 NEXT T
130 PEN 1
140 FOR Z=350 TO 40 STEP-1
150 SOUND 1,Z,1,15
160 SOUND 2,Z-2,1,15
170 SOUND 4,Z+2,1,15
180 NEXT Z
190 PEN 15
200 FOR K=1 TO 30
210 FOR T=1 TO 14:INK T,1:NEXT
220 INK 15,26
```

```
230 ' NAME OF COMPANY IN NEXT LINE.
240 ' REMEMBER TO CENTRE NAME.
250 '
260 LOCATE 5,11:PRINT "MIKROLEISURE"
270 LOCATE 7,13:PRINT "presents"
280 FOR T=1 TO 15:INK T,T:NEXT
290 NEXT K
300 '
310 ' START OF ACTUAL MESSAGE DISPLAY
320 '
330 CLS:MODE 1:INK 1,24:INK 2,26,0:INK
 3,2
340 PEN 3
350 PRINT "                  STYLES OF DISP
LAY                        ============
====="
360 PRINT:PRINT "  One could opt for a
 textual display   after the attention
-getting start. It   depends on what i
s being advertised and who to.";
370 PEN 1:PRINT "  Highlights ";:PEN 3
:PRINT "could be added to":PRINT "impr
ove presentation. The Amstrad is a  ve
rsatile machine in this respect."
380 PEN 3:PRINT:PRINT "  One can ";:PE
N 2:PRINT "flash ";:PEN 3:PRINT "words
 or sentances."
390 PRINT "Setting up a display is jus
t a matter ofgetting to grips with the
 ";:PEN 1:PRINT "PEN INK PAPER":PRINT"
CLS LOCATE PRINT";:PEN 3:PRINT " and "
;:PEN 1:PRINT "SOUND ";:PEN 3:PRINT "c
ommands."
400 DIM M(16):DIM D(16)
410 FOR H=1 TO 2
420 FOR T=1 TO 16
430 READ M(T):READ D(T)
```

```
44Ø SOUND 1,M(T),D(T),15
45Ø NEXT T
46Ø SOUND 1,Ø,6Ø
47Ø RESTORE
48Ø NEXT H
49Ø DATA 478,5Ø,319,5Ø,358,13,379,13,4
26,13,239,6Ø,319,6Ø,358,13,379,13,426,
13
5ØØ DATA 239,6Ø,319,6Ø,358,15,379,15,3
58,15,426,7Ø
51Ø WHILE INKEY$="":WEND:RUN
```

Straight Line Depreciation

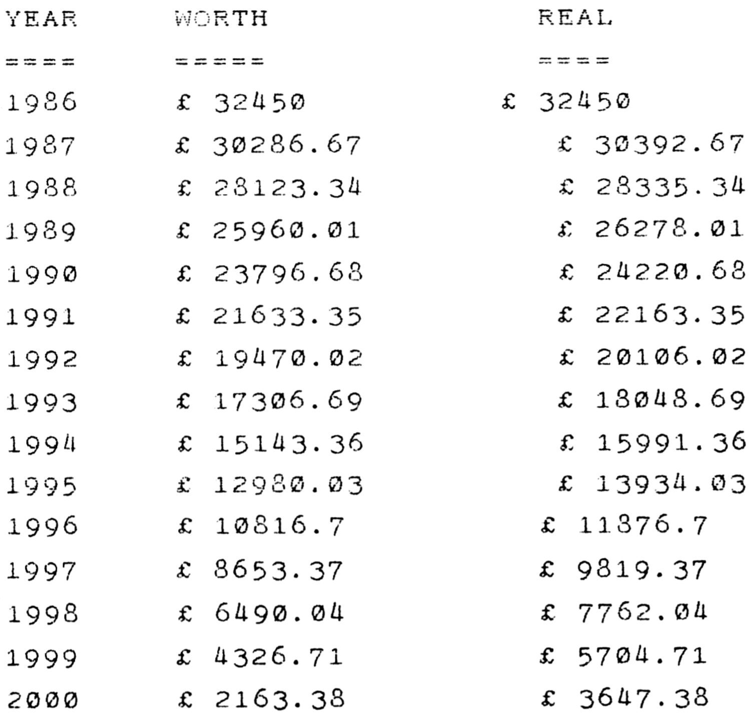
As most businessmen, or at least their accountants, know, depreciation is where the purchase price of an item used in business is gradually written off for tax purposes. This is a gross simplification of a more complex issue in which there are a number of different means of allowing for depreciation. One such type is 'straight line depreciation'.

This program calculates the straight line depreciation of an item, given its purchase price and useful life in years. (It would, however, be very easy to modify the program to work in months rather than years.) It also looks at situations where at the end of its useful life the item is still worth a sum. The program outputs the details, year, item's value under straight line depreciation, or item's value under straight line depreciation but allowing for a fixed value at the end of its useful life, either to the screen or to the printer.

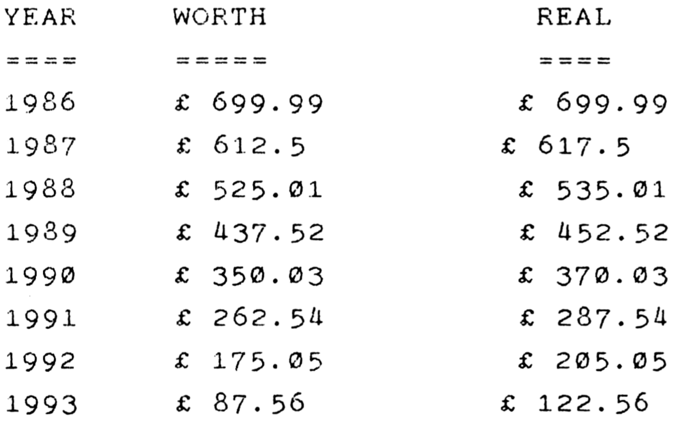
Following the program are two examples, one large and one somewhat shorter.

```
1Ø '  Straight Line Depreciation
2Ø '
3Ø Y=Ø:CLS
4Ø INPUT "Purchase Price:";P
5Ø PP=P
6Ø PRINT:PRINT
7Ø INPUT "Life Of Asset:";L
8Ø PRINT:PRINT
9Ø INPUT "Value at end of asset life:"
;EV
1ØØ D=(INT(P*1ØØ/L))/1ØØ
11Ø R=(INT((P-EV)*1ØØ/L))/1ØØ
12Ø PRINT:PRINT "It depreciates";D;"po
unds a year"
```

```
130 PRINT:PRINT
140 INPUT "First Year Of Use (1986 etc
):";Y
150 PRINT:PRINT
160 INPUT "Output to printer instead o
f screen Y/N?:";P$
170 P$=UPPER$(P$):IF P$="Y" THEN 300
180 CLS
190 X=3
200 PRINT " YEAR     WORTH           R
EAL"
210 PRINT " ====     =====           =
==="
220 PRINT Y:"    £";P;
230 LOCATE 27,X:PRINT "£";PP
240 P=P-D
250 PP=PP-R
260 IF P<1 THEN END
270 Y=Y+1
280 X=X+1:IF X=26 THEN X=25
290 GOTO 220



300 '           PRINTER
310 '
320 PRINT #8," YEAR     WORTH
     REAL"
330 PRINT #8," ====     =====
     ===="
340 PRINT #8,Y:"    #";P;
350 PRINT #8,"         #";PP
360 P=P-D
370 PP=PP-R
380 IF P<1 THEN END
390 Y=Y+1
400 GOTO 340
```



```
YEAR      WORTH               REAL
====      =====               ====
1986      £ 32450           £ 32450
1987      £ 30286.67           £ 30392.67
1988      £ 28123.34           £ 28335.34
1989      £ 25960.01           £ 26278.01
1990      £ 23796.68           £ 24220.68
1991      £ 21633.35           £ 22163.35
1992      £ 19470.02           £ 20106.02
1993      £ 17306.69           £ 18048.69
1994      £ 15143.36           £ 15991.36
1995      £ 12980.03           £ 13934.03
1996      £ 10816.7          £ 11376.7
1997      £ 8653.37          £ 9819.37
1998      £ 6490.04          £ 7762.04
1999      £ 4326.71          £ 5704.71
2000      £ 2163.38          £ 3647.38
```



```
YEAR      WORTH               REAL
====      =====               ====
1986      £ 699.99           £ 699.99
1987      £ 612.5           £ 617.5
1988      £ 525.01           £ 535.01
1989      £ 437.52           £ 452.52
1990      £ 350.03           £ 370.03
1991      £ 262.54           £ 287.54
1992      £ 175.05           £ 205.05
1993      £ 87.56           £ 122.56
```

Personalised Letters

Many businesses find that they have to write a lot of letters; often they are similar letters to a number of customers, potential customers, suppliers or different press sources. A lot of work can be saved using databases, files and mailing lists to compile one standard letter which is printed for each individual on the mailing list, together with a label containing the address of the individual.

This program performs a similar function but can be understood within minutes, and used by anybody. This program is less complex than combined mailing list/database packages, but then it is just a fraction of the size of such a program and performs a single defined function, producing straightforward personalised letters, with the sender's name at the top of the letter, the receiver's name at the start of the letter and an address label printed underneath the letter.

To keep it simple, I have written the program to work on a 40 character screen and hence a 40 character width printer. Converting it to run on a wider printer and screen width is a simple matter, but is of course left up to the individual users with their own particular printers.

How does the program work? Well, on running the program you will see a menu on the screen with five choices. If you haven't already created a file of names and addresses or if you wish to create a new file then select option four. With the file completed, it can be checked using option five. If you wish to edit one of the names/addresses then press the spacebar when it appears on the screen. If you wish to alter it then just press any other key. If you already have a file, then load it in using option number three. This works for both tape and disc.

With the names and addresses stored in the Amstrad's memory, you must now write the standard letter. There are five text blocks that can be used, the lines of characters that you write on the screen form the display that will be outputted to the printer along with your company's business address, date, reference number and so on. Mentioning the business address, this is held in line 580. The present one is fictitious — it's a simple matter to replace it with your own, or to delete it if you're using headed sheets or tractor paper. The gaps between each line of the address are for the print formatting.

Once the letter is written, you can print the letters by selecting the top option. The question about indented addresses is for the address labels. A good hint is to set up a name and address file with two items in it so that you don't waste too much time or paper while getting used to the program.

```
10 '   *****************************
20 '   *                           *
30 '   *    PERSONALISED LETTERS   *
40 '   *                           *
50 '   *****************************
60 '
70 '
80 GOSUB 510
90 GOTO 600
100 '            WRITE LETTER
110 '
120 CLS
130 LOCATE 4,4
140 INPUT "Letter Reference:";N$
150 LOCATE 4,8
160 INPUT "Date:";E$
170 LOCATE 1,12
180 PRINT "Note that pressing ENTER fi
```

```
nishes a textblock. Any blank lines le
ft in a text   block will not be print
ed. To effect theprinting of blank lin
es, lots of spaces must be used."
190 WHILE INKEY$="":WEND
200 GOSUB 470
210 LOCATE 1,1:LINE INPUT A$
220 LOCATE 1,7:LINE INPUT B$
230 LOCATE 1,13:LINE INPUT C$
240 LOCATE 1,19:LINE INPUT D$
250 GOTO 600
260 '           PRINT LETTERS
270 '
280 CLS:LOCATE 4,4:PRINT "Do you want
addresses indented (Y/N)?"
290 LP$=INKEY$:IF LP$="" THEN 290
300 IF LP$<>"Y" AND LP$<>"y" THEN ID=0
 ELSE ID=1
310 LOCATE 10,12:PRINT "Printing Lette
rs..."
320 FOR J=1 TO LIMIT
330 PRINT #8,F$
340 PRINT #8:PRINT #8,"
        ";E$
350 PRINT #8,"Ref:";N$:PRINT #8
360 PRINT #8,"Dear ";:PRINT #8,X$(J)
370 PRINT #8,A$:PRINT #8,B$:PRINT #8,C
$:PRINT #8,D$
380 PRINT #8:PRINT #8:PRINT #8:PRINT #
8,"-----------------------------------
-----"
390 IF ID=1 THEN 420
400 PRINT #8:PRINT #8,X$(J)
410 FOR Z=1 TO 5:PRINT #8,Y$(J,Z):NEXT
 Z:GOTO 440
420 PRINT #8:PRINT #8,"          ";X$(
J)
```

```
430 FOR Z=1 TO 5:PRINT #8,"            "
;Y$(J,Z):NEXT Z
440 PRINT #8:PRINT #8,"-----------------
--------------------------"
450 NEXT J
460 SOUND 1,100:GOTO 600
470 '
480 CLS
490 FOR T=6 TO 25 STEP 6:LOCATE 1,T:PR
INT "_":NEXT
500 RETURN
510 ' INITIALISE
520 BORDER 13:INK 1,0:INK 0,13:INK 2,1
530 PEN 1
540 LIMIT=1
550 KJ=810
560 WIDTH 40
570 DIM X$(800):DIM Y$(800,5)
580 :::::::::::::::::::::::::::::::::F$=
"                       Abacus Consultan
ts,                        27 Jameson St
reet,                         Wetherby,
                                 Yorkshi
re.                                 YK5
4TB."
590 RETURN
600 '              MENU
610 '
620 CLS:SOUND 1,200,20,15
630 PEN 2
640 LOCATE 16,5:PRINT "M E N U"
650 LOCATE 16,6:PRINT "======="
660 PEN 1
670 LOCATE 10,10:PRINT "1      Print Let
ters"
680 LOCATE 10,12:PRINT "2      Write Let
ter"
```

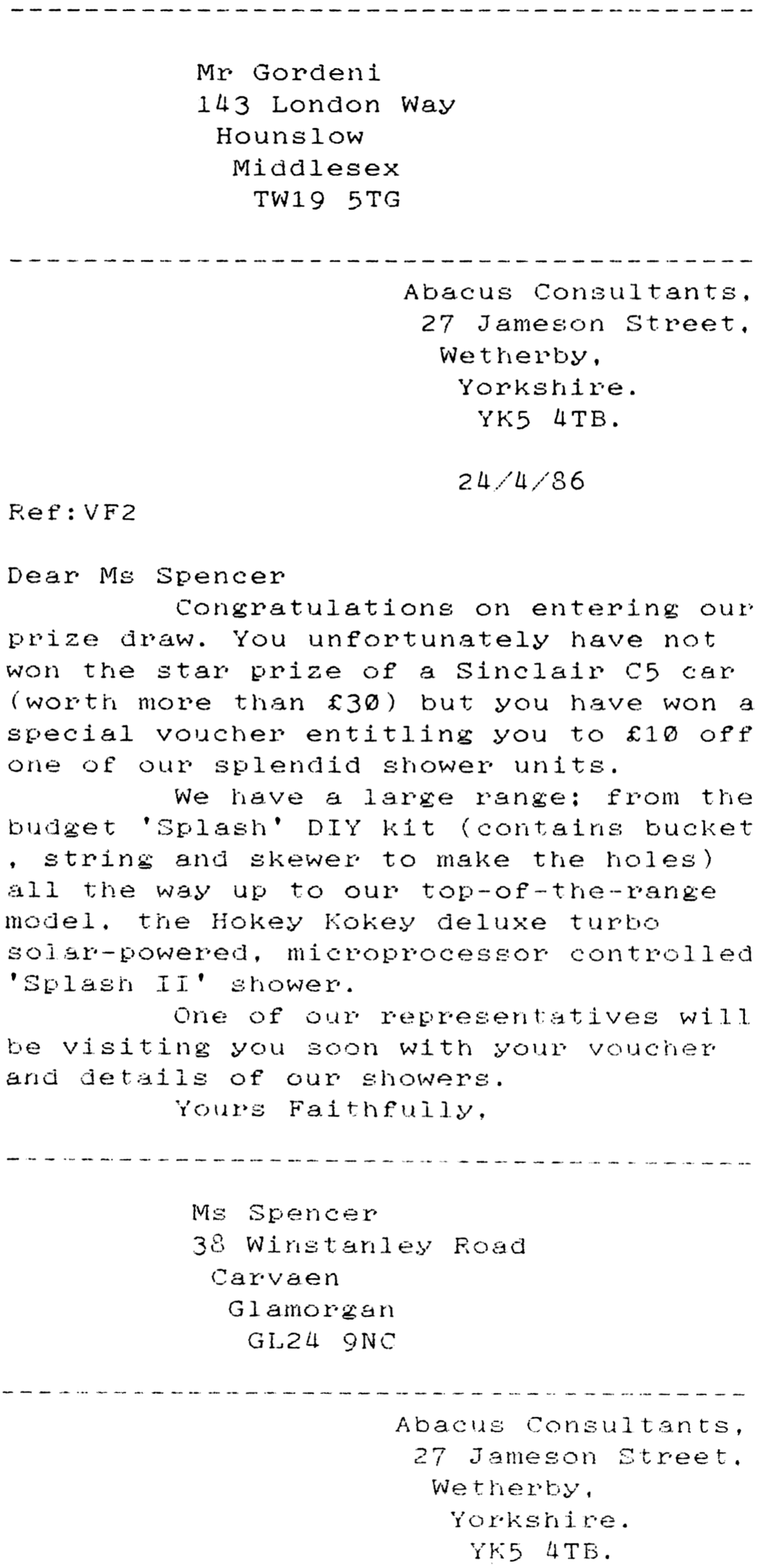
```
690 LOCATE 10,14:PRINT "3    Load Name
/Address file"
700 LOCATE 10,16:PRINT "4    Create Na
me/Address File"
710 LOCATE 10,18:PRINT "5    Edit Name
/Address file"
720 LOCATE 12,24:INK 3,6,21:PEN 3:PRIN
T "Select an Option":PEN 1
730 L$=INKEY$:IF L$="1" THEN 260
740 IF L$="2" THEN 100
750 IF L$="3" THEN 800
760 IF L$="4" THEN 960
770 IF L$="5" THEN 1240
780 GOTO 730
790 GOTO 790
800 '           Load Name/Address file
810 '
820 CLS:LOCATE 4,8
830 INPUT "Name of File:";FL$
840 PRINT:PRINT:INPUT "No. of clients
(if unknown put a greaterfigure than t
here definitely are) ";LIMIT
850 IF LIMIT<1 AND LIMIT>800 THEN 840
860 PRINT:PRINT
870 OPENIN FL$
880 FOR T=1 TO LIMIT
890 INPUT #9,X$(T):FOR Z=1 TO 5:INPUT
#9,Y$(T,Z):NEXT Z
900 IF EOF=-1 THEN 920
910 NEXT T
920 CLOSEIN
930 FOR T=1 TO LIMIT:IF X$(T)="" THEN
KJ=T-1:T=LIMIT
940 NEXT T:IF KJ<LIMIT THEN LIMIT=KJ
950 GOTO 600
960 '     Create Name/Address file
970 '
```

```
980 CLS:LOCATE 6,4:PRINT "Creating A N
ame/Address File."
990 LOCATE 6,5:PRINT "================
=============="
1000 LOCATE 4,8:PRINT "    Remember to
 press ENTER after each  address line.
 The computer handles five lines so if
 your address is shorter,  then just
 press ENTER for the remaining lines."
1010 PRINT:PRINT:INPUT "Number of data
 statements";LIMIT
1020 IF LIMIT<1 OR LIMIT>800 THEN GOTO
 1010
1030 CLS
1040 FOR J=1 TO LIMIT
1050 LOCATE 4,4
1060 INPUT "NAME:";X$(J)
1070 LOCATE 4,8:PRINT "Address?"
1080 FOR Z=1 TO 5:LINE INPUT Y$(J,Z):N
EXT Z
1090 CLS:NEXT J
1100 LOCATE 10,10:PRINT "Save File? (Y
/N)"
1110 LP$=INKEY$
1120 IF LP$="Y" OR LP$="y" THEN 1150
1130 IF LP$<>"" THEN 600
1140 GOTO 1110
1150 ' saving process
1160 PRINT:PRINT:INPUT "Name of file (
less than 8 characters) ";FS$:IF LEN(F
S$)<1 OR LEN(FS$)>8 THEN 1160
1170 PRINT:PRINT
1180 OPENOUT FS$
1190 FOR T=1 TO LIMIT
1200 PRINT #9,X$(T):FOR Z=1 TO 5:PRINT
 #9,Y$(T,Z):NEXT Z
1210 NEXT
```

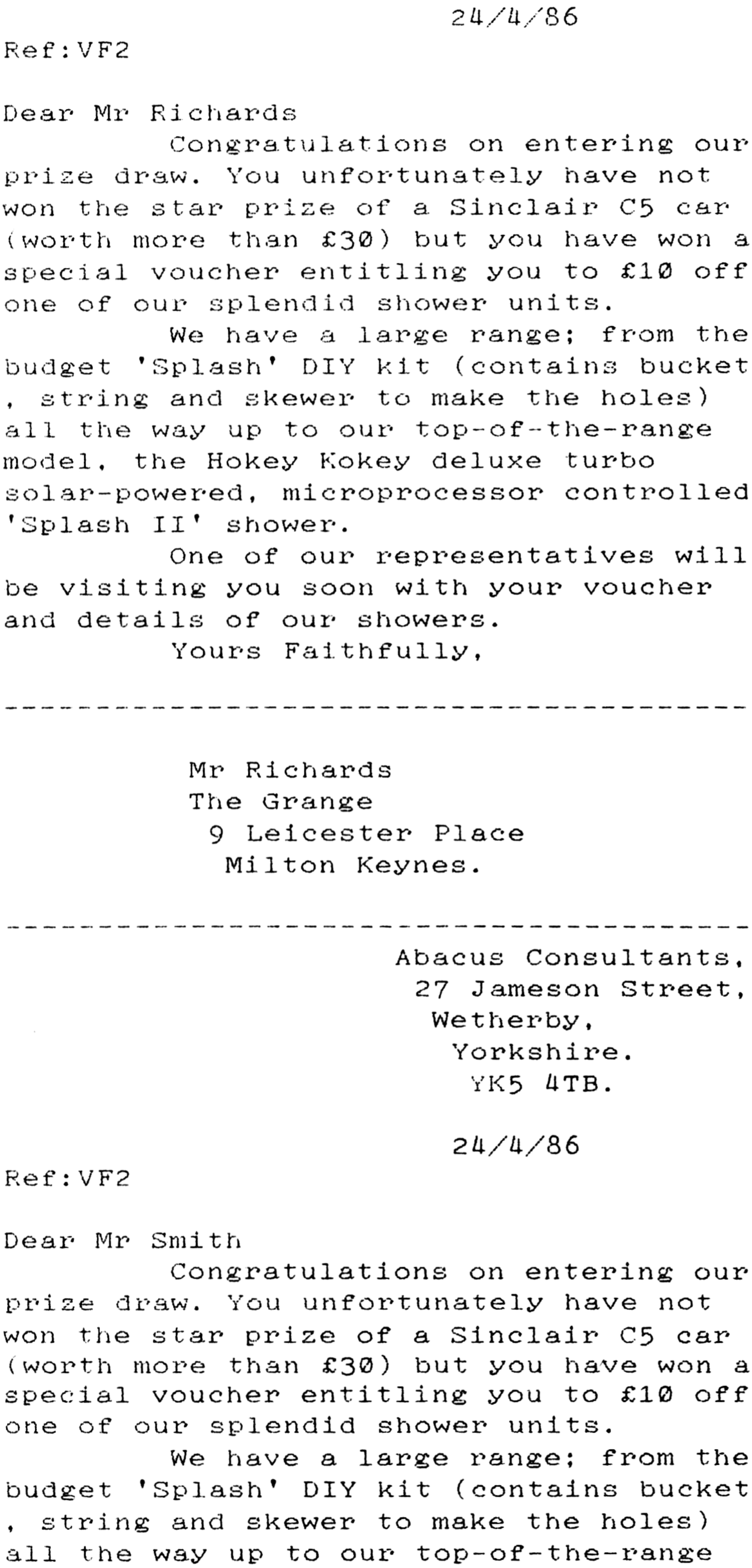
```
1220 CLOSEOUT
1230 GOTO 600
1240 '      Edit Name/Address File
1250 '
1260 IF X$(1)="" AND X$(2)="" AND X$(3
)="" THEN PEN 3:LOCATE 8,24:PRINT "  N
O FILE IN TO EDIT  ":WHILE INKEY$="":
WEND:GOTO 610
1270 CLS:LOCATE 1,25
1280 FOR T=1 TO LIMIT
1290 PRINT "============================
==============";
1300 PRINT X$(T)
1310 FOR Z=1 TO 5:PRINT Y$(T,Z):NEXT Z
1320 LP$=INKEY$:IF LP$="" THEN 1320
1330 IF LP$=" " THEN 1380
1340 PRINT:PRINT:INPUT "NAME:";X$(T)
1350 PRINT:PRINT:PRINT
1360 PRINT "Address?":PRINT
1370 FOR Z=1 TO 5:LINE INPUT Y$(T,Z):N
EXT Z
1380 NEXT T
1390 SOUND 1,300
1400 PRINT:PRINT:PAPER 2:PEN 0:PRINT "
END OF FILE.DO YOU WANT IT RESAVED (Y/
N)":PAPER 0:PEN 1
1410 LP$=INKEY$:IF LP$="" THEN 1410
1420 IF LP$="Y" OR LP$="y" THEN 1110 E
LSE 610
```

To give you a better idea of what the resultant format looks like, I have constructed a mythical name and address file as well as a not too serious standard letter and have let the program put them together and show you how they output the information onto printer.

This is the ideal program for those with programming ability and knowledge to have a look at. While it works perfectly as it is, programmers or potential programmers can easily add their own little improvements or embellishments to suit their own taste. This is a perfect illustration of one of the positive advantages of developing your own business programs: YOU decide the features you want.



```
----------------------------------------

          Mr Gordeni
          143 London Way
           Hounslow
            Middlesex
             TW19 5TG

----------------------------------------
                       Abacus Consultants,
                        27 Jameson Street,
                         Wetherby,
                          Yorkshire.
                           YK5 4TB.

                          24/4/86
Ref:VF2

Dear Ms Spencer
         Congratulations on entering our
prize draw. You unfortunately have not
won the star prize of a Sinclair C5 car
(worth more than £3Ø) but you have won a
special voucher entitling you to £1Ø off
one of our splendid shower units.
         We have a large range; from the
budget 'Splash' DIY kit (contains bucket
, string and skewer to make the holes)
all the way up to our top-of-the-range
model, the Hokey Kokey deluxe turbo
solar-powered, microprocessor controlled
'Splash II' shower.
         One of our representatives will
be visiting you soon with your voucher
and details of our showers.
         Yours Faithfully,

----------------------------------------

          Ms Spencer
          38 Winstanley Road
           Carvaen
            Glamorgan
             GL24 9NC

----------------------------------------
                       Abacus Consultants,
                        27 Jameson Street,
                         Wetherby,
                          Yorkshire.
                           YK5 4TB.
```



```
                              24/4/86
Ref:VF2

Dear Mr Richards
         Congratulations on entering our
prize draw. You unfortunately have not
won the star prize of a Sinclair C5 car
(worth more than £3Ø) but you have won a
special voucher entitling you to £1Ø off
one of our splendid shower units.
         We have a large range; from the
budget 'Splash' DIY kit (contains bucket
, string and skewer to make the holes)
all the way up to our top-of-the-range
model, the Hokey Kokey deluxe turbo
solar-powered, microprocessor controlled
'Splash II' shower.
         One of our representatives will
be visiting you soon with your voucher
and details of our showers.
         Yours Faithfully,

----------------------------------------

          Mr Richards
          The Grange
           9 Leicester Place
            Milton Keynes.

----------------------------------------
                     Abacus Consultants,
                      27 Jameson Street,
                       Wetherby,
                        Yorkshire.
                         YK5 4TB.

                              24/4/86
Ref:VF2

Dear Mr Smith
         Congratulations on entering our
prize draw. You unfortunately have not
won the star prize of a Sinclair C5 car
(worth more than £3Ø) but you have won a
special voucher entitling you to £1Ø off
one of our splendid shower units.
         We have a large range; from the
budget 'Splash' DIY kit (contains bucket
, string and skewer to make the holes)
all the way up to our top-of-the-range
```

```
model, the Hokey Kokey deluxe turbo
solar-powered, microprocessor controlled
'Splash II' shower.
          One of our representatives will
be visiting you soon with your voucher
and details of our showers.
          Yours Faithfully,

----------------------------------------

          Mr Smith
          17 Parkway Terrace
           Manchester
            M96 LB7

----------------------------------------
```

Mean, Standard Deviation And Variance

This program shows the use of a computer as a calculating tool, albeit of relatively simple statistics. The program is short and straightforward to use; you just follow the prompts and enter the number of items and then the item value, followed by how frequently that value occurs. The computer will then give you the three answers using the formulae below.

Pressing a key will re-run the program.

MEAN ($\bar{x}$) = $\underline{\Sigma f x}$
VARIANCE (σ^2) = Σ
STANDARD DEVIATION (σ) = Σ

```
10 ' MEAN, STANDARD DEVIATION, VARIANC
E
20 '
30 CLS
40 PRINT "                BASIC  STATISTICS
"
50 PRINT "                =================
"
60 PRINT,,,
70 INPUT "How many items";N
80 DIM A(N)
90 DIM B(N)
100 FOR Z=1 TO N
110 PRINT "Enter Item ";Z
120 INPUT A(Z)
130 INPUT "Enter Frequency ";B(Z)
```

```
140 NEXT Z
150 M=0:A=0:B=0
160 FOR Z=1 TO N
170 M=M+B(Z)
180 A=A+B(Z)*A(Z)
190 B=B+B(Z)*A(Z)*A(Z)
200 NEXT Z
210 CLS
220 PRINT "The mean is ";A/M
230 PRINT:PRINT
240 PRINT "Variance is ";B/M-A*A/(M*M)
250 PRINT:PRINT
260 PRINT,, "Standard Deviation is ";S
QR(B/M-A*A/(M*M))
270 PRINT:PRINT:PRINT:PRINT
280 PRINT "                    PRESS A KEY"
290 IF INKEY$<>"" THEN 30 ELSE 290
```

Car Insurance Policy

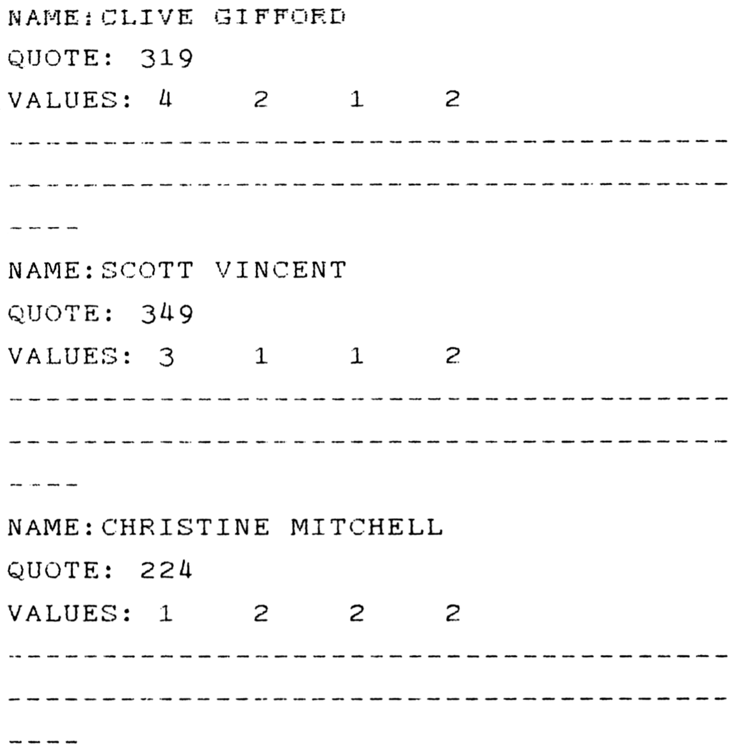
Before I hear insurance men everywhere say 'Oh, if only it was that simple', I must point out that the actual calculations of car insurance pricing made in this program are pure fiction. The whole point of the program is to act as an example of a CMCIP (that is, Computer-Marketed Consumer Interactive Package), a program where the consumer acts as the computer user, with the computer providing a useful function, the whole object of which is to sell a product or service. The program itself supposedly produces policy estimates for car insurance. It asks a number of questions, personalising the prompts from the start, gives the cost of the policy, rejects any 'bad risks', and if the consumer is interested in buying or discussing the policy further then it will endeavour to attract the attention of an assistant. The program also prints out the details of the question answered so that the assistant saves time by not having to go over them again. Though only a simple example, it does show how the Amstrad could be manipulated to provide customer service.

```
10 '          CAR INSURANCE POLICY
20 '
30 INK 0,0:
40 GOSUB 660
50 CLS
60 BORDER 0
70 FOR T=4 TO 15:PEN INT(T/4):PRINT TA
B(T/2);"SAFEGUARD CAR INSURANCE":NEXT
```

```
8Ø PEN 1
9Ø PRINT :PRINT
1ØØ INPUT "ENTER Your Name Please:";V$
11Ø FOR T=1 TO LEN(V$):IF MID$(V$,T,1)
=" " THEN N$=LEFT$(V$,T):T=LEN(V$)
12Ø NEXT T
13Ø '
14Ø '          ASKING QUESTIONS
15Ø '
16Ø FOR J=1 TO 4
17Ø CLS
18Ø PEN 2
19Ø PRINT Q$(J);N$;"?"
2ØØ PRINT:PRINT
21Ø PEN 1
22Ø FOR T=1 TO 5
23Ø IF A$(J,T)="" THEN 25Ø
24Ø PRINT A$(J,T),"press";T
25Ø NEXT T
26Ø A$=INKEY$:IF A$="" THEN 26Ø
27Ø PRINT CHR$(7)
28Ø IF A$<>"1" AND A$<>"2" AND A$<>"3"
 AND A$<>"4" AND A$<>"5" THEN 26Ø
29Ø X=VAL(A$)
3ØØ R(J)=X
31Ø IF A(J,X)=Ø THEN 26Ø
32Ø SC=SC+A(J,X)
33Ø NEXT J
34Ø '
35Ø '            CALCULATIONS!
36Ø '
37Ø CLS:LOCATE 12,12
38Ø PRINT "CALCULATING NOW..."
39Ø IF R(2)<3 AND R(3)=1 THEN SC=SC+3Ø
:IF R(1)=5 THEN SC=SC+1ØØ
4ØØ IF R(4)=1 AND R(2)=5 THEN 88Ø
41Ø BORDER Ø,1
```

```
420 FOR D=1 TO 2000:NEXT D
430 CLS
440 PRINT:PRINT
450 PRINT "The Estimated Cost Of An In
surance      Policy For Your Car ";N$
460 PRINT:PRINT "          is:";SC;"po
unds"
470 PRINT:PRINT
480 PRINT "If You Are Interested In Do
ing Business Then ";:PEN 3:PRINT "Plea
se Press The 'Y' Key":PEN 1:PRINT "If
Not Then We Do Hope To Hear From You S
oon."
490 G=0
500 WHILE G<3000
510 G=G+1
520 IF INKEY$="Y" OR INKEY$="y" THEN 5
50
530 WEND
540 RUN
550 BORDER 0,26
560 IF P$="Y" THEN PRINT #8,"NAME:";V$
:PRINT #8,"QUOTE:";SC:PRINT #8,"VALUES
:";:FOR T=1 TO 4:PRINT #8,R(T);"  ";:N
EXT T
570 IF P$="Y" THEN PRINT #8:PRINT #8,"
--------------------------------------
--------------------------------------
----"
580 IF P$="Y" THEN P$="N"
590 SOUND 1,100,20,15
600 IF INKEY$=" " THEN RUN
610 SOUND 1,140,20,15
620 GOTO 560
630 '
640 '            INITIALISATION
650 '
```

```
660 DIM Q$(4),A$(4,5),A(4,5)
670 SC=97
680 RESTORE
690 '
700 ' SET P$="Y" IF YOU HAVE A PRINTER
710 '
720 P$="Y"
730 FOR T=1 TO 4
740 READ Q$(T)
750 FOR K=1 TO 5:READ A(T,K):NEXT K
760 FOR K=1 TO 5:READ A$(T,K):NEXT K
770 NEXT T
780 RETURN
790 DATA "WHAT CC IS YOUR CAR ",1,15,4
0,80,200
800 DATA "UNDER 1000","1000-1300","130
0-1550","1600-2200","OVER 2200"
810 DATA "HOW OLD ARE YOU ",180,110,70
,30,90
820 DATA "UNDER 20","20-24","25-40","4
1-60","OVER 60"
830 DATA "MALE OR FEMALE ",1,15,0,0,0
840 DATA "MALE","FEMALE","","",""
850 DATA "ANY SERIOUS ILLNESS inc. EPI
LEPSY       ",100,1,0,0,0
860 DATA "YES","NO","","",""
870 '
880 '              REJECTION
890 '
900 CLS
910 PEN 3
920 LOCATE 1,8:PRINT "We are very sorr
y but we cannot accept  you for an ins
urance policy."
930 WHILE INKEY$="":WEND:PEN 1:RUN
```



```
NAME:CLIVE GIFFORD
QUOTE: 319
VALUES: 4    2    1    2
--------------------------------------
--------------------------------------
----
NAME:SCOTT VINCENT
QUOTE: 349
VALUES: 3    1    1    2
--------------------------------------
--------------------------------------
----
NAME:CHRISTINE MITCHELL
QUOTE: 224
VALUES: 1    2    2    2
--------------------------------------
--------------------------------------
----
```

The Super Heroes

The three packages that you are most likely to want to use with your Amstrad are a word-processor, a database and a spreadsheet. Here Scott Vincent, an experienced programmer with many published programs to his credit, provides Amstrad users with his own special version of each.

The programs are powerful, flexible and yet easy to use and add up to a formidable business program collection.

Super-Spread

When this program is run you will be presented with a spreadsheet made up of 299 cells. These are labelled from 0 to 12 across the top of the screen and from A to W down the left-hand side. A cell is identified by specifying first a letter and then a number. A9, B7, T10 and G12 are all valid cells.

A large cursor will appear on the screen, initially at cell A0, showing which is the currently active cell. The simplest way of moving this cursor to another cell is by using the cursor keys. The active cell and its contents are also displayed on the bottom line of the screen. Obviously, all cells are blank to start with.

A cell can contain a string of six characters, a number or a formula. The two arrows in the bottom left corner of the screen signify that you are in DIRECT mode and that the computer is waiting for you to enter something. A string must be preceded by ″ (shift 2) and can be any number of characters long. If it is six or less characters long then it is simply inserted into the current cell. If it is more than six characters long then the first six characters are entered into the current cell, the next six are entered into the cell one to the right and so on for the whole string. To delete the contents of any cell, enter an empty string.

If the first character you type in is not a quote mark then you must be entering either a numeric value or a formula. If the input does not contain any letters, then it is assumed to be a number and it is inserted into the current cell right justified. Although a cell displays only the first six characters, it can hold a lot more. To see the complete contents of a cell, position the cursor over it and look at the bottom line of the screen.

You cannot do calculations in direct mode. So if you entered, say 3*2, the computer would take the first number, 3, and discard the rest. This means that you cannot enter fractions such as ½ but you can enter them in their decimal representation (½ being 0.5). If you do wish to do some calculations or you wish to find the decimal representation of a fraction you must use CALCULATOR mode.

To change to calculator mode from direct mode you must either enter CALCULATOR or CALC, or press the] key (to the left of the ENTER key). Instead of two arrows you will see the word CALC: at the bottom of the screen. Now enter an expression using any of the following: ()^*/\+–=><. For example, a legal expression is: 2(3 + 7 – 4)/16 2 + (9 + 6)(8 –3.75). Note that you may, if you wish, omit the multiplication sign before opening brackets. Only one Boolean operator may be used between two operands, so instead of 3><2 you must use (3<2)+(3>2) which give the same result. Also, 3> = 2 can be written as (3>2) + (3 = 2). An invalid expression will yield an appropriate error message. When you press ENTER, the expression you have typed in is cleared from the screen and is replaced by its result. This may then be used in further calculations or you can just press ENTER to go back to direct mode. The last result displayed is assigned to the CLR key. This allows you to enter the result of a calculation into a cell very easily.

Formulae are expressions that use other cells as operands, for example: 3*F5 – (J3 + J4). These cells may themselves contain a formula or a number. A blank cell is given a value of zero. If a formula is entered into a cell then it is evaluated and the result is displayed in that cell whilst the actual formula appears at the bottom of the screen. If the formula above was entered into cell A0, say, and you then changed the contents of either F5, J3 or J4, the result in A0 would immediately be updated. If a number or formula is entered into any cell then all formulae are immediately recalculated, whether they are affected by the change or not. This is known as IMMEDIATE mode. If you are entering a lot of data, you do not want to have to wait until everything is recalculated each time you enter a single number or formula. The alternative mode to immediate mode stops all calculations and displays all the formulae that you have entered. No more calculations are performed until you go back into immediate mode.

To change to command mode from direct mode you either enter COMMAND or C or press the [key (to the left of the ENTER key). The word COMMAND: should then be displayed at the bottom of the screen. You can now enter one of 15 commands by pressing the first letter of the command. The commands are: BACKGROUND COLOUR, CELL FORMAT, DISPLAY COMMANDS, FOREGROUND COLOUR, GENERAL FORMAT, INVERT COLOURS, JUMP, KEY DEFINE, LOAD, MODE, PRINT, QUIT, SAVE, VAL and ZAP.

To move to another part of the spreadsheet quickly, use the Jump command followed by the cell that you want the cursor to jump to. The Zap command is used to erase everything you have typed in and give you a clean spreadsheet. The colour commands use keys 0 to 9 for colours 0 to 9 and keys A to Q for colours 10 to 26. As soon as a key is pressed, the colour will change accordingly. When using the Mode command, press key 0 for immediate mode or key 1 to display all formulae.

Every cell can be set to one of three formats. These are real, integer and price (two decimal places). There are two commands that set the format of a cell. General Format is initially set to real. If you enter a value into a cell that was previously blank, it is set to General Format; but if the cell already contained a value, the format is left as it was. Use the Cell Format command to change the format of a cell that is not blank. For both commands press key 0, 1 or 2 to select the desired format. If a cell contains a formula whose result takes up more than six characters, the Val command can be used to examine this result more closely. This command displays the result at the bottom of the screen where the formula itself would normally be.

The last few commands are self-explanatory. If you are still not sure what some of the commands do, simply run the program, type in some dummy data, and try them out.

```
10 '        *********************
20 '        *                   *
30 '        *    SUPER-SPREAD   *
40 '        *                   *
50 '        *********************
60 '
70 '
80 '
90 OPENOUT "dummy":MEMORY HIMEM-1:CLOS
EOUT
100 DIM t(22,12),d(22,12),c(298),r(298
),k(160),q(135),s(298),f$(22,12),c$(22
,12)
110 KEY DEF 16,1,128:KEY DEF 66,1,129:
KEY DEF 9,1,130
120 KEY 128,"":KEY 129,CHR$(13)+"[Q"+C
HR$(13):KEY 130,""
130 ON ERROR GOTO 4640
140 MODE 2:INK 0,0:BORDER 0:INK 1,26:P
APER 0:PEN 1
150 PRINT"    00     01     02     03
04     05     06     07     08     09     10
    11     12"
160 FOR n=2 TO 24:LOCATE 1,n:PRINT CHR
$(63+n):NEXT
170 MOVE 0,384:DRAW 639,384
180 MOVE 15,15:DRAW 15,399
190 x=0:y=0:f=0:fc=0:dis=241:genf=0:fc
ol=26:bcol=0
200 z$=CHR$(65+x)
210 IF y<10 THEN z$=z$+CHR$(48+y) ELSE
 z$=z$+"1"+CHR$(38+y)
220 IF f=1 AND t(x,y)=3 THEN w$=LEFT$(
f$(x,y),78-y*6) ELSE w$=LEFT$(c$(x,y),
6)
230 IF LEN(w$)<6 THEN w$=w$+SPACE$(6-L
EN(w$))
```

```
240 LOCATE 3+y*6,2+x:PRINT CHR$(24);w$
;CHR$(24)
250 LOCATE 42,25:PRINT z$;":  "
260 tt=t(x,y)
270 IF tt=0 THEN t$="blank"
280 IF tt=1 OR tt=2 THEN t$=c$(x,y)
290 IF tt=3 THEN t$=LEFT$(f$(x,y),33)
300 LOCATE 47,25:PRINT t$;SPACE$(33-LE
N(t$))
310 LOCATE 4,25:PRINT CHR$(1);CHR$(9);
"  ";CHR$(1);CHR$(8);
320 k$="":s=0
330 a$=INKEY$:IF a$="" GOTO 330
340 a=ASC(a$)
350 IF k$<>"" OR a<240 OR a>243 GOTO 4
10
360 dis=a
370 LOCATE 3+y*6,2+x:PRINT w$
380 x=x+(x>0 AND a=240)-(x<22 AND a=24
1)
390 y=y+(y>0 AND a=242)-(y<12 AND a=24
3)
400 GOTO 200
410 IF K$="" GOTO 440
420 IF a=127 THEN k$=LEFT$(k$,LEN(k$)-
1):GOTO 520
430 IF a=13 GOTO 540 ELSE GOTO 480
440 IF a=34 THEN s=1:GOTO 510
450 IF a=13 THEN a=dis:GOTO 370
460 IF a$="[" GOTO 1120
470 IF a$="]" GOTO 3070
480 IF s=0 THEN a$=UPPER$(a$) ELSE GOT
O 510
490 a=ASC(a$)
500 IF a<40 OR a=44 OR a=58 OR a=59 OR
 a=63 OR a=64 OR a>87 AND a<>92 AND a<
>94 GOTO 330
```

```
510 k$=k$+a$
520 IF k$="" THEN s=0
530 LOCATE 6,25:PRINT k$;" ";CHR$(1);C
HR$(8);" ";:GOTO 330
540 LOCATE 6,25:PRINT SPACE$(LEN(k$)+2
)
550 IF k$="C" OR k$="COMMAND" GOTO 112
0
560 IF k$="CALC" OR k$="CALCULATOR" GO
TO 3070
570 REM **** entry ****
580 IF s=0 GOTO 800
590 IF t(x,y)=3 THEN GOSUB 680
600 REM ** enter string **
610 k$=RIGHT$(k$,LEN(k$)-1):IF k$<>""
GOTO 630
620 t(x,y)=0:c$(x,y)="":GOTO 200
630 t(x,y)=2:m=LEN(k$):IF m<7 THEN c$(
x,y)=k$:GOTO 200
640 c$(x,y)=LEFT$(k$,6):k$=RIGHT$(k$,m
-6)
650 IF y=12 GOTO 200
660 LOCATE 3+y*6,2+x:PRINT c$(x,y):y=y
+1:GOTO 630
670 REM * delete formula *
680 n=0
690 IF r(n)<>x THEN n=n+1:GOTO 690
700 IF c(n)<>y THEN n=n+1:GOTO 690
710 fc=fc-1:IF fc=n GOTO 730
720 FOR m=n TO fc-1:r(m)=r(m+1):c(m)=c
(m+1):NEXT
730 n=INT((LEN(LEFT$(f$(x,y),78-y*6))-
1)/6)
740 IF f=0 OR n=0 THEN RETURN
750 FOR m=1 TO n
760 IF f=1 AND t(x,y+m)=3 THEN ww$=LEF
T$(f$(x,y+m),78-(y+m)*6) ELSE ww$=LEFT
```

```
$(c$(x,y+m),6)
770 IF LEN(ww$)<6 THEN ww$=ww$+SPACE$(
6-LEN(ww$))
780 LOCATE 3+(y+m)*6,2+x:PRINT ww$:NEX
T
790 RETURN
800 m=1
810 j$=MID$(k$,m,1):IF j$>="A" AND j$<
="W" GOTO 980
820 IF m<LEN(k$) THEN m=m+1:GOTO 810
830 REM **** number ****
840 t$=k$:j$=MID$(k$,2,1):IF (ASC(k$)=
45 OR ASC(k$)=43) AND (j$<"0" OR j$>"9
") THEN er=1:e$="invalid number":GOSUB
 4560:GOTO 530
850 IF t(x,y)=3 THEN GOSUB 680
860 IF t(x,y)=0 THEN d(x,y)=genf
870 v=VAL(k$):IF d(x,y)=1 THEN v=INT(v
+0.5)
880 IF d(x,y)<2 THEN k$=STR$(v):GOTO 9
20
890 v=INT(v*100+0.5):k$=STR$(v/100)
900 IF v/100=INT(v/100) THEN k$=k$+".0
0":GOTO 920
910 IF v/10=INT(v/10) THEN k$=k$+"0"
920 IF ASC(k$)=32 THEN k$=RIGHT$(k$,LE
N(k$)-1)
930 IF LEN(k$)<6 THEN k$=SPACE$(6-LEN(
k$))+k$
940 LOCATE 3+y*6,2+x:PRINT LEFT$(k$,6)
950 c$(x,y)=k$:t(x,y)=1:IF f=0 THEN GO
SUB 3230
960 GOTO 200
970 REM **** formula ****
980 m=1:t$=k$
990 j$=MID$(k$,m,1):IF j$<"A" OR j$>"W
" GOTO 1050
```

```
1000 h$=MID$(k$,m+1,1):j$=MID$(k$,m+2,
1)
1010 IF h$="0" AND j$>="0" AND j$<="9"
 THEN k$=LEFT$(k$,m)+RIGHT$(k$,LEN(k$)
-1-m):GOTO 1000
1020 v=VAL(MID$(k$,m+1,3))
1030 IF h$<"0" OR h$>"9" OR v>12 THEN
er=1:e$="no such cell":GOSUB 4560:GOTO
 530
1040 m=m+1-(v>9)
1050 m=m+1:IF m<=LEN(k$) GOTO 990
1060 IF t(x,y)=3 THEN GOSUB 680
1070 IF t(x,y)=0 THEN d(x,y)=genf
1080 r(fc)=x:c(fc)=y:fc=fc+1
1090 f$(x,y)=k$:t(x,y)=3:IF f=0 THEN G
OSUB 3230
1100 GOTO 200
1110 REM **** commands ****
1120 LOCATE 4,25:PRINT "COMMAND: ";
1130 a$=UPPER$(INKEY$):IF a$="" GOTO 1
130
1140 a=ASC(a$)
1150 IF a=13 GOTO 1230
1160 IF a$<>"D" GOTO 1280
1170 PRINT"display commands"
1180 GOSUB 3020:IF a=127 GOTO 1320
1190 v$="Background colour      Cell f
ormat      Foreground colour":GOSUB 12
40
1200 v$="General format      Invert co
lours      Jump      Key define":GOSUB
 1240
1210 v$="Load      Mode      Print
  Quit      Save      Val      Zap":GO
SUB 1240
1220 LOCATE 42,25:PRINT SPACE$(5)
1230 LOCATE 4,25:PRINT SPACE$(38):GOTO
```

```
 200
1240 LOCATE 4,25:PRINT SPACE$(8);v$:SP
ACE$(68-LEN(v$))
1250 a$=INKEY$:IF a$="" GOTO 1250
1260 IF ASC(a$)<>13 GOTO 1250
1270 RETURN
1280 IF a$<>"M" GOTO 1580
1290 PRINT"mode - ";
1300 a$=INKEY$:IF a$="" GOTO 1300
1310 IF ASC(a$)<>127 GOTO 1330
1320 LOCATE 13,25:PRINT SPACE$(29):LOC
ATE 13,25:GOTO 1130
1330 IF a$<>"0" GOTO 1470
1340 PRINT"immediate":GOSUB 3020
1350 IF a=127 GOTO 1320
1360 IF f=0 GOTO 310
1370 f=0
1380 IF fc=0 GOTO 310
1390 FOR n=0 TO fc-1
1400 xx=r(n):yy=c(n):c$(xx,yy)="":p=IN
T((LEN(LEFT$(f$(xx,yy),78-yy*6))-1)/6)
1410 FOR m=0 TO p:ww$=LEFT$(c$(xx,yy+m
),6)
1420 IF LEN(ww$)<6 THEN ww$=ww$+SPACE$
(6-LEN(ww$))
1430 LOCATE 3+(yy+m)*6,2+xx:PRINT ww$
1440 NEXT:NEXT
1450 GOSUB 3230
1460 GOTO 200
1470 IF a$<>"1" GOTO 1300
1480 PRINT"display formulae":GOSUB 302
0
1490 IF a=127 GOTO 1320
1500 IF f=1 GOTO 310
1510 f=1
1520 IF fc=0 GOTO 310
1530 FOR n=0 TO fc-1
```

```
1540 xx=r(n):yy=c(n):v$=f$(xx,yy):LOCA
TE 3+yy*6,2+xx:PRINT LEFT$(v$,78-yy*6)
;
1550 IF LEN(v$)<6 THEN PRINT SPACE$(6-
LEN(v$))
1560 NEXT
1570 GOTO 200
1580 IF a$<>"P" GOTO 1680
1590 PRINT"print"
1600 GOSUB 3020:IF a=127 GOTO 1320
1610 LOCATE 4,25:PRINT"* P R I N T I N
 G *"
1620 FOR n=0 TO 22:FOR m=0 TO 12
1630 w$=LEFT$(c$(n,m),6)
1640 IF LEN(w$)<6 THEN w$=w$+SPACE$(6-
LEN(w$))
1650 PRINT #8,w$;
1660 NEXT:PRINT #8:NEXT
1670 GOTO 1230
1680 IF a$<>"V" GOTO 1750
1690 PRINT"val ";z$
1700 GOSUB 3020:IF a=127 GOTO 1320
1710 IF t(x,y)<3 THEN t$="not a formul
a!":GOTO 300
1720 IF f=0 THEN t$=c$(x,y):GOTO 300
1730 IF c$(x,y)="" THEN t$="not yet ca
lculated" ELSE t$=c$(x,y)+"  (when las
t calculated)"
1740 GOTO 300
1750 IF a$<>"G" GOTO 1790
1760 PRINT"general format - ";
1770 GOSUB 1900:IF v=127 GOTO 1320
1780 genf=v:GOTO 310
1790 IF a$<>"C" GOTO 1980
1800 PRINT"cell ";z$;" format - ";
1810 GOSUB 1900:IF v=127 GOTO 1320
1820 d(x,y)=v:IF v=0 OR t(x,y)=0 OR t(
x,y)=2 OR f=1 GOTO 310
1830 k$=c$(x,y):a=VAL(k$):IF v=1 THEN
k$=STR$(INT(a+Ø.5)):GOTO 187Ø
184Ø a=INT(a*1ØØ+Ø.5):k$=STR$(a/1ØØ)
185Ø IF a/1ØØ=INT(a/1ØØ) THEN k$=k$+".
ØØ":GOTO 187Ø
186Ø IF a/1Ø=INT(a/1Ø) THEN k$=k$+"Ø"
187Ø IF ASC(k$)=32 THEN k$=RIGHT$(k$,L
EN(k$)-1)
188Ø IF LEN(k$)<6 THEN k$=SPACE$(6-LEN
(k$))+k$
189Ø c$(x,y)=k$:GOTO 2ØØ
19ØØ a$=INKEY$:IF a$="" GOTO 19ØØ
191Ø v=ASC(a$):IF v=127 THEN RETURN
192Ø IF a$<"Ø" OR a$>"2" GOTO 19ØØ
193Ø IF a$="Ø" THEN PRINT"real"
194Ø IF a$="1" THEN PRINT"integer"
195Ø IF a$="2" THEN PRINT"price"
196Ø GOSUB 3Ø2Ø:IF a=127 THEN v=127:RE
TURN
197Ø v=v-48:RETURN
198Ø IF a$<>"J" GOTO 215Ø
199Ø PRINT"jump to cell - ";
2ØØØ a$=UPPER$(INKEY$):IF a$="" GOTO 2
ØØØ
2Ø1Ø IF ASC(a$)=127 GOTO 132Ø
2Ø2Ø IF a$<"A" OR a$>"W" GOTO 2ØØØ
2Ø3Ø PRINT a$;:xx=ASC(a$)-65
2Ø4Ø a$=INKEY$:IF a$="" GOTO 2Ø4Ø
2Ø5Ø IF ASC(a$)=127 GOTO 132Ø
2Ø6Ø IF a$<"Ø" OR a$>"9" GOTO 2Ø4Ø
2Ø7Ø PRINT a$;:yy=VAL(a$)
2Ø8Ø a$=INKEY$:IF a$="" GOTO 2Ø8Ø
2Ø9Ø IF ASC(a$)=127 GOTO 132Ø
21ØØ IF ASC(a$)=13 GOTO 213Ø
211Ø IF a$<"Ø" OR a$>"9" GOTO 2Ø8Ø
212Ø yy=1Ø*yy+VAL(a$):IF yy>12 GOTO 13
2Ø
```

```
213Ø LOCATE 4,25:PRINT SPACE$(38)
214Ø LOCATE 3+y*6,2+x:PRINT w$:x=xx:y=
yy:GOTO 2ØØ
215Ø IF a$<>"I" GOTO 22ØØ
216Ø PRINT"invert colours"
217Ø GOSUB 3Ø2Ø:IF a=127 GOTO 132Ø
218Ø n=fcol:fcol=bcol:bcol=n
219Ø INK Ø,bcol:BORDER bcol:INK 1,fcol
:GOTO 31Ø
22ØØ IF a$<>"F" GOTO 228Ø
221Ø PRINT"foreground colour -";fcol
222Ø a$=UPPER$(INKEY$):IF a$="" GOTO 2
22Ø
223Ø a=ASC(a$):IF a=127 GOTO 132Ø
224Ø IF a=13 GOTO 123Ø
225Ø IF a$<"Ø" OR a$>"9" AND a$<"A" OR
 a$>"Q" GOTO 222Ø
226Ø fcol=a+7*(a>64)-48
227Ø LOCATE 32,25:PRINT fcol;" ":INK 1
,fcol:GOTO 222Ø
228Ø IF a$<>"B" GOTO 236Ø
229Ø PRINT"background colour -";bcol
23ØØ a$=UPPER$(INKEY$):IF a$="" GOTO 2
3ØØ
231Ø a=ASC(a$):IF a=127 GOTO 132Ø
232Ø IF a=13 GOTO 123Ø
233Ø IF a$<"Ø" OR a$>"9" AND a$<"A" OR
 a$>"Q" GOTO 23ØØ
234Ø bcol=a+7*(a>64)-48
235Ø LOCATE 32,25:PRINT bcol;" ":INK Ø
,bcol:BORDER bcol:GOTO 23ØØ
236Ø IF a$<>"K" GOTO 245Ø
237Ø LOCATE 42,25:PRINT SPACE$(38)
238Ø LOCATE 13,25:LINE INPUT"key defin
e COPY - ",k$
239Ø REM ** use 'shift @' for <RETURN>
 **
```

```
2400 IF k$="" GOTO 2440
2410 FOR n=1 TO LEN(k$)
2420 IF MID$(k$,n,1)="¦" THEN k$=LEFT$
(k$,n-1)+CHR$(13)+RIGHT$(k$,LEN(k$)-n)
2430 NEXT:KEY 130,k$
2440 LOCATE 4,25:PRINT SPACE$(76):GOTO
 200
2450 IF a$<>"L" GOTO 2670
2460 INPUT"load ",f$
2470 IF f$="" GOTO 1320
2480 dot=0
2490 FOR n=1 TO LEN(f$):IF MID$(f$,n,1
)="." THEN dot=n-ABS(99*dot)
2500 NEXT
2510 IF dot<0 OR dot=1 GOTO 1320
2520 IF dot=0 THEN f$=LEFT$(f$,8)+".sp
r":GOTO 2550
2530 IF dot>9 THEN f$=LEFT$(f$,8)+RIGH
T$(f$,LEN(f$)+1-dot):dot=9
2540 f$=LEFT$(f$,dot+3)
2550 LOCATE 4,25:PRINT"* L O A D I N G
 *    (";f$;")"
2560 OPENIN "!"+f$
2570 INPUT #9,x,y,f,fc,genf,fcol,bcol
2580 FOR n=0 TO 22:FOR m=0 TO 12:INPUT
 #9,c$(n,m),f$(n,m),t(n,m),d(n,m):NEXT
:NEXT
2590 FOR n=0 TO fc:INPUT #9,c(n),r(n):
NEXT
2600 CLOSEIN
2610 INK 0,bcol:BORDER bcol:INK 1,fcol
2620 FOR n=0 TO 22:FOR m=0 TO 12
2630 IF f=1 AND t(n,m)=3 THEN w$=LEFT$
(f$(n,m),78-m*6) ELSE w$=LEFT$(c$(n,m)
,6)
2640 IF LEN(w$)<6 THEN w$=w$+SPACE$(6-
LEN(w$))
```

```
265Ø LOCATE 3+m*6,2+n:PRINT w$:NEXT:NE
XT
266Ø GOTO 123Ø
267Ø IF a$<>"S" GOTO 284Ø
268Ø INPUT"save ",f$
269Ø IF f$="" GOTO 132Ø
27ØØ dot=Ø
271Ø FOR n=1 TO LEN(f$):IF MID$(f$,n,1
)="." THEN dot=n-ABS(99*dot)
272Ø NEXT
273Ø IF dot<Ø OR dot=1 GOTO 132Ø
274Ø IF dot=Ø THEN f$=LEFT$(f$,8)+".sp
r":GOTO 277Ø
275Ø IF dot>9 THEN f$=LEFT$(f$,8)+RIGH
T$(f$,LEN(f$)+1-dot):dot=9
276Ø f$=LEFT$(f$,dot+3)
277Ø LOCATE 4,25:PRINT"* S A V I N G *
    (";f$;")"
278Ø OPENOUT "!"+f$
279Ø WRITE #9,x,y,f,fc,genf,fcol,bcol
28ØØ FOR n=Ø TO 22:FOR m=Ø TO 12:WRITE
 #9,c$(n,m),f$(n,m),t(n,m),d(n,m):NEXT
:NEXT
281Ø FOR n=Ø TO fc:WRITE #9,c(n),r(n):
NEXT
282Ø CLOSEOUT
283Ø GOTO 123Ø
284Ø IF a$<>"Z" GOTO 292Ø
285Ø PRINT"zap"
286Ø GOSUB 3Ø2Ø:IF a=127 GOTO 132Ø
287Ø FOR n=1 TO 15:SOUND 7,2ØØ,2,7:SOU
ND 7,1ØØ,2,7:NEXT
288Ø LOCATE 4,25:PRINT "**  Z A P  **"
;SPACE$(1Ø);"ARE YOU SURE?";SPACE$(4Ø)
289Ø a$=UPPER$(INKEY$):IF a$<>"Y" AND
a$<>"N" GOTO 289Ø
29ØØ LOCATE 4,25:PRINT SPACE$(76)
```

```
2910 IF a$="Y" THEN RUN ELSE GOTO 200
2920 IF a$<>"Q" GOTO 1130
2930 PRINT"quit"
2940 GOSUB 3020:IF a=127 GOTO 1320
2950 INK 0,0:BORDER 0:INK 1,26:PAPER 0
:PEN 1
2960 FOR n=1 TO 4
2970 SOUND 7,50,10,7:FOR m=1 TO 100:NE
XT:NEXT
2980 LOCATE 4,25:PRINT "**  Q U I T  *
*";SPACE$(8);"ARE YOU SURE?";SPACE$(40
)
2990 a$=UPPER$(INKEY$):IF a$<>"Y" AND
a$<>"N" GOTO 2990
3000 IF a$="N" THEN LOCATE 4,25:PRINT
SPACE$(76):GOTO 200
3010 KEY DEF 16,1,16:KEY DEF 66,0,252:
KEY DEF 9,1,224:CLS:END
3020 a$=INKEY$:IF a$="" GOTO 3020
3030 a=ASC(a$):IF a<>13 AND a<>127 GOT
O 3020
3040 IF a=13 THEN LOCATE 4,25:PRINT SP
ACE$(38)
3050 RETURN
3060 REM **** calculator ****
3070 LOCATE 4,25:PRINT "CALC: ";CHR$(1
36)
3080 LOCATE 42,25:PRINT SPACE$(38);
3090 k$="":s=0
3100 a$=INKEY$:IF a$="" GOTO 3100
3110 a=ASC(a$)
3120 IF a=127 AND k$<>"" THEN k$=LEFT$
(k$,LEN(k$)-1):GOTO 3210
3130 IF a=13 AND (k$="" OR s=1) THEN L
OCATE 8,25:PRINT SPACE$(3+LEN(k$)):GOT
O 250
3140 IF a<>13 GOTO 3190
```

```
3150 LOCATE 10,25:PRINT SPACE$(LEN(k$)
+1)
3160 GOSUB 3690:IF er=1 GOTO 3210
3170 k$=STR$(k(0)):IF ASC(k$)=32 THEN
k$=RIGHT$(k$,LEN(k$)-1)
3180 s=1:KEY 128,k$:GOTO 3210
3190 IF a<40 OR a=44 OR a=58 OR a=59 O
R a>62 AND a<>92 AND a<>94 GOTO 3100
3200 s=0:k$=k$+a$
3210 LOCATE 10,25:PRINT k$;CHR$(136);"
 ";:GOTO 3100
3220 REM **** expand and display formu
lae ****
3230 IF fc=0 THEN RETURN
3240 et=1
3250 FOR u=0 TO fc-1
3260 xx=r(u):yy=c(u):cof=0:m=1:k$=f$(x
x,yy):s(xx*13+yy)=0
3270 j$=MID$(k$,m,1):IF j$<"A" OR j$>"
W" GOTO 3350
3280 rr=ASC(j$)-65:cc=VAL(MID$(k$,m+1,
2))
3290 IF cc>9 THEN v=m+2 ELSE v=m+1
3300 IF t(rr,cc)=2 THEN k$="zz":GOTO 3
360
3310 IF t(rr,cc)=0 THEN j$=CHR$(95):GO
TO 3340
3320 IF t(rr,cc)=1 THEN j$=CHR$(95+et)
:q(et)=VAL(c$(rr,cc)):et=et+1:GOTO 334
0
3330 j$=CHR$(231+rr)+CHR$(231+cc):cof=
1
3340 k$=LEFT$(k$,m-1)+j$+RIGHT$(k$,LEN
(k$)-v)
3350 m=m+1:IF m<LEN(k$) GOTO 3270
3360 IF cof=1 THEN c$(xx,yy)=k$ ELSE G
OSUB 3550
```

```
3370 NEXT
3380 ok=0:rec=0:fp=0
3390 xx=r(fp):yy=c(fp):IF s(xx*13+yy)=
1 THEN ok=ok+1:GOTO 3480
3400 n=1:k$=c$(xx,yy)
3410 c1=ASC(MID$(k$,n,1)):IF c1<231 GO
TO 3460
3420 c1=c1-231:c2=ASC(MID$(k$,n+1,1))-
231
3430 IF s(c1*13+c2)=0 GOTO 3480
3440 k$=LEFT$(k$,n-1)+CHR$(95+et)+RIGH
T$(k$,LEN(k$)-n-1)
3450 q(et)=VAL(c$(c1,c2)):et=et+1
3460 n=n+1:IF n<=LEN(k$) GOTO 3410
3470 rec=1:ok=ok+1:GOSUB 3550
3480 fp=fp+1:IF fp<fc GOTO 3390
3490 IF ok=fc THEN RETURN
3500 IF rec=1 GOTO 3380
3510 FOR fp=0 TO fc-1
3520 xx=r(fp):yy=c(fp)
3530 IF s(xx*13+yy)=0 THEN k$="yy":GOS
UB 3550
3540 NEXT:RETURN
3550 GOSUB 3700:IF er=0 GOTO 3590
3560 v$=" error":LOCATE 3+yy*6,2+xx:PR
INT" ERROR"
3570 FOR n=1 TO 1000:NEXT
3580 GOTO 3660
3590 v=k(0):IF d(xx,yy)=1 THEN v=INT(v
+0.5)
3600 IF d(xx,yy)<2 THEN v$=STR$(v):GOT
O 3640
3610 v=INT(v*100+0.5):v$=STR$(v/100)
3620 IF v/100=INT(v/100) THEN v$=v$+".
00":GOTO 3640
3630 IF v/10=INT(v/10) THEN v$=v$+"0"
3640 IF ASC(v$)=32 THEN v$=RIGHT$(v$,L
EN(v$)-1)
3650 IF LEN(v$)<6 THEN v$=SPACE$(6-LEN
(v$))+v$
3660 c$(xx,yy)=v$:s(xx*13+yy)=1
3670 LOCATE 3+yy*6,2+xx:PRINT LEFT$(v$
,6)
3680 RETURN
3690 REM ** evaluate **
3700 er=0:t$=k$
3710 IF k$="zz" THEN er=1:e$="string i
n formula":GOTO 4560
3720 IF k$="yy" THEN er=1:e$="recursiv
e formula":GOTO 4560
3730 m=1
3740 j$=MID$(k$,m+1,1):IF j$<>"(" GOTO
 3790
3750 j=ASC(MID$(k$,m,1))
3760 IF j=41 OR j>47 AND j<58 OR j>94
THEN k$=LEFT$(k$,m)+"*"+RIGHT$(k$,LEN(
k$)-m):m=m+1
3770 IF j=45 THEN k$=LEFT$(k$,m)+"1*"+
RIGHT$(k$,LEN(k$)-m):m=m+2
3780 GOTO 3820
3790 IF j$<>"-" GOTO 3820
3800 j=ASC(MID$(k$,m,1))
3810 IF j=41 OR j>47 AND j<58 OR j>94
THEN k$=LEFT$(k$,m)+"+"+RIGHT$(k$,LEN(
k$)-m):m=m+1
3820 IF m+2<LEN(k$) THEN m=m+1:GOTO 37
40
3830 m=1:v=0:kk=LEN(k$):ON ERROR GOTO
4630
3840 j$=MID$(k$,m,1):j=ASC(j$)
3850 IF j<45 OR j=47 OR j>57 AND j<95
GOTO 3980
3860 IF j>94 THEN n=LEN(k$)-m:m=m-1:k(
v)=q(j-95):GOTO 3960
```

```
3870 v$=""
3880 v$=v$+j$
3890 IF m=kk GOTO 3950
3900 m=m+1
3910 j$=MID$(k$,m,1):j=ASC(j$)
3920 IF j=46 OR j>47 AND j<58 GOTO 388
0
3930 IF j>94 THEN n=LEN(k$)-m:m=m-2:k(
v)=-q(j-95):GOTO 3960
3940 m=m-1
3950 n=LEN(k$)-m:m=m-LEN(v$):k(v)=VAL(
v$)
3960 k$=LEFT$(k$,m)+CHR$(95+v)+RIGHT$(
k$,n)
3970 m=m+1:v=v+1:kk=LEN(k$)
3980 IF m<kk THEN m=m+1:GOTO 3840
3990 ON ERROR GOTO 4640
4000 b1=0:b2=0
4010 m=1
4020 j$=MID$(k$,m,1)
4030 IF j$="(" THEN b1=m
4040 IF b1>0 AND j$=")" THEN b2=m:GOTO
 4090
4050 IF j$=")" THEN er=1:e$="missing '
('":GOTO 4560
4060 IF m<LEN(k$) THEN m=m+1:GOTO 4020
4070 IF b1>0 AND b2=0 THEN er=1:e$="mi
ssing ')'":GOTO 4560
4080 p$=k$:GOSUB 4140:RETURN
4090 p$=MID$(k$,b1+1,b2-b1-1)
4100 GOSUB 4140:IF er=1 THEN RETURN
4110 k$=LEFT$(k$,b1-1)+p$+RIGHT$(k$,LE
N(k$)-b2)
4120 GOTO 4000
4130 REM ** numeric expression **
4140 IF p$="" GOTO 4500
4150 s$="^":m=1
```

```
416Ø GOSUB 446Ø:IF er=1 THEN RETURN
417Ø IF s$="^" THEN k(t1)=k(t1)^k(t2):
GOTO 416Ø
418Ø s$="*":m=1
419Ø GOSUB 446Ø:IF er=1 THEN RETURN
42ØØ IF s$="*" THEN k(t1)=k(t1)*k(t2):
GOTO 419Ø
421Ø s$="/":m=1
422Ø GOSUB 446Ø:IF er=1 THEN RETURN
423Ø IF s$="" GOTO 426Ø
424Ø IF k(t2)<>Ø THEN k(t1)=k(t1)/k(t2
):GOTO 422Ø
425Ø er=1:e$="cannot divide by zero":G
OTO 456Ø
426Ø s$="\":m=1
427Ø REM *** NOTE: '\' is taken to mea
n INT(x/y) ***
428Ø GOSUB 446Ø:IF er=1 THEN RETURN
429Ø IF s$="" GOTO 432Ø
43ØØ IF k(t2)=Ø GOTO 425Ø
431Ø k(t1)=INT(k(t1)/k(t2)):GOTO 428Ø
432Ø s$="+":m=1
433Ø GOSUB 446Ø:IF er=1 THEN RETURN
434Ø IF s$="+" THEN k(t1)=k(t1)+k(t2):
GOTO 433Ø
435Ø REM * relational operators *
436Ø s$="=":m=1
437Ø GOSUB 446Ø:IF er=1 THEN RETURN
438Ø IF s$="=" THEN k(t1)=k(t1)=k(t2):
GOTO 437Ø
439Ø s$=">":m=1
44ØØ GOSUB 446Ø:IF er=1 THEN RETURN
441Ø IF s$=">" THEN k(t1)=k(t1)>k(t2):
GOTO 44ØØ
442Ø s$="<":m=1
443Ø GOSUB 446Ø:IF er=1 THEN RETURN
444Ø IF s$="<" THEN k(t1)=k(t1)<k(t2):
```

```
GOTO 4430
4450 RETURN
4460 IF MID$(p$,m,1)=s$ GOTO 4490
4470 IF m=LEN(p$) THEN s$="":RETURN
4480 m=m+1:GOTO 4460
4490 IF m>1 AND m<LEN(p$) GOTO 4510
4500 er=1:e$="illegal expression":GOTO
 4560
4510 t1=ASC(MID$(p$,m-1,1))-95:t2=ASC(
MID$(p$,m+1,1))-95
4520 IF t1<0 OR t2<0 GOTO 4500
4530 p$=LEFT$(p$,m-1)+RIGHT$(p$,LEN(p$
)-1-m)
4540 m=m-1:RETURN
4550 REM * error occured *
4560 IF er=2 THEN LOCATE 4,25:PRINT SP
ACE$(6)
4570 LOCATE 10,25:PRINT CHR$(7);"ERROR
 - ";e$;
4580 k$=t$:n=0
4590 n=n+1:IF n<1000 AND INKEY$="" GOT
O 4590
4600 LOCATE 10,25:PRINT SPACE$(8+LEN(e
$))
4610 IF er=2 GOTO 310
4620 RETURN
4630 er=1:e$="illegal value":RESUME 45
60
4640 er=2:e$="in line"+STR$(ERL):RESUM
E 4560
```

Super-Base

This program uses a number of machine code routines to perform tasks which would be too slow if they were written in BASIC. Using machine code allowed me to use a data structure known as a 'linked list' which makes insertion and deletion of records very fast and simple operations to perform. It also allows a large number of records to be sorted in a few seconds rather than a few minutes as the order of the records in a linked list is totally independent of their physical location in memory.

The first BASIC program reads in the machine code from data statements and pokes it into memory. Type this program into your computer and save it. Now type in the main BASIC listing and save this. Finally, load the first program in again and run it. Once the machine code has been poked in, it will be saved as a binary file called MCODE. If you typed in all of the data statements correctly, you will not need this program any more. If you are saving the programs on tape, make sure that you save MCODE straight after the main listing.

When you run the main BASIC program, it will reserve some memory and then load in MCODE. A full stop is used as a prompt, and when it appears, the program is waiting for you to enter a command. You only need to press the first letter to obtain a command word. Once you have done this you can either press DEL to abort or ENTER to execute it. It is best to enter some test data to start with to make sure that you typed in the machine code correctly. If the program crashes, you will have to reset the computer and load in the first BASIC program again. The checksums should have found most of the errors for you, but, if not, you must check every line of data against the listing given and, once corrected, run the program again to create a new version of MCODE.

The database program knows the following commands: CREATE, KILL, NEW LABEL, MODE, RECORDS, DISPLAY, EDIT, BROWSE, FIND, INTERCHANGE, ORDER, PRINT, LOAD and SAVE. Most of the commands will not actually do anything until a file has been defined and information has been inserted.

To define a file use the CREATE command. The following example shows a file being created with four fields to each record:

```
 create
How many fields? 4
FIELD 0
  label: name
  type (Char or Num): C
  size (no. of chars): 20
FIELD 1
  label: age
  type (Char or Num): N
  size (no. of chars): 2
FIELD 2
  label: occupation
  type (Char or Num): C
  size (no. of chars): 30
FIELD 3
  label: tel. no.
  type (Char or Num): C
  size (no. of chars): 20
```

```
 create
File already created
FIELD 0 name            C   20
FIELD 1 age             N    2
FIELD 2 occupation      C   30
FIELD 3 tel. no.        C   20
```

KILL erases the current file which then allows you to create a new one.

NEW LABEL lets you change the label of the specified field. MODE is used to select either mode 1 or mode 2. RECORDS tells you the total number of records in the file, the amount of memory used by the file and the amount of memory you have spare. DISPLAY allows you to set out a record how you want. When you first use this command, the field labels will be displayed down the left-hand side of the screen. To select a field, press the key corresponding to that field. So, to select field 6 press key 6, to select field 10 press A and so on. The selected field will change colour and you can then move it around the screen using the cursor keys. Once it is in the desired position, press ENTER. Now you can either select another field and position that, or you can press ENTER again to go back to the prompt.

EDIT is used to add new records to the file, delete records, change existing records and print records. If the file contains more than one record when you use this command, you will be asked which record you wish to edit. If the file is empty, it is assumed that you wish to insert a new record. The cursor is positioned at the start of field 0. If you type in some information and press ENTER, the cursor moves to field 1 and so on. If you are at the last field and you press ENTER, the cursor moves to the start of the next record. To add a new record to an existing file, move to the last field of the last record in the file and press ENTER. Use the cursor keys to move left and right in a field or to move to the previous or next field. SHIFT left moves you to the start of the current field and SHIFT right moves you to the end. Pressing SHIFT up moves you to the previous record, while SHIFT down moves you to the next record. Also, CONTROL up takes you to the first record in the file and CONTROL down takes you to the last record. To delete the record currently being displayed, press the CLR key. The program will ask for confirmation, so don't worry if you press it accidentally. The DEL key works as normal. If you wish to print the current record on a printer, press CTRL O. To quit editing, press CTRL Q.

BROWSE can be used to examine all of the records in turn very quickly. It will either display a whole record or any single field of a record. FIND locates one or more records satisfying a given condition such as: name="Bloggs" or date>=1978. You can type in as many characters as you wish when searching through a character field, regardless of the size of that field. So if you had a name field, for example, and you searched for name>"G", the program would display all the names that began with a letter between H and Z inclusive. If the condition was name="Fre", then the name "Fred" would satisfy it and so would "Freda". INTERCHANGE swaps the positions of two records in the file. ORDER will rearrange all the records into alphabetic or numeric order, depending on the type of the field specified. PRINT sends all of the records to the printer in order. A file can be saved using the SAVE command followed by filename and, similarly, it can be loaded using LOAD filename.

```
10 '        *********************
20 '        *                   *
30 '        *     SUPER-BASE    *
40 '        *                   *
50 '        *********************
```

```
6Ø '
7Ø '
8Ø '
9Ø get=2ØØ38:insert=2ØØ5Ø:deleterec=2Ø
129:findnext=2Ø172:findrec=2Ø186:swapr
ec=2Ø212
1ØØ reclenptr=2Ø285:fstptr=2Ø287:curpt
r=2Ø289:emptyptr=2Ø291:fldlenptr=2Ø293
11Ø DEF FNdeek(addr)=PEEK(addr)+256*PE
EK(addr+1)
12Ø DEF FNlsb(num)=num-INT(num/256)*25
6
13Ø DEF FNmsb(num)=INT(num/256)
14Ø IF HIMEM=19997 THEN topmem=FNdeek(
19998):GOTO 19Ø
15Ø OPENOUT "dummy":MEMORY HIMEM-1:CLO
SEOUT
16Ø topmem=HIMEM:MEMORY 19997
17Ø POKE 19998,FNlsb(topmem):POKE 1999
9,FNmsb(topmem)
18Ø LOAD"mcode"
19Ø tot=Ø:nof=-1:mde=1:stp=1
2ØØ ON ERROR GOTO 447Ø
21Ø MODE 1:PAPER Ø:PEN 1:BORDER 6:INK
Ø,6:INK 1,26:INK 2,24:INK 3,14
22Ø PEN 3-mde:LOCATE 1,25:PRINT"Databa
se":PRINT CHR$(164);" 1986  S.L.Vincen
t"
23Ø PEN 1:PRINT:PRINT". ";
24Ø a$=UPPER$(INKEY$):IF a$="" GOTO 24
Ø
25Ø IF ASC(a$)<32 GOTO 23Ø
26Ø REM ****** CREATE ******
27Ø IF a$<>"C" GOTO 61Ø
28Ø PRINT"create";
29Ø GOSUB 4Ø2Ø
3ØØ IF a=127 GOTO 23Ø
```

```
310 IF nof=-1 GOTO 380
320 PRINT"  File already created"
330 FOR f%=0 TO nof
340 PRINT"  FIELD";f%;TAB(12);f$(f%);T
AB(30);
350 IF t(f%)=0 THEN PRINT"C"; ELSE PRI
NT"N";
360 PRINT TAB(33);s(f%):NEXT
370 GOTO 230
380 reclen=0
390 INPUT"  How many fields";f%
400 IF f%<=0 GOTO 230
410 IF f%>40 GOTO 390
420 nof=f%-1:DIM f$(nof):DIM t(nof):DI
M x(nof):DIM y(nof):DIM s(nof)
430 FOR f%=0 TO nof
440 PRINT"  FIELD";f%
450 PRINT"    ";:INPUT"label:",f$(f%)
460 PRINT"    type (Char or Num):";
470 a$=UPPER$(INKEY$):IF a$<>"C" AND a
$<>"N" GOTO 470
480 PRINT a$:t(f%)=-(a$="N")
490 PRINT"    ";:INPUT"size (no. of ch
ars):",s(f%)
500 IF s(f%)<=0 OR s(f%)>255 GOTO 490
510 POKE fldlenptr+f%,s(f%):reclen=rec
len+s(f%)
520 NEXT
530 GOSUB 4080
540 FOR f%=0 TO nof
550 x(f%)=18*INT(f%/23)+LEN(f$(f%))+2
560 IF x(f%)>40*mde THEN x(f%)=40*mde
570 y(f%)=(f% MOD 23)+3
580 NEXT
590 GOTO 230
600 REM ****** KILL ******
610 IF a$<>"K" GOTO 710
```

```
620 PRINT"kill";
630 GOSUB 4020
640 IF a=127 GOTO 230
650 IF nof=-1 THEN RUN
660 PRINT CHR$(7);"  O.K. to delete cu
rrent file? ";
670 a$=UPPER$(INKEY$):IF a$<>"Y" AND a
$<>"N" GOTO 670
680 PRINT a$:IF a$="N" GOTO 230
690 RUN
700 REM ****** NEW LABEL ******
710 IF a$<>"N" GOTO 810
720 PRINT"new label";
730 GOSUB 4020
740 IF a=127 OR nof=-1 GOTO 230
750 GOSUB 4180
760 PRINT"  old - ";f$(f%)
770 INPUT"  new - ",f$(f%)
780 IF x(f%)-LEN(f$(f%))<2 THEN x(f%)=
LEN(f$(f%))+2
790 GOTO 230
800 REM ****** MODE ******
810 IF a$<>"M" GOTO 930
820 PRINT"mode";
830 a$=INKEY$:IF a$="" GOTO 830
840 a=ASC(a$)
850 IF a=127 THEN PRINT CHR$(17):GOTO
230
860 IF a=13 THEN PRINT:GOTO 230
870 IF a$<"1" OR a$>"2" GOTO 830
880 mde=VAL(a$):MODE mde
890 IF mde=2 GOTO 230
900 FOR f%=0 TO nof:IF x(f%)>40 THEN x
(f%)=40
910 NEXT:GOTO 230
920 REM ****** RECORDS ******
930 IF a$<>"R" GOTO 1030
```

```
940 PRINT"records";
950 GOSUB 4020
960 IF a=127 GOTO 230
970 PRINT"  total number of records -"
:tot
980 m=(reclen+2)*tot
990 PRINT"  memory used by file -";m;"
bytes"
1000 PRINT "  memory spare -";topmem-m
-reclen-19997;"bytes"
1010 GOTO 230
1020 REM ***** DISPLAY *****
1030 IF a$<>"D" GOTO 1320
1040 PRINT"display";
1050 GOSUB 4020
1060 IF a=127 OR nof=-1 GOTO 230
1070 PRINT"  include labels on printou
t?"
1080 a$=UPPER$(INKEY$):IF a$<>"Y" AND
a$<>"N" GOTO 1080
1090 stp=-1*(a$="Y")
1100 CLS:PEN 5-2*mde:PRINT"field:"
1110 PRINT STRING$(40*mde,"_")
1120 PEN 3-mde:FOR f%=0 TO nof
1130 LOCATE x(f%)-LEN(f$(f%))-1,y(f%):
PRINT f$(f%);":"
1140 NEXT
1150 a$=UPPER$(INKEY$):IF a$=CHR$(13)
THEN LOCATE 1,25:GOTO 230
1160 IF a$<"0" OR a$>CHR$(48+nof-7*(no
f>9)) OR a$>"9" AND a$<"A" GOTO 1150
1170 f%=ASC(a$)-48+7*(ASC(a$)>64)
1180 PEN 5-2*mde:LOCATE 7,1:PRINT f%;"
 ";f$(f%);"  ";
1190 IF t(f%)=0 THEN PRINT"C"; ELSE PR
INT"N";
1200 PRINT" ";s(f%)
```

```
1210 PEN 1:b=LEN(f$(f%))
1220 LOCATE x(f%)-b-1,y(f%):PRINT f$(f
%);":"
1230 a$=INKEY$:IF a$="" GOTO 1230
1240 x=x(f%):y=y(f%):a=ASC(a$)
1250 IF a=13 THEN LOCATE 7,1:PRINT SPA
CE$(40*mde-6):GOTO 1120
1260 x=x+(a=242 AND x>b+2)-(a=243 AND
x<40*mde)
1270 y=y+(a=240 AND y>3)-(a=241 AND y<
25)
1280 IF x=x(f%) AND y=y(f%) GOTO 1230
1290 LOCATE x(f%)-b-1,y(f%):PRINT SPAC
E$(b+1):x(f%)=x:y(f%)=y
1300 GOTO 1220
1310 REM ****** EDIT ******
1320 IF a$<>"E" GOTO 1950
1330 PRINT"edit";
1340 GOSUB 4020
1350 IF a=127 OR nof=-1 GOTO 230
1360 IF tot<2 THEN r%=0:GOTO 1400
1370 INPUT"  record";r%
1380 IF r%<0 GOTO 1370
1390 IF r%>=tot THEN r%=tot-1
1400 CLS:PEN 5-2*mde:PRINT"RECORD"
1410 PRINT STRING$(40*mde,"_")
1420 PEN 3-mde:FOR f%=0 TO nof
1430 LOCATE x(f%)-LEN(f$(f%))-1,y(f%):
PRINT f$(f%);":"
1440 NEXT:f%=0
1450 PEN 5-2*mde:CALL findrec,r%
1460 IF r%<tot THEN LOCATE 7,1:PRINT r
%;SPACE$(12):GOTO 1480
1470 LOCATE 8,1:PRINT"to be inserted"
1480 PEN 1:FOR e%=0 TO nof:f$=SPACE$(s
(e%)):CALL get,e%,@f$
1490 LOCATE x(e%),y(e%):PRINT LEFT$(f$
,40*mde+1-x(e%)):NEXT
1500 x=x(f%):y=y(f%):b=x(f%):e=b+s(f%)
-1
1510 f$=SPACE$(s(f%)):CALL get,f%,@f$
1520 LOCATE b,y:PRINT LEFT$(f$,40*mde+
1-b)
1530 LOCATE x,y:PRINT CHR$(24);MID$(f$
,x+1-b,1);CHR$(24)
1540 a$=INKEY$:IF a$="" GOTO 1540
1550 a=ASC(a$)
1560 IF a=240 AND f%>0 THEN GOSUB 1740
:f%=f%-1:GOTO 1500
1570 IF a=241 AND f%<nof THEN GOSUB 17
40:f%=f%+1:GOTO 1500
1580 IF a=242 AND x>b THEN x=x-1:GOTO
1520
1590 IF a=243 AND x<e AND x<40*mde THE
N x=x+1:GOTO 1520
1600 IF a=244 AND r%>0 THEN GOSUB 1740
:r%=r%-1:GOTO 1450
1610 IF a=245 THEN GOSUB 1740:GOTO 171
0
1620 IF a=246 AND x>b THEN x=b:GOTO 15
20
1630 IF a<>247 OR x=e GOTO 1660
1640 IF e>40*mde THEN x=40*mde ELSE x=
e
1650 GOTO 1520
1660 IF a=248 AND r%>0 THEN r%=0:GOTO
1450
1670 IF a=249 AND r%<tot-1 THEN r%=tot
-1:GOTO 1450
1680 IF a<>13 GOTO 1820
1690 GOSUB 1740
1700 IF f%<nof THEN f%=f%+1:GOTO 1500
1710 IF r%=tot GOTO 1530
1720 IF a=13 THEN f%=0
```

```
1730 r%=r%+1:GOTO 1450
1740 LOCATE b,y:PRINT LEFT$(f$,40*mde+
1-b)
1750 IF r%<tot THEN CALL insert,f%,@f$
:RETURN
1760 b$=SPACE$(s(f%)):IF f$=b$ THEN RE
TURN
1770 empty=FNdeek(emptyptr)
1780 IF empty+reclen+3<topmem GOTO 180
0
1790 PEN 5-2*mde:LOCATE 10,2:PRINT CHR
$(7);"OUT OF MEMORY!!":PEN 1:RETURN
1800 CALL insert,f%,@f$:tot=tot+1:PEN
5-2*mde
1810 LOCATE 7,1:PRINT r%;SPACE$(12):PE
N 1:RETURN
1820 IF a=127 AND x>b THEN f$=LEFT$(f$
,x-b-1)+RIGHT$(f$,e+1-x)+" ":x=x-1:GOT
O 1520
1830 IF a<>16 OR r%=tot GOTO 1890
1840 PEN 5-2*mde:LOCATE 1,1:PRINT CHR$
(7);"O.K. to delete record?"
1850 a$=UPPER$(INKEY$):IF a$<>"Y" AND
a$<>"N" GOTO 1850
1860 LOCATE 1,1:PRINT"RECORD";r%;SPACE
$(14)
1870 IF a$="Y" THEN CALL deleterec:tot
=tot-1:GOTO 1450
1880 PEN 1:GOTO 1540
1890 IF a=15 THEN GOSUB 4320
1900 IF a=17 THEN GOSUB 1740:LOCATE 1,
25:GOTO 230
1910 IF a<32 OR a>126 GOTO 1540
1920 f$=LEFT$(f$,x-b)+a$+MID$(f$,1+x-b
,e-x):x=x-(x<e AND x<40*mde)
1930 GOTO 1520
1940 REM ****** BROWSE ******
```

```
1950 IF a$<>"B" GOTO 2190
1960 PRINT"browse";
1970 GOSUB 4020
1980 IF a=127 OR tot=0 GOTO 230
1990 PRINT"  display complete record?
";
2000 a$=UPPER$(INKEY$):IF a$<>"Y" AND
a$<>"N" GOTO 2000
2010 PRINT a$:IF a$="Y" THEN e%=-1:PRI
NT:GOTO 2030
2020 GOSUB 4180:e%=f%
2030 PRINT"  press SPACE BAR to scroll
"
2040 PRINT"  press ENTER to quit"
2050 PRINT:r%=0:CALL findrec,0
2060 a$=INKEY$:IF a$="" GOTO 2060
2070 a=ASC(a$):IF a=13 GOTO 230
2080 IF a<>32 GOTO 2060
2090 PRINT:IF e%=-1 GOTO 2120
2100 f$=SPACE$(s(e%)):CALL get,e%,@f$
2110 PRINT f$:GOTO 2160
2120 FOR f%=0 TO nof
2130 f$=SPACE$(s(f%)):CALL get,f%,@f$
2140 PRINT f$;" ";:NEXT
2150 PRINT
2160 r%=r%+1:IF r%<tot THEN CALL findn
ext:GOTO 2060
2170 GOTO 230
2180 REM ****** FIND ******
2190 IF a$<>"F" GOTO 2860
2200 PRINT"find";
2210 GOSUB 4020
2220 IF a=127 OR tot=0 GOTO 230
2230 GOSUB 4180
2240 PRINT"  expression: ";f$(f%);
2250 LINE INPUT a$:a$=a$+"."
2260 PRINT:eq=0:sm=0:gr=0
```

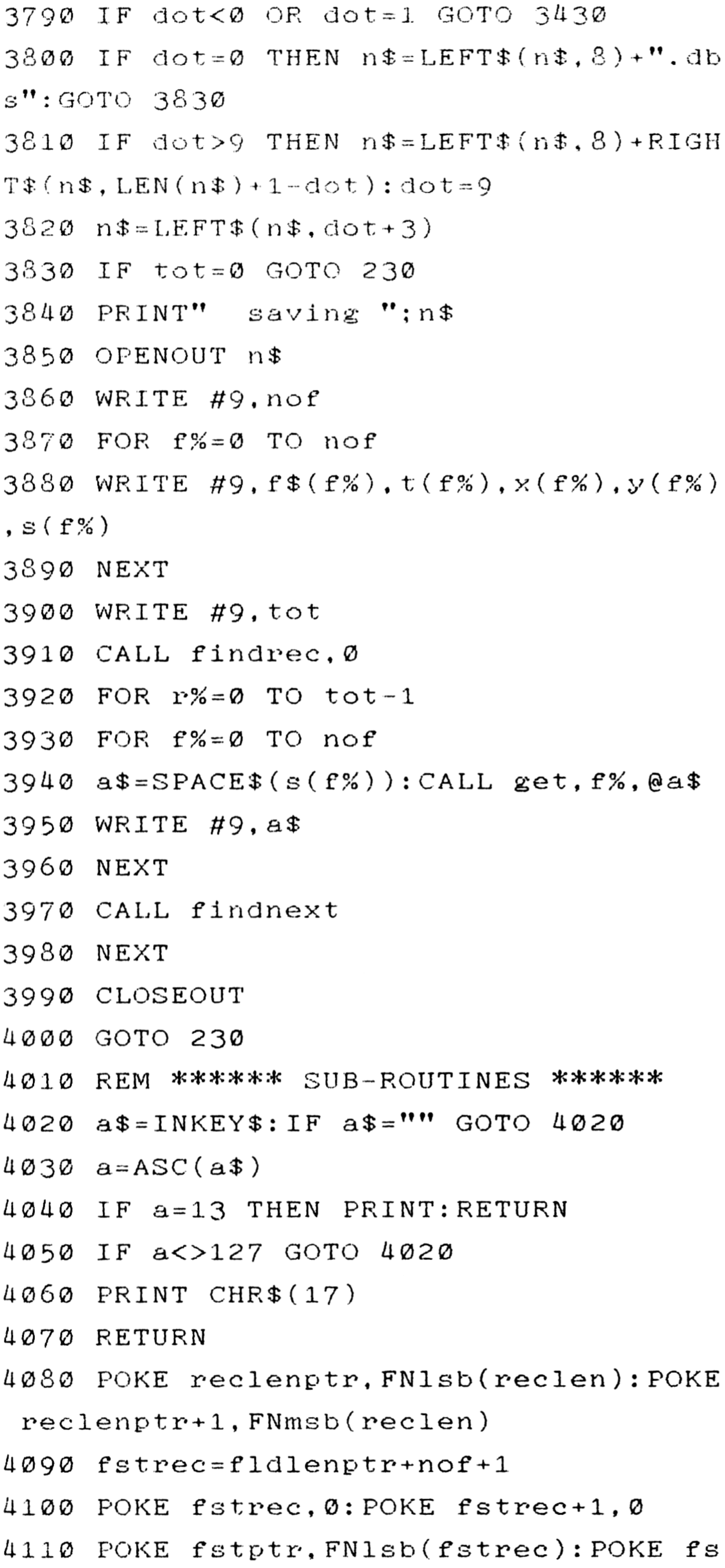
```
2270 IF ASC(a$)=61 THEN eq=1:a$=RIGHT$
(a$,LEN(a$)-1)
2280 IF ASC(a$)=60 THEN sm=1:a$=RIGHT$
(a$,LEN(a$)-1):GOTO 2270
2290 IF ASC(a$)=62 THEN gr=1:a$=RIGHT$
(a$,LEN(a$)-1):GOTO 2270
2300 IF eq=0 AND sm=0 AND gr=0 THEN PR
INT"  ERROR operator missing":GOTO 224
0
2310 a$=LEFT$(a$,LEN(a$)-1)
2320 IF a$="" THEN PRINT"  ERROR opera
nd missing":GOTO 2240
2330 IF t(f%)=0 GOTO 2710
2340 IF ASC(a$)=34 THEN PRINT"  ERROR
field type is numeric":GOTO 2240
2350 ON ERROR GOTO 2500
2360 v=VAL(a$):b$=SPACE$(s(f%))
2370 ON ERROR GOTO 2490
2380 r%=0:CALL findrec,r%
2390 GOSUB 2530:IF r%=-1 THEN PRINT"
search failed":GOTO 230
2400 GOSUB 2620
2410 GOSUB 2670
2420 GOSUB 2510:IF r%=-1 GOTO 2480
2430 PEN 5-2*mde:LOCATE 12,1:PRINT"pre
ss SPACE BAR for next"
2440 k$=INKEY$:IF k$="" GOTO 2440
2450 IF k$=" " THEN LOCATE 12,1:PRINT
SPACE$(24):GOTO 2410
2460 ON ERROR GOTO 4470
2470 LOCATE 1,24:GOTO 230
2480 PEN 5-2*mde:LOCATE 12,1:PRINT"end
 of search":GOTO 2460
2490 n=0:RESUME NEXT
2500 PRINT"  ERROR number expected":RE
SUME 2230
2510 IF r%=tot-1 THEN r%=-1:RETURN
```

```
2520 CALL findnext:r%=r%+1
2530 CALL get,f%,@b$
2540 n=VAL(b$)
2550 IF n=v AND eq=1 OR n<v AND sm=1 O
R n>v AND gr=1 THEN RETURN
2560 GOTO 2510
2570 IF r%=tot-1 THEN r%=-1:RETURN
2580 CALL findnext:r%=r%+1
2590 CALL get,f%,@b$
2600 IF b$=a$ AND eq=1 OR b$<a$ AND sm
=1 OR b$>a$ AND gr=1 THEN RETURN
2610 GOTO 2570
2620 CLS:PEN 5-2*mde:PRINT"RECORD"
2630 PRINT STRING$(40*mde,"_")
2640 PEN 3-mde:FOR e%=0 TO nof
2650 LOCATE x(e%)-LEN(f$(e%))-1,y(e%):
PRINT f$(e%);":"
2660 NEXT:RETURN
2670 PEN 5-2*mde:LOCATE 7,1:PRINT r%
2680 PEN 1:FOR e%=0 TO nof:f$=SPACE$(s
(e%)):CALL get,e%,@f$
2690 LOCATE x(e%),y(e%):PRINT LEFT$(f$
,40*mde+1-x(e%))
2700 NEXT:RETURN
2710 IF ASC(a$)<>34 THEN PRINT"  ERROR
 quote marks missing":GOTO 2240
2720 a$=RIGHT$(a$,LEN(a$)-1)
2730 IF RIGHT$(a$,1)=CHR$(34) THEN a$=
LEFT$(a$,LEN(a$)-1)
2740 IF a$=""" GOTO 2320
2750 a$=LEFT$(a$,s(f%)):b$=a$
2760 r%=0:CALL findrec,r%
2770 GOSUB 2590:IF r%=-1 THEN PRINT"
search failed":GOTO 230
2780 GOSUB 2620
2790 GOSUB 2670
2800 GOSUB 2570:IF r%=-1 GOTO 2480
```

```
2810 PEN 5-2*mde:LOCATE 12,1:PRINT"pre
ss SPACE BAR for next"
2820 k$=INKEY$:IF k$="" GOTO 2820
2830 IF k$=" " THEN LOCATE 12,1:PRINT
SPACE$(24):GOTO 2790
2840 GOTO 2470
2850 REM ****** INTERCHANGE ******
2860 IF a$<>"I" GOTO 2990
2870 PRINT"interchange";
2880 GOSUB 4020
2890 IF a=127 OR tot<2 GOTO 230
2900 IF tot=2 THEN q%=0:r%=1:GOTO 2960
2910 INPUT"  first record";q%
2920 IF q%<0 OR q%>tot-1 GOTO 2910
2930 INPUT"  second record";r%
2940 IF r%<0 OR r%>tot-1 GOTO 2930
2950 IF q%=r% GOTO 230
2960 CALL swaprec,q%,r%
2970 PRINT"  records";q%;"and";r%;"swa
pped":GOTO 230
2980 REM ****** ORDER ******
2990 IF a$<>"O" GOTO 3330
3000 PRINT"order";
3010 GOSUB 4020
3020 IF a=127 OR tot<2 GOTO 230
3030 GOSUB 4180
3040 PRINT"  sorting into ";
3050 IF t(f%)=0 THEN PRINT"alphabetica
l"; ELSE PRINT"numerical";
3060 PRINT" order"
3070 a$=SPACE$(s(f%)):b$=a$:t=t(f%)
3080 IF t=1 THEN ON ERROR GOTO 3310
3090 FOR r%=1 TO tot-1
3100 CALL findrec,r%-1
3110 CALL get,f%,@a$
3120 CALL findnext
3130 CALL get,f%,@b$
```

```
314Ø IF t=Ø GOTO 318Ø
315Ø b=VAL(a$)
316Ø a=b:b=VAL(b$)
317Ø IF b>=a GOTO 329Ø ELSE GOTO 319Ø
318Ø IF b$>=a$ GOTO 329Ø
319Ø q%=r%-1
32ØØ CALL swaprec,q%,q%+1
321Ø IF q%=Ø GOTO 329Ø
322Ø q%=q%-1:CALL findrec,q%
323Ø CALL get,f%,@a$
324Ø IF t=Ø GOTO 328Ø
325Ø b=VAL(a$)
326Ø a=b:b=VAL(b$)
327Ø IF a>b GOTO 32ØØ ELSE GOTO 329Ø
328Ø IF a$>b$ GOTO 32ØØ
329Ø NEXT
33ØØ ON ERROR GOTO 447Ø:GOTO 23Ø
331Ø b=Ø:RESUME NEXT
332Ø REM ****** PRINT ******
333Ø IF a$<>"P" GOTO 342Ø
334Ø PRINT"print";
335Ø GOSUB 4Ø2Ø
336Ø IF a=127 OR tot=Ø GOTO 23Ø
337Ø FOR r%=Ø TO tot-1
338Ø PRINT"  printing record";r%
339Ø GOSUB 432Ø:NEXT
34ØØ GOTO 23Ø
341Ø REM ****** LOAD ******
342Ø IF a$<>"L" GOTO 373Ø
343Ø INPUT"load ",n$
344Ø IF n$="" GOTO 23Ø
345Ø dot=Ø
346Ø FOR n=1 TO LEN(n$):IF MID$(n$,n,1
)="." THEN dot=n-ABS(99*dot)
347Ø NEXT
348Ø IF dot<Ø OR dot=1 GOTO 343Ø
349Ø IF dot=Ø THEN n$=LEFT$(n$,8)+".db
```

```
s":GOTO 3520
3500 IF dot>9 THEN n$=LEFT$(n$,8)+RIGH
T$(n$,LEN(n$)+1-dot):dot=9
3510 n$=LEFT$(n$,dot+3)
3520 IF nof>-1 THEN PRINT"  File alrea
dy exists":PRINT"  Use kill to erase i
t":GOTO 230
3530 PRINT"  loading ";n$
3540 OPENIN n$
3550 reclen=0:INPUT #9,nof
3560 DIM f$(nof):DIM t(nof):DIM x(nof)
:DIM y(nof):DIM s(nof)
3570 FOR f%=0 TO nof
3580 INPUT #9,f$(f%),t(f%),x(f%),y(f%)
,s(f%)
3590 POKE fldlenptr+f%,s(f%):reclen=re
clen+s(f%)
3600 NEXT
3610 GOSUB 4080
3620 INPUT #9,tot
3630 CALL findrec,0
3640 FOR r%=0 TO tot-1
3650 FOR f%=0 TO nof
3660 INPUT #9,a$:CALL insert,f%,@a$
3670 NEXT
3680 CALL findnext
3690 NEXT
3700 CLOSEIN
3710 GOTO 230
3720 REM ****** SAVE ******
3730 IF a$<>"S" GOTO 240
3740 INPUT"save ",n$
3750 IF n$="" GOTO 230
3760 dot=0
3770 FOR n=1 TO LEN(n$):IF MID$(n$,n,1
)="." THEN dot=n-ABS(99*dot)
3780 NEXT
```



```
3790 IF dot<0 OR dot=1 GOTO 3430
3800 IF dot=0 THEN n$=LEFT$(n$,8)+".db
s":GOTO 3830
3810 IF dot>9 THEN n$=LEFT$(n$,8)+RIGH
T$(n$,LEN(n$)+1-dot):dot=9
3820 n$=LEFT$(n$,dot+3)
3830 IF tot=0 GOTO 230
3840 PRINT"  saving ";n$
3850 OPENOUT n$
3860 WRITE #9,nof
3870 FOR f%=0 TO nof
3880 WRITE #9,f$(f%),t(f%),x(f%),y(f%)
,s(f%)
3890 NEXT
3900 WRITE #9,tot
3910 CALL findrec,0
3920 FOR r%=0 TO tot-1
3930 FOR f%=0 TO nof
3940 a$=SPACE$(s(f%)):CALL get,f%,@a$
3950 WRITE #9,a$
3960 NEXT
3970 CALL findnext
3980 NEXT
3990 CLOSEOUT
4000 GOTO 230
4010 REM ****** SUB-ROUTINES ******
4020 a$=INKEY$:IF a$="" GOTO 4020
4030 a=ASC(a$)
4040 IF a=13 THEN PRINT:RETURN
4050 IF a<>127 GOTO 4020
4060 PRINT CHR$(17)
4070 RETURN
4080 POKE reclenptr,FNlsb(reclen):POKE
 reclenptr+1,FNmsb(reclen)
4090 fstrec=fldlenptr+nof+1
4100 POKE fstrec,0:POKE fstrec+1,0
4110 POKE fstptr,FNlsb(fstrec):POKE fs
```

```
tptr+1,FNmsb(fstrec)
4120 POKE curptr,FNlsb(fstrec):POKE cu
rptr+1,FNmsb(fstrec)
4130 emptyrec=fstrec+2+reclen
4140 FOR n=fstrec+2 TO emptyrec-1:POKE
 n,32:NEXT
4150 POKE emptyrec,0:POKE emptyrec+1,0
4160 POKE emptyptr,FNlsb(emptyrec):POK
E emptyptr+1,FNmsb(emptyrec)
4170 RETURN
4180 a=INT(nof/2)
4190 FOR f%=0 TO a
4200 IF LEN(f$(f%))=0 THEN a$="<BLANK>
" ELSE a$=f$(f%)
4210 PRINT TAB(4);CHR$(48+f%-7*(f%>9))
;" ";a$;
4220 g%=a+f%+1:IF g%>nof THEN PRINT:GO
TO 4250
4230 IF LEN(f$(g%))=0 THEN a$="<BLANK>
" ELSE a$=f$(g%)
4240 PRINT TAB(24);CHR$(48+g%-7*(g%>9)
);" ";a$
4250 NEXT
4260 PRINT"  field? ";
4270 a$=UPPER$(INKEY$)
4280 IF a$<"0" OR a$>CHR$(48+nof-7*(no
f>9)) OR a$>"9" AND a$<"A" GOTO 4270
4290 f%=ASC(a$)-48+7*(ASC(a$)>64)
4300 PRINT f$(f%):PRINT
4310 RETURN
4320 FOR n=500 TO 300 STEP -40:SOUND 1
,n,1,7:NEXT
4330 CALL findrec,r%
4340 FOR n=1 TO 25:pd=0
4350 p%=-1:FOR m=0 TO nof
4360 IF y(m)<>n OR x(m)<=pd GOTO 4390
4370 IF p%=-1 THEN p%=m:py=x(m):GOTO
```

```
4390
4380 IF x(m)<py THEN p%=m:py=x(m)
4390 NEXT
4400 IF p%=-1 THEN PRINT #8:GOTO 4460
4410 pd=py
4420 a$=SPACE$(s(p%)):CALL get,p%,@a$
4430 IF stp=0 THEN PRINT #8,TAB(x(p%))
;a$;:GOTO 4350
4440 b$=f$(p%)
4450 PRINT #8,TAB(x(p%)-LEN(b$)-1);b$;
":";a$;:GOTO 4350
4460 NEXT:RETURN
4470 PRINT"  ERROR in line";ERL
4480 RESUME 230
```

```
10 '       SUPER-BASE MACHINE CODE
20 '
30 lineno=190
40 sum=0:num=lineno
50 MEMORY 19999
60 FOR n=20000 TO 20284
70 READ a$:a=VAL("&"+a$)
80 sum=sum+a:POKE n,a
90 IF (n+1) MOD 20>0 GOTO 130
100 READ a$:IF VAL("&"+a$)=sum THEN su
m=0:num=num+30:GOTO 130
110 PRINT"ERROR in line";num;",";num+1
0;"or";num+20
120 STOP
130 NEXT
140 READ a$:IF VAL("&"+a$)=sum GOTO 17
0
150 PRINT"ERROR in line";num;"or";num+
10
160 STOP
170 SAVE"mcode",b,20000,285
180 STOP
```

```
190 '
200 '         MACHINE CODE
210 '
220 '         AND CHECKSUMS
230 '
240 DATA 2A,41,4F,23,23,B7,C8,01,45,4F
250 DATA 16,00,F5,0A,5F,F1,19,03,3D,20
260 DATA 05F2
270 DATA F7,C9,E5,DD,6E,00,DD,66,01,4E
280 DATA 23,5E,23,56,E1,06,00,C9,DD,7E
290 DATA 0987
300 DATA 02,CD,20,4E,CD,36,4E,ED,B0,C9
310 DATA DD,7E,02,CD,20,4E,CD,36,4E,EB
320 DATA 09C8
330 DATA ED,B0,2A,41,4F,23,7E,B7,C0,ED
340 DATA 5B,43,4F,72,2B,73,13,1A,B7,20
350 DATA 085D
360 DATA 15,13,ED,4B,3D,4F,D5,E1,13,36
370 DATA 20,ED,B0,22,43,4F,36,00,23,36
380 DATA 06EB
390 DATA 00,C9,32,44,4F,AF,12,1B,1A,32
400 DATA 43,4F,AF,12,13,13,ED,4B,3D,4F
410 DATA 05F3
420 DATA 0B,D5,E1,13,36,20,ED,B0,C9,ED
430 DATA 4B,41,4F,21,3F,4F,7E,B9,20,06
440 DATA 0864
450 DATA 23,7E,2B,B8,28,06,5E,23,56,EB
460 DATA 18,F0,0A,77,03,23,0A,77,3A,44
470 DATA 0622
480 DATA 4F,02,0B,3A,43,4F,02,ED,43,43
490 DATA 4F,C9,2A,41,4F,5E,23,56,7A,B7
500 DATA 0677
510 DATA C8,ED,53,41,4F,C9,2A,3F,4F,DD
520 DATA 4E,00,DD,46,01,78,B1,28,09,5E
530 DATA 0820
540 DATA 23,56,EB,0B,78,B1,20,F7,22,41
550 DATA 4F,C9,DD,4E,00,DD,46,01,78,B1
```

```
560 DATA Ø8A2
570 DATA 2Ø,Ø5,21,3F,4F,18,Ø7,ØB,2A,3F
580 DATA 4F,CD,E3,4E,E5,DD,4E,Ø2,DD,46
590 DATA Ø6E9
600 DATA Ø3,78,B1,2Ø,Ø5,21,3F,4F,18,Ø7
610 DATA ØB,2A,3F,4F,CD,E3,4E,D1,3E,Ø2
620 DATA Ø5F1
630 DATA 4E,23,46,C5,EB,4E,23,46,C5,EB
640 DATA 7Ø,2B,71,EB,D1,C1,7Ø,2B,71,69
650 DATA Ø9CC
660 DATA 6Ø,3D,2Ø,E8,C9
670 DATA Ø26E
```

Super-Word

This is a word-processor which right justifies text. This means that extra spaces are added to a line of text, if necessary, so that you get a straight margin down the right-hand side of the text as well as down the left. This makes it look much neater.

A large portion of the program is machine code, and so you need to enter this, and save it, using the hex loader program given. Enter and run the program and you will see the number 10000 displayed on the screen. This is the address where the machine code will be poked into memory. If you look at the list of numbers given, you will see that the first number in the first row is also 10000. This number is followed by ten two-digit hexadecimal numbers. Type these into your computer, but do not put a space between each one. When you have typed in all ten numbers, press ENTER. The address 10010 will then be displayed and you must type in the next ten numbers in the same way. Do this for all of the numbers given, making sure that you always read the numbers from the line whose address matches that shown on the screen. When you have entered all of the numbers, the program will save them in a binary file called WORDCODE. The main BASIC program will load in this file when it is run.

Your text file can have lines of up to 80 characters wide but you are given the option of having margins to the left and to the right of the text. The size of a margin is specified in characters. For no margin, enter a size of 0. If your printer is 80 characters across, a good size to use is 10 for both margins. Run the program and try entering some text. Don't worry about getting to the end of a line, just keep typing. If you are typing a word that 'falls off' the end of a line, that word will be moved to the start of the next line down and the line that it was removed from will then be right justified. This happens very quickly and so you do not have to wait for the computer to catch up with you again. You only need to use ENTER when you are starting a new paragraph. The DEL key deletes the character immediately before the cursor. If you enter a word on a line that already contains text you will overwrite it. To insert text you must position the cursor at the place where you want it inserted and then press CTRL W.

This moves the text to the right of the cursor down a line and gives you some space to insert the text. To insert a blank line use CTRL I, and to delete a line use CTRL D.

Inserting and deleting words may mess up the right-hand margin. To re-justify the whole text file again, use CTRL J. If a line of text is followed by a blank line, then the computer assumes that it is the end of a paragraph and does not justify it.

Although the screen is wide enough to display the whole width of the text, it only displays 21 lines at a time. If the cursor is at the bottom of the screen and you are typing in a line of text, when you reach the end of the line the screen will scroll up eight lines. If the cursor is at the top of the screen and you press DEL the screen will scroll down eight lines unless the top of the text file is reached before then. If you look at line 50 of the basic listing, you will see the SCROLL is set to 8. This is the number of lines that the screen scrolls each time you reach the edge. You can change it to a different value if you wish.

The current column and row of the cursor are constantly displayed on the bottom line of the screen. This is not the screen position but is the position in the text file. The total number of lines in the text file is also displayed. The cursor can be moved around in the text using the cursor keys. To move to the start of a line use SHIFT left and to move to the end of a line use SHIFT right. To go to the top or bottom of the file use SHIFT up and SHIFT down respectively.

Use CTRL N to start a new file. Obviously, this command destroys any file already in memory and so it asks for confirmation before doing so. CTRL S asks you for a filename and saves the current text file, while CTRL L loads in a previously saved text file. CTRL P sends the complete file to the printer.

There is an underline option in the program, but this may not work on some printers because the control codes may be different. Look at line 650 which turns underline on and line 660 which turns it off. Look in your printer manual and, if the codes are wrong, change them.

You may wish to add other printer control codes to the program. As an example I will explain how to add an italics option.

First decide what key the user has to press to toggle italic printing. I used CTRL U for underline, but we can't use CTRL I because it is already being used. We will use CTRL T instead. We need a basic variable to say whether italic mode is currently on or off. We will call it ITAL. Edit line 60 and add :ITAL=0 immediately after UND=0. We can make any characters we wish appear on the screen when CTRL T is pressed. Underline uses a right arrow to show that it has been turned on and a left arrow to show it has been turned off. We can use character 194 for italics on and 195 for italics off. These are two diagonal lines sloping opposite ways. Now find out the code for CTRL T. The code for CTRL A is 1, CTRL B is 2, and so on. So CTRL T is 20. Now you can add the following lines to the program:

```
351 IF a=20 AND ital=0 THEN ital=1:a=194:POKE start+1,a:CALL
inschar:GOTO 250

352 IF a=20 AND ital=1 THEN ital=0:a=195:POKE start+1,a:CALL
inschar:GOTO 250
```

All you have to do now is to tell the print routine the control codes. Look them up in your printer manual and type in the following lines. The codes I have used should work on most printers.

```
671 IF ch=194 THEN PRINT#8,"    ";CHR$(27);CHR$(52);
672 IF ch=195 THEN PRINT #8,CHR$(27);CHR$(53);"    ";
```

You now have an italic printing option. You can add other control codes, if you wish, in the same way.

```
10 MEMORY 9999
20 n=10000
30 PRINT n;:INPUT"  ",a$
40 IF LEN(a$)/2<>INT(LEN(a$)/2) THEN P
RINT"***** RE-ENTER LAST LINE *****":G
OTO 30
50 b$=LEFT$(a$,2)
60 POKE n,VAL("&"+b$)
70 n=n+1
80 IF n=11728 GOTO 110
90 a$=RIGHT$(a$,LEN(a$)-2)
100 IF a$="" GOTO 30 ELSE 50
110 PRINT"Saving code as WORDCODE"
120 SAVE"wordcode",b,10000,1728
```

```
1 '          ******************
2 '          *                *
3 '          *   SUPER-WORD   *
4 '          *                *
5 '          ******************
6 '
7 '
8 '
10 start=10000:ef=0
20 init=start+422:inschar=start+436:re
draw=start+571:cursor=start+667
30 insline=start+1255:delline=start+13
29:insword=start+1387:justify=start+14
46
40 INK 0,0:BORDER 0:INK 1,26:MODE 2
50 scroll=8:ch=0
```

```
60 und=0
70 PRINT:INPUT"Size of left margin (in
 characters)";left
80 IF left<0 OR left>79 GOTO 70
90 PRINT:INPUT"Size of right margin (i
n characters)";right
100 left=left+1:right=80-right
110 IF right<left OR right>80 GOTO 90
120 IF HIMEM>start-1 THEN PRINT:PRINT"
LOADING CODE":MEMORY start-1:LOAD"word
code"
130 FOR n=start TO start+8:READ a:POKE
 n,a:NEXT
140 DATA 0.0.0,152,58,3.0.1,1
150 POKE start+2,scroll
160 POKE start+6.left:POKE start+9,lef
t:POKE start+10,right
170 POKE start+11,1+right-left
180 CLS:CALL init
190 CLS:CH=1:LOCATE left,3:PRINT CHR$(
24);" ";CHR$(24)
200 MOVE 0,372:DRAW 639,372
210 MOVE 0,27:DRAW 639,27
220 LOCATE 1,1:PRINT"CTRL H for help"
230 LOCATE 1,25:PRINT "COL";TAB(12);"R
OW";TAB(30);"TOTAL NUMBER OF LINES"
240 IF ch=1 THEN CALL redraw:ch=0
250 LOCATE 4,25:PRINT PEEK(start+6):LO
CATE 15,25:PRINT PEEK(start+7):LOCATE
51,25:PRINT PEEK(start+8);
260 a$=INKEY$:IF a$="" GOTO 260
270 a=ASC(a$):POKE start+1.a
280 IF PEEK(start+8)*(PEEK(start+11)+1
)>34000-start THEN LOCATE 65,1:PRINT C
HR$(7);"OUT OF MEMORY"
290 IF a=13 OR a>31 AND a<127 THEN CAL
L inschar:GOTO 250
```

```
300 IF a=127 OR a>239 AND a<252 THEN C
ALL cursor:GOTO 250
310 IF a=4 THEN CALL delline:ch=1:GOTO
 240
320 IF a=9 THEN CALL insline:ch=1:GOTO
 240
330 IF a=23 THEN CALL insword:GOTO 250
340 IF a=21 AND und=0 THEN und=1:a=243
:POKE start+1,a:CALL inschar:GOTO 250
350 IF a=21 AND und=1 THEN und=0:a=242
:POKE start+1,a:CALL inschar:GOTO 250
360 IF a=10 THEN CALL justify:ch=1:GOT
O 240
370 IF a<>12 GOTO 430
380 LOCATE 1,1:PRINT"* LOAD *";SPC(5);
390 INPUT"filename";f$
400 LOAD"!"+f$
410 FOR n=0 TO 11:POKE start+n,PEEK(14
988+n):NEXT
420 CLS:ch=1:GOTO 200
430 IF a<>19 GOTO 500
440 LOCATE 1,1:PRINT"* SAVE *";SPC(5);
450 INPUT"filename";f$
460 FOR n=0 TO 11:POKE 14988+n,PEEK(st
art+n):NEXT
470 a=PEEK(start+8):b=PEEK(start+11)+1
480 SAVE"!"+f$,b,14988,12+a*b
490 GOTO 540
500 IF a<>14 GOTO 550
510 LOCATE 1,1:PRINT"* NEW *";SPC(5);"
are you sure?"
520 a$=UPPER$(INKEY$):IF a$<>"Y" AND a
$<>"N" GOTO 520
530 IF a$="Y" THEN RUN
540 LOCATE 1,1:PRINT SPACE$(80):GOTO 2
20
550 IF a<>16 GOTO 760
```

```
560 LOCATE 1,1:PRINT SPACE$(3);"Double
 Line Spacing (Y/N)?"
570 a$=UPPER$(INKEY$):IF a$<>"Y" AND a
$<>"N" GOTO 570
580 IF a$="Y" THEN ef=1 ELSE ef=0
590 LOCATE 1,1:PRINT SPACE$(70)
600 LOCATE 1,1:PRINT SPACE$(3);"OK to
print?"
610 a$=UPPER$(INKEY$):IF a$<>"Y" AND a
$<>"N" GOTO 610
620 IF a$="N" THEN GOTO 540
630 LOCATE 1,1:PRINT"*  P R I N T I N
G  *"
640 a=PEEK(start+8)-1:b=PEEK(start+11)
-1
650 FOR n=0 TO a
660 PRINT #8,TAB(PEEK(start+9));
670 FOR m=0 TO b
680 ch=PEEK(15000+n*(b+2)+m)
690 IF ch<128 THEN PRINT #8,CHR$(ch);:
GOTO 720
700 IF ch=243 THEN PRINT #8," ";CHR$(2
7);CHR$(45);CHR$(1);
710 IF ch=242 THEN PRINT #8,CHR$(27);C
HR$(45);CHR$(0);" ";
720 NEXT
730 IF ef=1 THEN PRINT #8
740 PRINT #8:NEXT
750 GOTO 540
760 IF a<>8 GOTO 260
770 MODE 1:PRINT TAB(15);"HELP PAGE"
780 PRINT:PRINT:PRINT
790 PRINT TAB(3);"CTRL D - delete line
"
800 PRINT TAB(3);"CTRL H - display hel
p page"
810 PRINT TAB(3);"CTRL I - insert blan
```

```
k line"
820 PRINT TAB(3);"CTRL J - rejustify c
omplete file"
830 PRINT TAB(3);"CTRL L - load file"
840 PRINT TAB(3):"CTRL N - new file"
850 PRINT TAB(3);"CTRL P - print file"
860 PRINT TAB(3);"CTRL S - save file"
870 PRINT TAB(3);"CTRL U - underline t
oggle"
880 PRINT TAB(3);"CTRL W - insert word
(s)"
890 LOCATE 15,23:PRINT"Press ENTER"
900 a$=INKEY$:IF a$<>CHR$(13) GOTO 900
910 MODE 2:ch=1:GOTO 200
```

```
10000     00 00 00 00 00 00 00 00 00 00

10010     00 00 CD 9C BB CD 26 27 CD 9C

10020     BB C9 2A 15 27 CD 75 BB 2A 13

10030     27 7E CD 5A BB C9 E5 2A 15 27

10040     CD 75 BB E1 7E CD 5A BB 23 10

10050     F9 C9 2A 13 27 5D 54 1B 1B D5

10060     1A FE 20 28 2E 06 01 1B 1A FE

10070     20 28 09 04 3A 1B 27 B8 20 F3

10080     D1 C9 C5 E5 13 1A 77 3E 20 12

10090     23 13 10 F7 22 13 27 E1 C1 C5

10100     CD 34 27 C1 3A 16 27 80 32 16
```

```
1Ø11Ø       27 E1 E5 Ø6 ØØ 7E FE 2Ø 2Ø ØC

1Ø12Ø       Ø4 3A 1B 27 3D 3D B8 28 49 2B

1Ø13Ø.      18 EF 3E Ø2 32 1Ø 27 3A 1A 27

1Ø14Ø       9Ø 4F 5D 54 D5 F5 3A 19 27 B9

1Ø15Ø       28 5B 2B ØD 7E FE 2Ø 2Ø F3 3A

1Ø16Ø       19 27 B9 28 4E 2B ØD 7E FE 2Ø

1Ø17Ø       28 F3 F1 E1 E5 F5 91 47 5D 54

1Ø18Ø       13 7E 12 2B 1B 1Ø FA AF 32 1Ø

1Ø19Ø       27 3A 1A 27 47 F1 D1 13 3C B8

1Ø2ØØ       2Ø C6 AF 32 1Ø 27 E1 ED 5B 15

1Ø21Ø       27 D5 7B 3D 32 15 27 3A 19 27

1Ø22Ø       32 16 27 3A 1B 27 47 3D 5F 16

1Ø23Ø       ØØ AF ED 52 CD 34 27 D1 ED 53

1Ø24Ø       15 27 C9 F1 D1 4F 3A 1Ø 27 FE

1Ø25Ø       Ø2 28 CD 79 6B 62 18 8E 2E Ø2

1Ø26Ø       1E 16 3A 19 27 67 3A 1A 27 57

1Ø27Ø       3E ØØ 25 15 CD 5Ø BC C9 3A 17

1Ø28Ø       27 F5 3E 17 32 15 27 3A 12 27

1Ø29Ø       47 C5 E5 Ø6 Ø1 CD 12 28 E1 3A
```

```
1Ø3ØØ     1B 27 47 CD 34 27 23 3A 18 27

1Ø31Ø     47 3A 17 27 B8 28 Ø9 3C 32 17

1Ø32Ø     27 C1 1Ø DF 18 ØD C1 Ø5 28 Ø9

1Ø33Ø     C5 Ø6 Ø1 CD 12 28 C1 1Ø F7 F1

1Ø34Ø     32 17 27 3A 12 27 47 3E 18 9Ø

1Ø35Ø     32 15 27 C9 3A 17 27 F5 3E Ø3

1Ø36Ø     32 15 27 3A 12 27 47 C5 E5 Ø6

1Ø37Ø     ØØ CD 12 28 E1 3A 1B 27 F5 47

1Ø38Ø     CD 34 27 F1 87 3C 5F 16 ØØ AF

1Ø39Ø     ED 52 3A 17 27 3D 28 Ø8 32 17

1Ø4ØØ     27 C1 1Ø DB 18 Ø2 C1 Ø5 F1 32

1Ø41Ø     17 27 3A 12 27 C6 Ø2 9Ø 32 15

1Ø42Ø     27 C9 Ø1 A8 61 21 98 3A 11 99

1Ø43Ø     3A 36 2Ø ED BØ C9 CD 26 27 3A

1Ø44Ø     11 27 FE ØD 28 65 4F 3A 1Ø 27

1Ø45Ø     B7 28 ØE AF 32 1Ø 27 79 FE 2Ø

1Ø46Ø     28 51 C5 CD 44 27 C1 79 2A 13

1Ø47Ø     27 77 CD 26 27 3A 1A 27 47 3A

1Ø48Ø     16 27 B8 28 ØA 3C 32 16 27 23
```

```
10490      22 13 27 18 30 3E 01 32 10 27

10500      3A 18 27 21 17 27 BE 20 04 3C

10510      32 18 27 3A 19 27 32 16 27 21

10520      17 27 34 2A 13 27 23 23 22 13

10530      27 3A 15 27 3C 32 15 27 FE 18

10540      CC 26 28 CD 1C 27 C9 AF 32 10

10550      27 3A 16 27 47 3A 1A 27 90 5F

10560      16 00 2A 13 27 19 22 13 27 18

10570      B9 2A 13 27 3A 19 27 47 3A 16

10580      27 90 5F 16 00 AF ED 52 3A 1B

10590      27 3C 5F 16 00 3A 17 27 4F 3A

10600      15 27 D6 03 28 07 47 AF ED 52

10610      0D 10 FA ED 5B 15 27 D5 1E 03

10620      3A 19 27 57 ED 53 15 27 3A 1B

10630      27 47 16 15 C5 CD 34 27 23 1C

10640      7B 32 15 27 C1 3A 18 27 B9 28

10650      03 0C 18 03 21 F0 9B 15 20 E6

10660      D1 ED 53 15 27 18 84 AF 32 10

10670      27 CD 26 27 3A 11 27 FE 7F 20
```

```
1Ø68Ø      67 3A 16 27 21 19 27 BE 2Ø 34

1Ø69Ø      3A 17 27 3D 2Ø Ø8 3E Ø7 CD 5A

1Ø7ØØ      BB C3 2F 29 32 17 27 2A 13 27

1Ø71Ø      E5 3A 1B 27 3C 5F 16 ØØ AF ED

1Ø72Ø      52 3A 15 27 3D 32 15 27 FE Ø2

1Ø73Ø      CC 72 28 E1 2B 22 13 27 3A 1A

1Ø74Ø      27 3C 3D 32 16 27 47 3A 1A 27

1Ø75Ø      9Ø 3C F5 4F Ø6 ØØ 2A 13 27 5D

1Ø76Ø      54 1B ED 53 13 27 ED BØ F1 47

1Ø77Ø      2A 13 27 CD 34 27 C3 2F 29 FE

1Ø78Ø      FØ 2Ø 54 3A 17 27 3D 28 A3 32

1Ø79Ø      17 27 2A 13 27 3A 1B 27 3C 5F

1Ø8ØØ      16 ØØ AF ED 52 22 13 27 3A 15

1Ø81Ø      27 3D 32 15 27 FE Ø2 C2 2F 29

1Ø82Ø      3A 12 27 47 3A 16 27 4F C5 3E

1Ø83Ø      Ø1 32 12 27 3A 19 27 47 3A 16

1Ø84Ø      27 9Ø 5F 16 ØØ 78 32 16 27 AF

1Ø85Ø      ED 52 CD 72 28 C1 78 32 12 27

1Ø86Ø      79 32 16 27 C3 2F 29 FE F1 2Ø
```

```
10870      57 3A 17 27 21 18 27 BE CA C8

10880      29 3C 32 17 27 2A 13 27 3A 1B

10890      27 3C 5F 16 00 19 22 13 27 3A

10900      15 27 3C 32 15 27 FE 18 C2 2F

10910      29 3A 12 27 47 3A 16 27 4F C5

10920      3E 01 32 12 27 3A 19 27 47 3A

10930      16 27 90 5F 16 00 78 32 16 27

10940      AF ED 52 CD 26 28 C1 78 32 12

10950      27 79 32 16 27 C3 2F 29 FE F2

10960      20 4A 3A 16 27 21 19 27 BE 28

10970      0E 3D 32 16 27 2A 13 27 2B 22

10980      13 27 C3 2F 29 3A 17 27 3D CA

10990      C8 29 32 17 27 2A 13 27 2B 2B

11000      22 13 27 3A 15 27 3D 32 15 27

11010      FE 02 20 0D 23 3A 1B 27 5F 16

11020      00 AF ED 52 CD 72 28 3A 1A 27

11030      32 16 27 C3 2F 29 FE F3 20 42

11040      3A 16 27 21 1A 27 BE 28 0E 3C

11050      32 16 27 2A 13 27 23 22 13 27
```

```
11Ø6Ø     C3 2F 29 3A 17 27 21 18 27 BE

11Ø7Ø     CA C8 29 3C 32 17 27 2A 13 27

11Ø8Ø     23 23 22 13 27 3A 19 27 32 16

11Ø9Ø     27 3A 15 27 3C 32 15 27 FE 18

111ØØ     CC 26 28 C3 2F 29 FE F4 28 Ø4

1111Ø     FE F8 2Ø 19 3A 19 27 32 16 27

1112Ø     3E Ø1 32 17 27 3E Ø3 32 15 27

1113Ø     21 98 3A 22 13 27 C3 4B 29 FE

1114Ø     F5 28 Ø4 FE F9 2Ø 36 3A 19 27

1115Ø     32 16 27 3A 18 27 32 17 27 FE

1116Ø     15 FA 9E 2B 3E 15 3C 3C 32 15

1117Ø     27 3A 1B 27 3C 5F 16 ØØ 21 98

1118Ø     3A 22 13 27 3A 18 27 3D CA 4B

1119Ø     29 47 19 1Ø FD 22 13 27 C3 4B

112ØØ     29 FE F6 28 Ø4 FE FA 2Ø 1A 3A

1121Ø     16 27 47 3A 19 27 32 16 27 2A

1122Ø     13 27 B8 28 Ø4 2B Ø5 18 F9 22

1123Ø     13 27 C3 2F 29 3A 16 27 47 3A

1124Ø     1A 27 32 16 27 2A 13 27 B8 28
```

```
11250      EA 23 04 18 F9 CD 26 27 3A 19
11260      27 47 3A 16 27 90 5F 16 00 AF
11270      ED 52 E5 3A 17 27 47 3A 18 27
11280      3C 32 18 27 90 47 21 00 00 3A
11290      1B 27 3C 5F 16 00 19 10 FD 4D
11300      44 D1 19 2B E5 3A 1B 27 3C 5F
11310      16 00 19 EB E1 ED B8 23 3A 1B
11320      27 47 3E 20 77 23 10 FA C9 CD
11330      26 27 3A 19 27 47 3A 16 27 90
11340      5F 16 00 AF ED 52 3A 17 27 47
11350      3A 18 27 4F 90 28 D9 47 79 3D
11360      32 18 27 E5 21 00 00 3A 1B 27
11370      3C 5F 16 00 19 10 FD 4D 44 EB
11380      D1 19 ED B0 EB 18 BB CD F7 2B
11390      3A 16 27 21 19 27 96 CA 4B 29
11400      47 C5 2A 13 27 2B 3A 1B 27 3C
11410      5F 16 00 EB 19 E5 7E 12 3E 20
11420      77 2B 1B 10 F7 D1 C1 23 13 3A
11430      1B 27 90 47 1A 77 3E 20 12 23
```

```
11440        13 10 F7 C3 4B 29 CD 6A 2B 2A

11450        13 27 3A 1B 27 3D 5F 16 00 19

11460        3A 1A 27 4F 23 0C 3A 19 27 B9

11470        CA B6 2D 2B 0D 7E FE 20 28 F2

11480        3A 19 27 B9 28 34 2B 0D 7E FE

11490        20 20 F3 3A 19 27 B9 28 1A 2B

11500        0D 7E FE 20 20 E6 E5 5D 54 23

11510        3A 1A 27 3C 91 47 7E 12 23 13

11520        10 FA E1 18 E0 5D 54 23 3A 1B

11530        27 47 7E 12 23 13 10 FA 3A 17

11540        27 21 18 27 BE CA B6 2D 2A 13

11550        27 3A 1B 27 3C 5F 16 00 19 5D

11560        54 2B 2B 06 00 7E FE 20 20 04

11570        2B 04 18 F7 78 FE 00 28 7B FE

11580        01 28 6E 23 23 05 EB D5 3A 19

11590        27 4F 7E FE 20 20 0D 3A 1A 27

11600        B9 20 03 D1 18 60 23 0C 18 EE

11610        51 3A 1A 27 BA 28 09 23 14 7E

11620        FE 20 20 F3 15 2B 0D 7A 91 4F
```

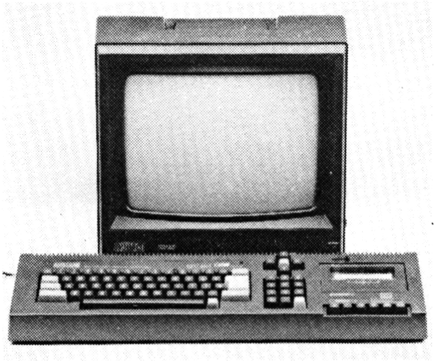
```
11630       D1 78 91 FA AD 2D 41 2B 10 FD

11640       23 41 7E 12 3E 20 77 23 13 10

11650       F7 2A 13 27 3A 1B 27 3C 5F 16

11660       00 19 3A 19 27 47 7E FE 20 20

11670       85 3A 1A 27 B8 28 04 23 04 18

11680       F1 CD 77 2A CD 41 2C CD 1F 2A

11690       C3 1C 2D CD 77 2A CD 44 27 C3

11700       B9 2C 3A 1B 27 47 2A 13 27 CD

11710       34 27 3A 17 27 21 18 27 BE CA

11720       2F 29 CD 77 2A C3 B9 2C
```

The Amstrad Range

Amstrad CPC 464

64K RAM. Integral cassette, Centronics printer port, expansion port. Maximum resolution 640 by 400 pixels, 3-channel sound. Complete with either black and white or colour monitor.

Prices in Jan 1986: £199 B&W, £299 Colour, but prices will be falling. Can be connected up to Amstrad 3-inch disc drive retailing for £159.95, second disc drive costing £99.99.



The Amstrad CPC 464 – built-in cassette, professional keyboard and, here, complete with colour monitor

The Amstrad CPC 664 with integral disc drive

Amstrad CPC 664

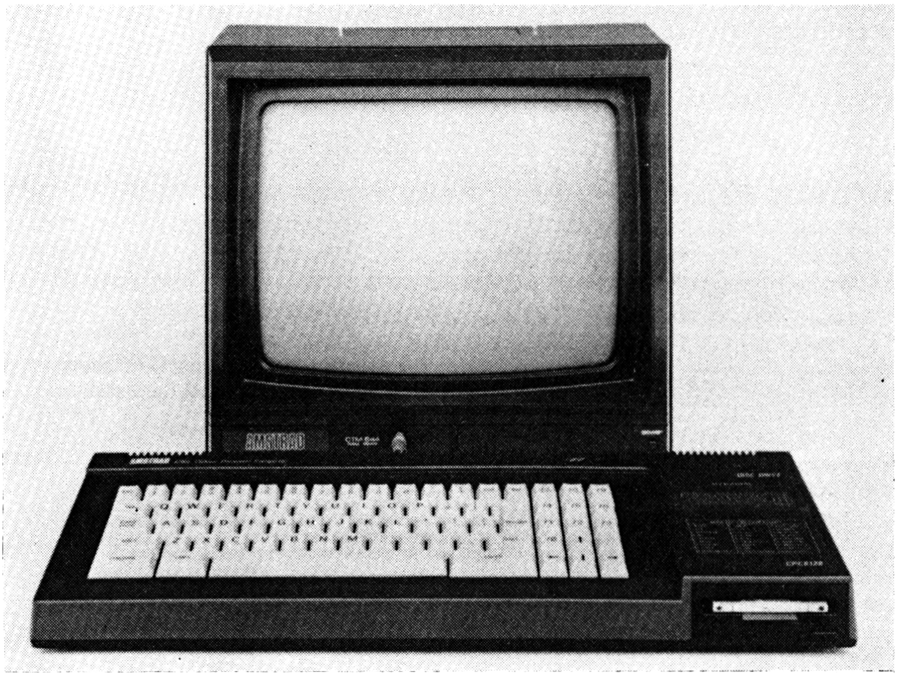
64K RAM. Biggest difference to the 464 is undoubtedly the integral disc drive replacing the cassette. The 3-inch disc handles a good range of software, essentially anything that works for the 464 on disc works on the 664.

Discontinued after a very short run on the British market. A very good second-hand buy as the model is still fully supported by Amstrad in guarantee terms and is indirectly supported by much software, both business and leisure, that is compatible with it.

Amstrad CPC 6128

128K RAM. Similar look and design to its predecessors but does feature some significant improvements. There's the same disc drive as in the 664, but with 128K memory. This machine can run full-blown CPM programs with the only modification being transfer of the program to a 3-inch disc. The 128K RAM is organised into the 64K normally accessible by the computer and a further 64K divided into banks. With extra commands, it is very easy to write one's own simple database program. The 64K has been utilised by software houses to provide improved versions of existing programs as well as completely new packages available exclusively on the 6128. Individual users will find that they benefit far more from the software houses' use of the extra memory than from their own efforts to use the spare RAM.

Latest price at publication date: £299 B&W, £399 Colour.



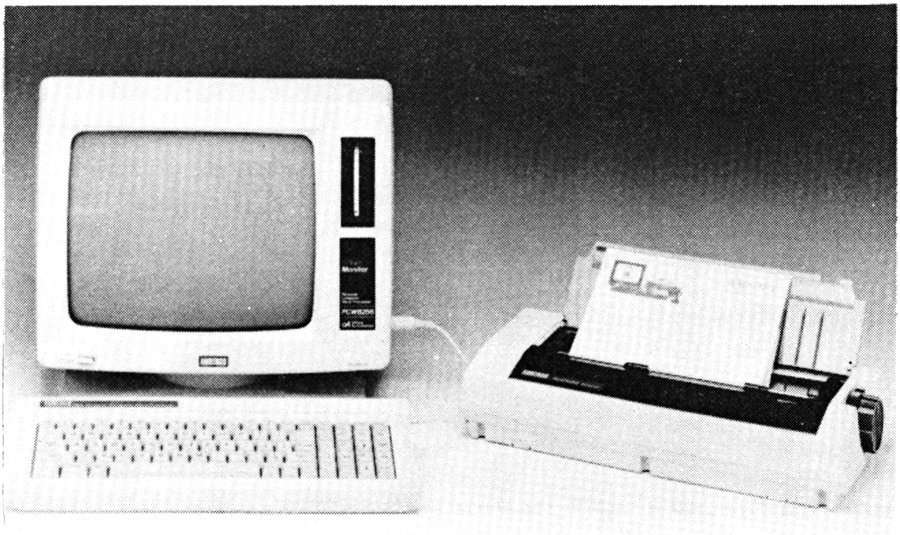
The Amstrad CPC 6128 – compact, sleek design coupled with CP/M Plus capabilities

Amstrad PCW 8256

256K RAM. Affectionately known as 'Joyce', this machine is a great departure from the previously accepted Amstrad models. It comes complete with monitor, disc drive built into the monitor, a professional keyboard and an NLQ (Near Letter Quality) printer — all for around £450! The 8256 is more of a dedicated word-processor than a microcomputer. It comes complete with a powerful word-processor, Locoscript, which utilises the printer, disc drive and vast memory very well. It is, however, a computer which just requires its language or languages loaded in from disc rather than finding them resident in its memory. Having CPM Plus capability makes the PCW 8256 capable of running many, many software packages already available, while independent software houses are gradually putting together a fair range of 8256-specific titles which play to the strengths of the new machine. Expansion is catered for with room for a one megabyte (a thousand K) disc as well as the capability to hook up to various interfaces.

The user receives Mallard BASIC, Dr Logo and the GSX Graphics Extension on disc with the machine.

The first three machines I dealt with all use a similar BASIC language with just a few differences here and there. Mallard BASIC, as supplied with *this* machine, is somewhat different as the following section will try to explain.



The Amstrad PCW 8256 word-processor/personal computer comes complete with built-in disc drive, keyboard, monitor, correspondence-quality printer and word-processing software

Mallard BASIC

This BASIC is very powerful and, while supplied with the PCW 8256 as standard, it is now available on disc for owners of the 6128 machine as well.

I do not intend to embark on a detailed discussion of the intricacies of Mallard BASIC, but merely to mention some of the major differences between Mallard and the standard Locomotive BASIC.

Firstly, it is a BASIC just like Locomotive BASIC — all the common commands work in the same way. Programs using IF/THEN, FOR/NEXT, READ, DATA, DIM, GOTO, GOSUB, PRINT and so on, will not be affected. Most of the functions such as ABS and CHR$ still work in the same way as well.

There are no graphics or sound commands in Mallard BASIC; these are, in part, provided by another package supplied with the PCW 8256, the GSX Graphics Extension System developed by Digital Research. Mallard BASIC is a language for the business or serious programmer and user.

Programs written in Locomotive BASIC will have to rid themselves of any sound or graphical pretensions. They will also have to slightly modify their file handling to disc routines, though this would be a simple matter with the Mallard BASIC manual in hand.

One major improvement over Locomotive BASIC is the set of commands belonging to a file access method named JETSAM. These allow for quick and speedy access to, and manipulation of, data files and will make many forms of data manipulation programs simpler to write as well as more effective.

The manual supplied with the language, certainly the 8256 version anyway, is very well written and includes some sample programs illustrating the power of Mallard BASIC. Some of the programs in this book will run without modification in Mallard BASIC, while others will need an hour's conversion.

To conclude, there follows a list of all the commands and functions available in Mallard.

Available Keywords

ABS
ASC
ADDKEY
ADDREC
BUFFERS
ATN
AUTO
CHAIN
CALL
CDBL
CLEAR
CHR$
CINT
CONSOLIDATE
CLOSE
COMMON
CREATE
CONT
COS
CVI
CSNG
CVD
CVUK
CVIK
CVS
DEF FN
DATA
DEC$
DEFSNG
DEF USR
DEFINT
DELETE
DEFDBL
DEFSTR
DIR
DELKEY
DIM
END
DISPLAY
EDIT
ERASE
EOF
ERA
ERL
ERR
ERROR
EXP
FETCHKEY$
FETCHRANK
FETCHREC
FIELD
FILES
FIND$
FIX
FOR
FRE
GET
GOSUB
GOTO
HEX$
HIMEM
IF
INKEY$
INP
INPUT
INPUT
INPUT$
INSTR
INT
KILL
LEFT$
LEN
LET
LINE INPUT
LIST
LOAD
LOC
LOF
LOG
LOG10
LOWER$
LPOS
LPRINT
LSET
MAX
MEMORY
MERGE
MID$
MIN
MKD$
MKI$
MKIK$
MKS$
MKUK$
NAME
NEW
NEXT
OCT$
ON
OPEN
OPTION
OSERR
OUT
PEEK
POKE
POS
PRINT
PUT
RANDOMIZE
READ
REM
REN
RENUM
RESET
RESTORE
RESUME
RETURN
RIGHT$
RND
ROUND
RSET
RUN
SAVE
SEEKKEY
SEEKNEXT
SEEKPREV
SEEKRANK
SEEKREC
SEEKSET
SGN
SIN
SPACE$
SPC
SQR
STOP
STR$
STRING$
STRIP$
SWAP
SYSTEM
TAB
TAN
TRON
TROFF
TYPE
UPPER$
UNT
USR
VAL
VARPTR
VERSION
WAIT
WEND
WHILE
WIDTH
WIDTH PRINT
WRITE
ZONE

Useful Addresses

Advantage
33 Malyns Close
Chinnor
Oxfordshire OX9 4EW
Tel: 0844 52075

Amstrad User Magazine
169 Kings Road
Brentwood
Essex CM14 4EF
Tel: 0277 230222

Arnor Ltd
The Studio
Ledbury Place
Croydon CR10 1ET
Tel: 01 688 6223

Campbell Systems
57 Traps Hill
Loughton
Essex IG10 1TD
Tel: 01 508 5058

Camsoft (Cambrian
Software Works)
Unit 2
Maenofferen
Blaenau Ffestiniog
Gwynedd
Tel: 0766 831878

Caxton Software Ltd
Lading House
10-14 Bedford Street
London WC2 9HE
Tel: 01 379 6502

Computing With
The Amstrad
Database Publications
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY
Tel: 061 456 8835

Cumana Ltd
Pines Trading Estate
Broad Street
Guildford GU3 3BH
Tel: 0483 503121

Datastar Systems UK
Unicom House
182 Royal College St
London NW1 9NN
Tel: 01 482 1711

Dialog Software
20 New Row
London WC2N 4LA
Tel: 01 502 4328/4423

Digital Research
(UK) Ltd
Unit 12
Fenton Way
Southfields
Basildon
Essex SS15 6SL
Tel: 0276 684959

DK'Tronics
Englands Lane
Gorleston
Great Yarmouth
Norfolk NR31 6BE
Tel: 0493 602926

Garwood Ltd
45 Plovers Mead
Wyatts Green
Essex CM15 OPS
Tel: 0277 823747

Gemini Marketing Ltd
Gemini House
Concorde Road
Exmouth
Devon EX8 4RS
Tel: 0395 265165

Micro Simplex Ltd
8 Charlotte Street
West
Macclesfield
Cheshire SK11 6EF
Tel: 0625 615000

New Star Software Ltd
45 Plovers Mead
Wyatts Green
Essex CM15 OPS
Tel: 0277 823747

Quest International
Computers Ltd
School Lane
Chandlers Ford
Hampshire SO5 3YY
Tel: 04215 66321

Sagesoft plc
NEI House
Regent Centre
Gosforth
Newcastle Upon Tyne
NE3 3DS
Tel: 091 284 7077

Tasman Software Ltd
Springfield House
Hyde Terrace
Leeds LS2 9LN
Tel: 0532 438301

Timatic Systems Ltd
Newgate Lane
Fareham
Hants PO14 1AN
Tel: 0329 239953

Wrexham Computer
Centre
24 Abbot Street
Wrexham
Clwyd LL11 1TA
Tel: 0978 358832

Glossary Of Specific Amstrad And Business Terms

Unlike many glossaries, this is not a general one containing just about every piece of 'computerese' that could be found. Only the jargon that will help you as an Amstrad owner or user interested in business computing has been included.

Algorithm: The sequence of steps used to solve a specific program

Alphanumeric: A term usually used to describe a keyboard which means that it consists of both numeric and alphabetical keys

Application software: Programs tailored to a specific task, such as accounting or forecasting profit margins

ASCII: Acronym for American Standard Code for Information Interchange. It consists of a numerical code, almost universally accepted, for each number, letter and other characters

Back-up: Copies of files made to protect against unintentional loss or file erasure

Baud: This is a measure of the speed of transfer of data

Bootstrap: This is a program, run into the computer when it powers up, which allows the computer to accept and understand other programs

Buffer: A storage mechanism which holds inputted data from any input device and then releases it at a rate which the computer decides

Bug: A term for an error in a program

Centronics: A type of interface allowing you to connect various peripherals, most commonly printers, to your computer

COBOL: Common Business Orientated Language. It is a standard programming language invented by Grace Hopper (who also coined the term 'bug'), which is very close to English

Control characters: An additional set of characters obtained by pressing the CONTROL key plus a letter

CP/M: This stands for Control Program/Microcomputer and is an almost universal disc operating system marketed by Digital Research

CRT: Cathode Ray Tube, a TV-like screen device

Cursor: A marker on the VDU screen that indicates where the next character will be displayed

Daisywheel: A type of printer so named for its printing element which is flower-shaped with a character at the end of each 'petal'

Data: A general term for information processed by the computer

Database: A collection of data, organised to permit rapid user access

Debug: To get rid of the bugs

Disc, diskette or disk: All mean the same and usually apply to a floppy disc. A storage medium for data and programs, discs come in 8-inch, 5¼-inch, 3½-inch and 3-inch sizes

Documentation: The written instructions and explanations which come with a program

DOS: Short for Disk Operating System, it is a clever program which allows the computer control of the disc system

Dot matrix printer: A printer which forms its characters from a collection of dots formed into a grid

Double-density: Discs that offer this have double the storage space of one disc

Field: A collection of characters forming a distinct group. Used most often in databases (see **Record)**

File: A group of related records which are processed together, for example, Payroll File

Firmware: The solid components of a system (the keyboard and so on) are known as the **hardware,** the programs to be typed or loaded into the computer are called the **software** while the programs that are actually hard-wired into a circuit in the computer are called the **firmware**

GIGO: Garbage In Garbage Out is the saying, and very true it is too. Always check that the data being inputted is correct at all times

Hard copy: A printed or typed copy of information held by the computer

Hardware: (See **firmware**)

I/O: Input/output functions, as discussed in the first chapter

Interface: In computer terms, this is some form of physical connection between a computer and another item, be it a printer, a disc drive or another computer

Justified copy: Copy with an even right-hand margin, usually achieved by padding out a line with extra spaces

Kilobyte or K: The standard measurement of computer memory. Any business machine worth its salt has at least 64K

Menu: A list of alternative user actions supplied by a program which can be selected from by the user

Modem: A device which enables one computer or terminal to communicate with another via a telephone link

Mouse: This is an input device the size of a cassette case which is rolled over a desk or flat surface moving a cursor on the VDU screen. This can be manipulated to select options and make decisions with appropriate software and is a simpler device to use than, say, a keyboard

Network: A group of computers linked in some way and working together

Numeric Pad: A keypad used for entering numeric information, similar to the arrangement of keys on a calculator

OCR: Optical Character Reader. A fascinating device that can scan the pages of a book or other text and translate the images it sees into digital information that can be fed into a computer

Parameter: Some form of specification, usually numeric

Peripheral: This term applies to any hardware device that is an accessory to the computer and is connected up to that computer

Program: This is a series of instructions which the computer follows to carry out a pre-determined task

Ragged-right: The usual typewriter style of printed output with the right margin uneven

Reset: This is a control function which returns the computer to the point it was at when it was first turned on

RS232: This, like Centronics, is a popular interface for use with microcomputers

Scrolling: The movement of text up and down the screen to allow text which does not fit on the screen to be viewed and edited

Software: (See **firmware)**

Syntax: The structure rules which govern the use of a computer language or an applications package's system of commands

Terminal: A device, usually with a keyboard and a screen, for input and output of data, but without any actual calculating power

Thermal Printer: Printer which uses heat-sensitive paper

Time-sharing: The sharing of a large computer facility among many users, each equipped with a terminal

Turnkey System: A computer system that is ready to run when delivered, needing only the metaphorical 'turn of the key' (probably a special access code to be typed in) before it can work

VDU: Visual Display Unit. Anything from a high quality monitor to a humble black and white TV constitutes a VDU when used with a computer

Word-processor: A dedicated computer (or program package) with intelligent typewriter facilities including such features as correction, multiple copies and so on

BIBLIOGRAPHY

There are really very few books that are of use to the Amstrad owner wishing to use his machine to aid his business. The two that I've found and used are noted below:

How To Computerise Your Business Successfully
D.Davey/R.Womack, Interface Publications
ISBN: 0 7248 1134 6
This is the most useful guide that I have found on this subject. It consists of a series of essays by experts in the field of business computerisation. The small businessman may find a number of the chapters too large-scale for his use, but it makes interesting reading nonetheless.

How To Profit From Your Computer
T.G.Lewis, Hayden Press
ISBN: 0 8104 5761 X
Readers prepared to put up with the quirky style and Americanisms of this offering from across the Atlantic will gain much useful knowledge. Look around for this book in libraries and in book sales — my copy cost me just £1.

DOING BUSINESS WITH YOUR

AMSTRAD

An increasing number of businesses, .large and small, are finding that a computer is a necessary investment as the drive towards greater technological efficiency continues. But with the machine installed, where do you go from there? What can it do? Will it save a lot of time? What software should you buy? This book answers such queries ... and many more.

As well as information and advice on using your Amstrad for business purposes, the book also contains a number of ready-to-run business programs that will work on both cassette- and disk-based systems.

Written by an experienced computer journalist who has already been through many of the problems you may face and who has drawn on the experiences of business software programmers and users, this is an essential and comprehensive guide to DOING BUSINESS WITH YOUR AMSTRAD.

ISBN 0 86369 146 3
£4.99

00499
9 780863 691461

DOING BUSINESS WITH YOUR AMSTRAD CLIVE GIFFORD

Virgin

AMSTRAD CPC

MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA

https://acpc.me/