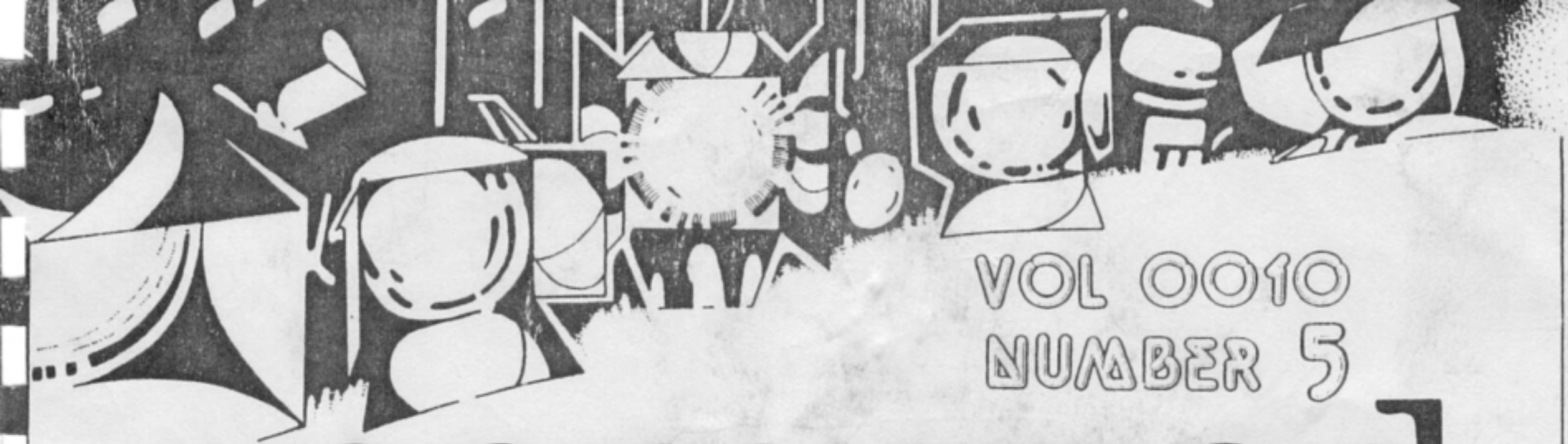


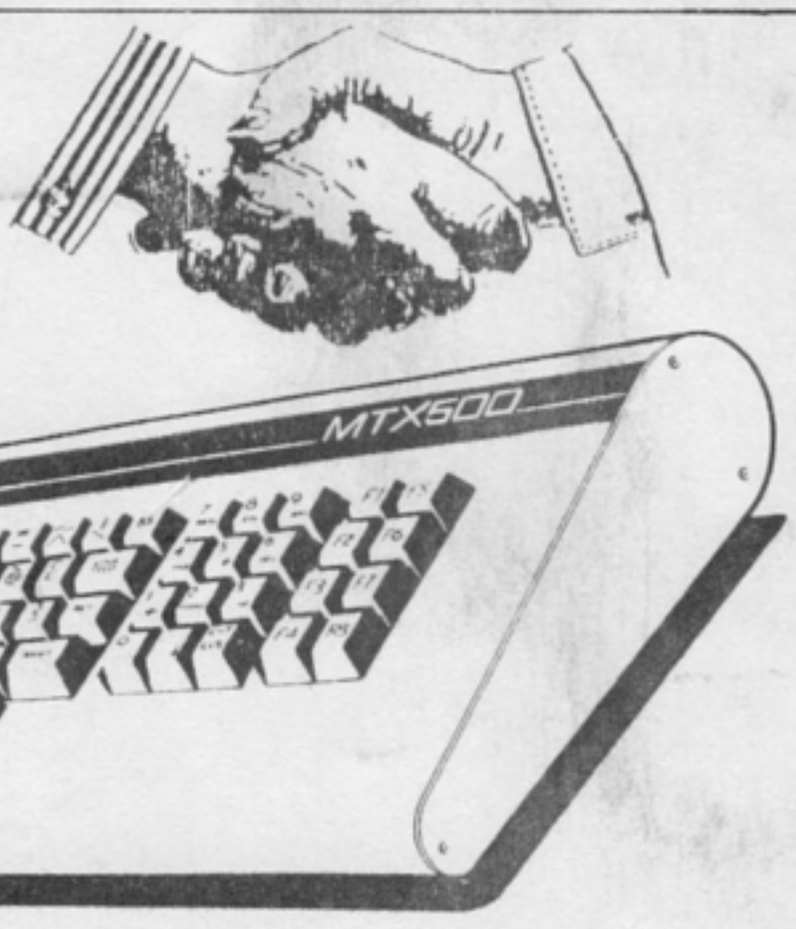
VOL 0010
NUMBER 5

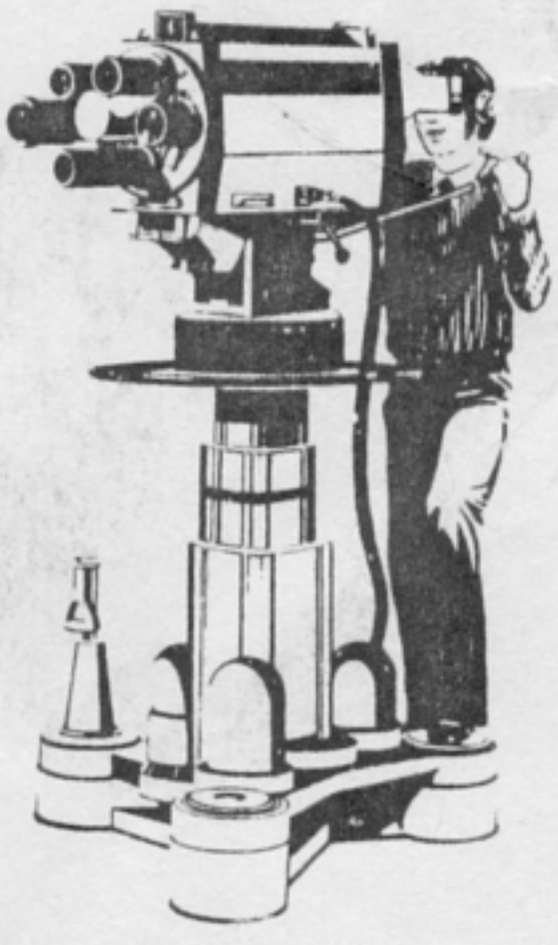
memopad

Memotech Computer User Club Magazine

Contents

EDITORIAL	PAGE 129
HIGHSCORES	PAGE 130
REVIEW..BOOKS	PAGE 131
HISTORY OF COMPUTING	PAGE 131
VIEWPOINT	PAGE 136
FLASH..RAINBOW	PAGE 140
ASSEMBLY LINE	PAGE 141
DISC ADAPTION FOR EDASM ...	PAGE 143
DEBOUNCE THOSE KEYS	PAGE 145
SOUND EDITOR	PAGE 146
CLASH OF THE ROBOTS	PAGE 147
BOOKS..2 NEW UTILITIES	PAGE 150
SHARED BANK ACCOUNT	PAGE 151
HARDWARE	PAGE 157
SOFTWARE	PAGE 158





£1.25

Editorial

THE Z80 IS DEAD! I remember reading this in some obscure magazine about three years ago. Judging from the letters you have written, on the subject of a new computer, you must have read the same article.

It is apparent that Microsoft never read the article before they chose the "dead chip" as the backbone of their new standard, MSX. Go on, laugh. Even so, MSX has sold over one million machines throughout the world - not a bad achievement for a machine with out-dated technology.

If Alan Sugar, head of Amstrad, had read this article then he, or his advisors must have been raving lunatics to have used this same out of date technology in their CPC 464, and subsequent machines. Surely they must have realised they had no chance of their product ever becoming a best seller !

MSX 2 is now poised for an invasion of these shores. In fact, an advance party has already arrived. Its critics will say that it has no chance of success because Microsoft have, once again, robbed the grave and are using the Z80 as the heart of the new machine. However, scoffers be warned, this is no ordinary machine.

Microsoft have re-designed the old Texas 9918 video chip and have built the V/9938 chip. This new chip is software compatible with the older chip. The V/9938 boast ten graphic resolutions with the highest resolution at 512 x 424 pixels. Eight sprites on a line are allowed and these can be multi-coloured. The new chip can also manage a video ram of 128K and has a palette of 256 colours. Hardware screen scrolls in any direction are supported along with raster scan interrupts chosen by the programmer. External video cameras can be attached and light pens are also catered for.

Rumour has it that Tatung are also using this chip in their new Z80 based 'Amstrad Killer' due to be released later this year.

Memotech have the guts of a superior machine. It would be wise of them, and us, to concentrate on improving the existing hardware, upgrading its possibilities, and giving the user a new rom based Basic instead of starting from scratch and making the mistakes of other manufacturers e.g. Commodore et al. Development of a new system costs a lot of money - "BIG MONEY !" (sorry Max), and I am sure Memotech do not have that kind of financial backing. So. Let's stay alive ! Just like the good old Z80.

Disc owners will be pleased to know that two very nice utilities are about to be released EDASM and SDX utilities....they are mentioned later in the magazine. A unique disassembler is now available for SDX & tape based systems. It not only allows you to disassemble the rom but it also disassembles RST28 & RST 10 calls with MTX notes attached.

Many thanks for the kind letters I have received since I took over, they are appreciated.

Keep on tapping.

TM

GENPAT HIT LIST....

arcade

1.	SEPULCRI SCELERATI	MEGASTAR
2.	SUPA CODER	SYNTAXsoft
3.	USER BASIC	SYNTAXsoft
4.	MANIC MINER	SOFTWARE PROJECTS
5.	ROLLA BEARING	MEGASTAR
6.	JET SET WILLY	SOFTWARE PROJECTS
7.	SMG	MEGASTAR
8.	ESCAPE FROM ZARKOS	MEGASTAR
9.	MEMOSKETCH	SYNTAXsoft
10.	DOWNSTREAM DANGER	MEGASTAR

adventure

1.	SNOWBALL	LEVEL 9
2.	LORDS OF TIME	LEVEL 9
3.	EMERALD ISLE	LEVEL 9
4.	THE KEYS TO TIME	SENTIENT
5.	MURDER AT THE MANOR	SENTIENT

educational

1.	HELLI-MATHS	SENTIENT
2.	MATHS 1	CONTINENTAL
3.	WORDS & PICTURES	CONTINENTAL
4.	SPELLI-COPTER	SENTIENT
5.	FIRST WORDS	CONTINENTAL

High Scores

Can you do better ?

ASTRO-PAC	213,430	Michael Hunt
KNUCKLES	1,147,360	T. Eriksson
CHAMBERBIDS	Completed 4 mins	T. Eriksson
MISSION ALPHATRON	74,240	Gordon Hurd
TAPEWORM	175,980	Richard Franks
TOADO	178,292	Gavin Gaunt and Nicholas Locke
POT HOLE PETE	105,630	Richard Franks
MAXIMA	1,479,710	S. Olander
STAR COMMAND	140,430	Ian Nichols
PHAID	26,000	Sally Street
ORLOIOS	62,400+	M. Hurley
KILOMEDE	82,253	Richard Nash
3D TACHYON FIGHTER	10,700	Lesa Woodger
CONTINENTAL RAIDERS	106,240	Sean Haverty
BLOOMO	148,203	Elizabeth Mahon
QUANTUM	14	Ian Cartwright
QOGO 2	205,000	R. Siddall
MINEFIELD	1,500	David Nash
FLUMPOX	239,550	Andrew Miller
TURBO	18,610	Michael Hunt
FATHOMS DEEP	2,300	T. Eriksson
AGROVATOR	875,000+	P. Howard
FIREHOUSE FREDDIE	29,620	T.Eriksson
QOGO	43,960	T. Eriksson
ARCADIANS	25,900	Adrian Joynson
MISSILE COMMAND	27,580	Adrian Joynson
LITTLE DEVILS	34,320	Leslie Hawks
FELIX IN THE FACTORY	14,740	Peter Crighton
HUNCHY	7,900	R.Harmer
SON OF PETE	17,233	T. Eriksson
HAWKWARS	16,600	T. Eriksson
ESCAPE FROM ZARCOS	46 Items	P+G Howard
SALTY SAM	40,642	Andrew Johnson
MISSION OMEGA	9,350	R.Harmer
ICEBURG	17,431	Alan Dobson
SNOWBALL	450	P. Crighton
EMERALD ISLE	300/1000	Richard Franks
SUPERBIKE	23.9Kms	B. Clark
DOODLEBUG	3,820	Andrew Miller
DR. FRANKIE	65,435	J Graham
TARGET ZONE	14,560	P. Howard
MINER DICK	22,520	R. Siddall
JUMPING JACK	26,120	Andrew Miller
SURFACE SCANNER	72,060	T. Eriksson
CAVES OF ORB	421/500	P. Koraten
SEPULCRI SCELERATI	7,925	Andrew Miller GAME COMPLETED
S.M.G.	6,140	Andrew Miller
COMBAT	42,310	Andrew Miller
QUAZZIA	26,660	Andrew Miller
OBLITERATION ZONE	32,670	Alan Dobson
ASTROMILON	3,680	Alan Dobson
CRYSTAL	1,033	Alan Dobson
DRIVE THE CEC-S	7,255	Alan Dobson
THE WALL	39,720	T. Eriksson

Mike Nash has completed Qogo 2 and has quoted the final message - "At last, you have found the Oxyan diamonds"

Can you beat these high scores ? Do you have a high score for a game not mentioned above ?

REVIEWS *Books*

Two books that appeared on my desk recently are SECRETS OF BETTER BASIC and HOW TO BUILD A PROGRAMME. Both of them are aimed at the basic programmer and the latter is a must for any serious computer buff.

HOW TO BUILD A PROGRAMME
Jack Emmerichs
DILITHIUM PRESS

This book is on how to develop computer programmes. It is aimed at the person who has little or no experience in computing. The procedure of development is shown using TOP-DOWN structured programming.

The blurb on the book says 'here is your chance to look over an experienced programmer's shoulder. This book shows how to develop an original idea into a set of instructions.....Included are errors and bugs which you will undoubtedly run into.....'

The techniques employed are demonstrated in Basic and Pascal so the book is also valuable to the novice Pascal programmer.

The book opens by explaining how to define a programme and gives brief descriptions of the main components required in an average programme. The card game BLACK JACK is the actual example used to demonstrate the techniques employed by the author to develop a programme.

I found this book very interesting, easy to read, and very well written. I would say this is a must for anyone who takes their hobby seriously. The examples are written in standard Basic and are easily implemented on the MTX.

Included within the book are:
Defining a programme
Gathering different types of instructions
Structure of the statements
How data types are used
Example computer programmes for real application
The start of a programmer's project
Breaking functions into smaller units
Developing the data dictionary
Developing the programme instructions
Creating the programme
Testing the final programme
Identifying the changes to be made

SECRETS OF BETTER BASIC
Ernest E Mau
Hayden

In this book the author reveals some of the tricks used by software writers for creating Basic programmes that run faster.

Some amazing techniques are discussed on testing small modules, timing their execution, and then altering their structure and re-testing. In some cases, small alterations to the programme cause the resulting code to run 20% faster.

At each juncture the author gives examples along with execution times. This allows the reader to see the difference in running times between the first draft and the final programme.

Standard Basic is explained fully, some of the instructions are relevant for your computer, some are not. No matter, the book gives a good grounding should you decide to exchange computers at a later date.

One minor drawback with the book is that the points discussed are demonstrated using Applesoft and Microsoft Basic. This is only a minor point as the programmes are easily implemented using any Basic and the improvements can easily be tested using the methods described in the book.

This is an excellent reference source that can teach you a great deal about how to improve your programmes and how to speed up execution time.

DATA BASE MANAGEMENT SYSTEMS
Karl Townsend
Dilithium Press

This book is a must for anyone using or about to use CPM for the first time. It is not a text book, it is a comparison guide of the different data base systems available.

Most users require a data base and in the majority of cases each user will have different requirements. Karl Townsend takes an objective look at the many data bases available. He also will help you to understand how management systems can improve your way of dealing with information.

Some of the systems covered are:

D Base
KSAM 80
Prism
Condor
Datastar
Quest

One startling fact came to light after reading this book:

A 1260 item inventory file sorted in two and a half minutes using Condor. On D Base 2 it took twenty-two and a half minutes! ★

The History of Computing (I think!) Part 1

P Knaggs

By the word COMPUTER we do not necessarily mean a machine that can calculate values. In fact the word is an English word not used much until these 'Machine' things came around. However, the word really means something like: the action of counting or calculating a mathematical expression, even if it is 1 plus 1 .

So when I say the history of computing I really mean the history of numerical systems and counting.

O.K. less of the definitions and on with the story of the HISTORY OF COMPUTING AIDS:

Pre-Historic man, being a human like the rest of us, needed a way of boasting to his peers about what a great hunter he is, and how many deer he had killed recently. To do this he used a digit (a finger, thumb, or even a toe) to represent one deer. By raising 1 digit, meaning 1 deer, 2 digits (fingers) meaning 2 deers and so on up to 10 digits (8 fingers and 2 thumbs) where upon he ran out of digits. He, thereby invented a number system that we have kept ever since and is one of the fundamentals of any calculations, that is the concept of a BASE, in this instance base 10.

Unfortunately, the method I have just described has two limitations:
a. It is impossible to represent no digits (the value of ZERO)
b. The elder members of the village found it hard to represent their boasts as it was many times that of the number of digits available to them. (This system can only go up to 10 digits). They were not happy about this at all.

So the elders set someone to work about finding an answer to this problem. In the fullness of time, this young whipper-snapper came up with the idea of making marks in the sand or dirt. For straight lines (one line, to one digit), and then a fifth to group the lot together. This method was chosen because one group can represent one hand (or 5 digits). Although this was e called it (a modest man was our Blase) was an adding machine, where you turned a dial (marked out rather like a telephone dial) the correct amount for the number you wish to add, and then the machine added it to the number it s digits so a great pile of stones equals a great number of deer. Whilst this did enable you to move your numbers around it did have some problems of its own.

a. Some of the elders cheated the system by placing large stones on the bottom to make the pile look larger than it should be.
b. After a while again the stones got to be a bit heavy and cumbersome to carry around in skin bags.

This poor young man was sent away, again, to think of something. This time he came up with the idea of cutting notches out of a lump of wood. One notch, one digit.

Well this was phenominal, wood was easy to notch, plenty of it around, could be carried easily. In fact this method of counting was used for many centuries, and looked like going on some more if it weren't for these rather dominate people called the ROMANS.

Well, the Romans being such forceful people, went and designed their own values for representing the digits. These are known as "Roman Numerals". Unfortunately, whilst being easier to engrave into material (such as stone), these numerals tended to get rather long for their own good. It was also very impracticable to calculate with them (i.e. the value of 4 would be IV or 1 digit (I) before 5 (V), this as you can imagine can be quite complex for a number like 1985) . Now try to calculate IV times III. The answer is in fact XII, not an easy calculation you will agree, especially when it comes to the large numbers. This gave rise to another way of writing numbers down. This time invented by the Arabs, but pinched by the Greeks, who really wanted numbers for way-out calculations of all types. To show how they represented numbers is not difficult to explain as we use them during our every day uses. I have used the Arabic representation of numbers throughout this essay. It was them who invented the symbols , 1 for one, 2 for two, 3 for three,and so on up to 9 for nine.Then things got interesting. Indeed a major step in the history of the calculator had already taken place (the writing down of numbers) and more were about to, and all of them by the Greeks.

One of these innovations was the adventation of the Abacus, which used (and still does, for it is still in use today, by some people) several rows of beads to represent digits. The next row down represents 1 full row of the one above. Thus inventing a real base 10 number system For the abacus to work to this method, involves another quite theological alteration of the way that they thought of numbers in those days (mind you, the Greeks were hot on theological revelations and that kind of thing). This is the idea of ZERO, a digital representation of nothing. It was the abacus that pointed out to people that there was a use for such a value, and so it was invented. This then now allows us to represent 10 as 1 set of ten digits and no single digits. (By the way, this is the other innovation, if you hadn't sensed).

(I.e. the number 6546 would mean:

6 sets of 10 sets of 10 sets of 10 single digits	(6000)
5 sets of 10 sets of 10 single digits	(500)
4 sets of 10 single digits	(40)
6 single digits	(6)
Total:	(6546)

Now as people got more used to this system they managed to do more and more calcultions in their head. However, they still used the old abacus for the longer sums (such as 7 digit calculations - remember 1 digit can now represent 0 through to 9 different values).

At this point an English man by the name of John Napier (well he was Scottish, that's close enough for me) came on to the scene . He, through some method workd out that every number had a contra Logarithm. As we all know, logarithms can be used for all kinds of long, complelx calculations that involve multiplication, division, raising to the power, rooting, ad infinitum.

Because of their usefulness, people went and made a fortune making copies of them for the computers (or mathematicians). In fact the well known slide rule, works for logarithms. This little device killed off the tables approach amongst the mathematicians.

In fact the slide rule has only recently been replaced by more modern calculating machines. Indeed, a great many of them are still in use, and every school boy should have one and know how to use it.

Next, in order of history, came the PASCLINE (invented around mid 1640's) which was invented by a French son of a Tax Collector by the name of Blase PASCAL. The Pascline, as he called it (a modest man was our Blase) was an adding machine, where you turned a dial (marked out rather like a telephone dial) the correct amount for the number you wish to add, and then the machine added it to the number it already had. This saved you having to keep remembering or writing down the number, as the machine did it for you. Unfortunately, to subtract a number you had to go inside the machine and alter something. Not a good idea for a commercial adding machine. The PASCLINE was not taken up by business who could not see the advantage over their Accountants using slide rules.

Later on PASCAL teamed up with a Gottfied Leibnitz and Leibnitz is credited with re-designing the PASCLINE to make it multiply and divide as well, (and thus the first true calculating machine was born) but this was some 50 years after the first PASCLINE was manufactured.

So after the flop of the PASCLINE no one wanted to know about this new one and all the mathematicians had continued using the slide rules anyway for another couple of centuries (a rather quiet time in human history all round I understand).

So enter (stage left) one Lady Ada Lovelace. Now Ada was a prominent mathematician who got tangled up with a person who is now termed as the father of moder computing machines. It is only through Ada's notes that we have any inclination of how these early contraptions worked. The not-so-unsung genious that she worked for was one Charles Babbage. Babbage designed this monster of a machine that he called a "Difference Engine". This little thing was a machine that used the difference method of mathematics to generate polynomial expressions. Quite why anyone wanted to do this I don't know, but there you are. His machine also used a number base known as BINARY (that is a system that only has 2 digits, a 0 or a 1). The reason for this was quite simple. It is extremely difficult to design machinery to work in DENERY (with 10 different states or digits), it is much simpler to design something which can be said to be On or Off (On is 1 , and Off is 0). Okay, this system does take some getting used to but it does make design and manufacturing much simpler.

Babbage was fool enough to show his contraption to the Government of the day. After a lot of bickering about prices, they finally managed to commission a load of these things from our Mr. Babbage. (Records show that this was in 1822).

After eleven years of work on mass-production of the parts for his engine, Babbage had to finally give in saying that the current manufacturing technology could not give the accuracy of the parts to make the damned thing work in the first place, or something like that. After having a rest for a bit, he came back to force by designing another such device, but much more complex. This one he called his "ANALITIC ENGINE", unfortunately he could only design it, as again the parts to make it work simply could not be manufactured at that point in time. He never lived to see if it worked or not, however, part of it have since been made and can now be found in London's Science Museum. (By the way, yes they do work).

Now what was so amazing about this little device is that it could store (remember) a list of instructions as opposed to his DIFFERENCE Engine which simply got on with the job it was designed for. All Babbage had to do was tell it what to do with the numbers that he will give it in a moment. When this was done he then gave it a number which it manipulated in accordance with his instructions and then a result was produced at the other end. When he received the result , he could then put in another number. So it consisted of some form of Input device, a Processing operation, and an Output device. It also had to remember the programme that involves some form of memory.

To tell this device of his all of these things, he was using a device called a Jacquard card, which had been designed by a Mr. Jacquard to control his weaving looms (at the turn of the nineteenth century). These cards were simply a series of holes or not holes (back to BINARY, a hole for 1, and no hole for 0) where a bar came through to push the thread of the weave. There were several of these cards strung together to make up a repeated pattern (the last in the sequence was attached to the first in a never ending loop).

The reason that we call Babbage the Father of modern computing is due to his theologies.

I He found that if a machine like this were to work then it would have to remember what it had just done or what to do with things. A list of instructions. Without these instructions such a machine would be useless.

II He saw the potential of using the binary system, which for all intents and purposes is just as adequate as the Denary system. With this, is that he also saw the potential of the Jacquard card to store the binary instructions and data, and further simplicity of the bits and pieces that go to make up one of these calculating machines.

III He saw that a computing machine would need to be broken down into three sections. That is the sections of:

Input	-	Give it the instructions and the data
Processing	-	Manipulate the data in accordance to the instructions
Output	-	Tell the operating human what it came up with

All of the above concepts have been used in computing machines ever since and will probably continue to be.★

FOR SALE
The following at 50% of GENPAT list price + postage
1 Newword ROM
2 BRIDGE 3 PONTOON/BLACKJACK 4 DRAUGHTS 5 CHESS 6 REVERSI
7 BACKGAMMON 8 KNUCKLES 9 BLOBBO 10 OBLOIDS 11 SNAPPO
12 STAR COMMAND 13 3D TACHYON FIGHTER 14 KILOPEDE
15 MINEFIELD 16 THE ZOO 17 HELI-MATHS 18 SPELLI-COPTER
19 FIRST LETTERS 20 WORDS AND PICTURES 21 MATHS 1 22 PHAID
23 TURBO 24 PHYSICS 1 25 BASIC BUSINESS 26 UTILITIES
27 GOLDMINE 28 FELIX 29 POTHOLE PETE 30 MEMOSKETCH
31 COMPOSER 32 OBLITERATION ZONE 33 TOADO 34 NEMO

Would also be interested in swapping for MTX hard/software especially pascal ROM, disc FORTH or RAM for 512.

Contact : David Jones, Kingham (060871) 8089

View Letters TO THE EDITOR Point

93 Campbell Ave,Stevenston,Ayrshire, KA20 4BT.

I see that Memotech have decided to advertise in the December issues of some of the popular magazines, not before time. Maybe this publicity drive will sell lots more MTX's and thus spur the software writers into writing some good software for the MTX. The main areas of software which are badly represented on the standard MTX are the business and scientific packages and the lack of good if any educational software. If the MTX is to make a big impact on schools then it'll need to get some good educational packages and also since many educational centres are now using the language Comal (i.e. in Scotland, Iceland, Norway, etc.) as the programming language, even the BBC B has now got a comal package, so Memotech could do no wrong in commissioning such a package and put it onto a Rom board and sell the MTX and this package for the schools. This would be an advantage for the MTX.

Another area the Memotech could do with improving is the graphics and have a pipe or tube where it could add a second co-processor like the powerful 68008 cpu. If this port was available then they could commission Cumana to write the interfacing software for its excellent OS-9/68000 upgrade package which it offers for the BBC and Q1 computers. The package offers a 68008 cpu (for BBC only), 512K ram, Winchester disk and floppy disk interface controllers, clock, rom expansion 144K (for QL only) and software:- Basic OS-9,c,Pascal,compilers, assembler, graphics and word processors, all for #700. This operating system will give Unix compatibility with any compiled languages like C. In my view this is a far more powerful package than the boring and out of date CP/M facilities.

The MTX has been out for about 2 years and for the last 1-5 years the only thing to emerge from Memotech was the sdx disk add-on. I am sure that Memotech have been working toward a new machine or some kind of pipe to allow another processor. The MTX is an excellent designed computer and offers many outstanding features which make it the best Z80 based computer. A decent 80 column board and graphics board (offering pixel resolution and full addressable pixel colour) with a pipe interface would make the MTX very appealing to the more serious users who are looking for a decent 16-bit micro - see the last few issues of memopad. The Cumana package mentioned above is outstanding value and if the MTX could get in there then the MTX, BBC B and QL would share the same OS-9 operating system and set the standard in Britain. This would lead to greater software support and secure the MTX's future.

Hope all at Genpat have a happy Christmas and a good and successful New Year.
ALAN F. WILSON (C1203)

P.S. What's Genpat's views on the OS-9 package and the likelihood of Memotech making a 'pipe' or tube interface. Also what has Memotech been up to in the last year and a half on the technical side.

Peter Fowles has the following comments to make on "Where do we go from here?"

I have read a lot of letters with suggestions on this point, and I think a lot of them are missing the point. The question "where do we go?" is irrelevant - we are already there.

The Memotech is, and remains a very good system (one of the best I feel) and has the potential to go far and last long, so any ideas of drastic modification and redesign are idiotic and pointless (akin to improving perfection).

The question is not "where do we go?" but "what can we do?" So, what do we have - Hardware and Software.

Hardware
Well as I have said I believe that the MTX is well designed and its hardware up to any standard you may set; when people talk about hardware they usually mean expansion potential - which I think the MTX has - or has it?

At the moment I own a MTX 500 which I am about to begin to expand, but at this time know very little of the hardware (Expansion) characteristics of the MTX, as Mr Hodgson (Vol.2 No 4) said, "I know how a disc is sectored what I don't know is how difficult is it to put tape software onto disc or how much work is required to get from a single 250K disc to a CPM compatible system or to be on par with the 500K FDX system." (Could you please answer these questions). This type of information, when made available, can sell a machine, and also the number of games available for the computer. This brings me very nicely to.....

Software
The software of a machine can make or break it, a look at the software listing of any machine will sort it out from being a serious business type machine to a simple home arcade! (And right now the MTX looks like the latter!)

Computers always have been work machines (which can also entertain) and should be treated as such so the number of jobs a computer can do is important to its success. (Just look at the BBC - mediocre hardware bit bags of software to do anything).

So the moral of this story is more hardware information and enough hardware to suit all users (from the games player who wants bigger and better games to the small business with a large data base and information processing requirement). This should help create interest in the machine, and possibly, a wider variety of usable software, to aid, not entertain, would be made available to the end user. More work! Less play!

3 Mayburn Bank, Loanhead, Midlothian, EH20 9EZ.
Since I have just fitted a NewWord ROM to our MTX512, and understand that it is not possible to run Pothole Pete with the ROM fitted, I would like to swap "Pothole Pete" for "Goldmine". Could you please advertise the above in the next issue of "Memopad".

With regard to the aforementioned ROM, I was somewhat disappointed to find that, having purchased a printer with several character styles, auto-underlining, etc. i.e. STAR SG10, I am unable to set these up from NewWord. Could you please advise me if a version of the ROM which supports the features of my printer is available at a replacement price, or is there a programme which can be input to allow such calls to be made.
Yours sincerely, J. CULLEN. C1122.

You can send custom print controls to your printer, via NewWord, by using the ^XR & ^XQ ^XW & ^XE commands.

Suppose, for instance, that you wanted to use the ELONGATED TEXT mode from within your document. On the DMX80 the control sequence is:

ESC + "W" + CHR$(1) ..... Turn it on.
ESC + "W" + CHR$(0) ..... Turn it off.

NewWord expects the sequences to be sent in hexidecimal form, and they are transmitted by using the DOT COMMAND.

The HEX code for ESC = 1B
The HEX code for "W" = 57

To install the command you would type:
.XR1B571 ..... TURN IT ON
.XQ1B570 ..... TURN IT OFF

Now, whenever you want to turn the elongated mode on you would type: ^PR. And to turn it off type: ^PQ. Your document would then look like this:-

.XR1B571
.XQ1B570

Now is the time for ^Rall good men^Q to come to the aid of the party.

NOTE ^ signifies the CTRL key.

ED.

HAVE YOU GOT A PEDIGREE CERTIFICATE?

Mr. H. J. Arkle, 12 Woodford Close, Witherwack Estate, Sunderland, Tyne & Wear would like any members who are into family-trees and Genealogy to get in touch. How about it Mr. Smith??

The following is an extract from Church Computer. Interesting to say the least!

THE MACHINE MARKET

(Article in the November edition (12) of Church Computer)

Nigel Hardcastle on the changing market scene

GOODBYE SPECTRUM?
GOODBYE BBC?

Lots of different machines are used by our members but the Spectrum and the BBC have a special place. They have respectively the claim to be the cheapest realistic tape and disc based system.

Both will continue to be used and bought for some time to come. The sheer volume of software and the vast user base makes it certain that present owners have chosen well. For educational programs and specialist church programs they must be the best provided for group in the UK. But things are changing and their time as the most obvious choices may be coming to an end.

What has brought this about is the drop in the price of CP/M and MSDOS machines which only a year ago were so expensive that most of us didn't even bother to read about them. In particular the Amstrad 6128 and 8256, the cheaper Apricots and now a 30% discount for CCUG members on Ferranti list prices. (Incidentally Keith, the "offer being suggested is a Ferranti PC860 (Twin floppies), Philips BM7502 Mono Monitor and a Centronics GLP Printer/tractor for #1311 inclusive of VAT).

What are CP/M and MSDOS y6ight be asking? Well they are "Operating Systems". These are programs for suitably designed software that try to make one machine look like another. CP/M runs on a wide range of 8 bit micros that use the Z80 chip. It doesn't matter what micro you have, if it will run CP/M it will run WORDSTAR and about 3000 other programs. (Editors please note there can be some compatability problems). Unfortunately, earlier Amstrads only had enough memory to run a very few CP/M programs. MSDOS is a similar system for 16 bit machines. Naturally, software houses like writing software for these systems because they are so easy to convert and so many people can buy them.

NOT ALL FRUSTRATION IS DUE TO THE CONSERVATIVES

Dear Genpat People,

I am writing this letter in total frustration! Let me introduce myself. My name is Kevin Hoffman and I'm an exasperated Memotech owner from the colonies. My exasperation is due to the management and staff of Memotech Incorporated, USA. I have many minor problems with my system. Foremost is the fact that Memotech-USA does not seem to be able to supply any information on the undocumented files supplied on the system disc that came with my computer, such as; VDEB.COM, TVI.COM, COLOR.COM, BAUD.COM, etc..

I have also been unable to get certain data such as I/O port addresses used by peripherals accessed through the RS-232 ports or how to initialize said ports. It seems that MTX-USA's knowledge is limited to off the shelf software and how to use them out of the box without any understanding of what's happening internally.

I'm not a total novice to computers, nor am I an expert hacker. I am on a kind of middle ground right now. Due to the lack of all software not being equal I have undertaken the task of obtaining some degree of programming skills in the assembly and basic dialects, but not knowing various pertinent addresses seems to be having certain detrimental affects on my applications progresses. Any help you could give on this score would be greatly appreciated.

Another problem I have been plagued with has been in the area of telecommunications. It seems that a gentle soul named Ward Christianson wrote a communication program which utilises a error detection system during file transfers that virtually certifies 99.98% botch up protection. Now on the surface this seems like a good thing, but since I don't have it as part of my CONTACT.COM communication package I have suffered many frustrating data cut offs. It has been getting more and more popular to incorporate this xmodem protocol, as it's called, into existing RCPM and BBS systems. And they do it in such a way that you are cut off from receiving your file using the "TYPE" command after some predetermined amount of lines have been transferred, then you are told to use "S XMODEM [filespec]". When this happens my grey matter chills. Now to the quick of it. Being as how CONTACT.COM utilises outside modules for file transfers, I was wondering if you people across the seas have run into the same problem and possibly made up different modules supporting this protocol.

Another point is the lack of literature on this system. I know that there are more books on this computing gem but I have not been able to find them on this continent. Currently I own "The Memotech Games Book" by Owen and Audrey Bishop, "The Memotech MTX Program Book" by Peter Goode, "Memotech Computing" by Sir Ian Sinclair and the manuals that came with my system, however I have hear of a couple more titles and I'm sure you have heard of a few yourselves.

Now I know it sounds like I'm deriding Memotech-USA somewhat so I have to say that they have been free with the knowledge of this jewel that they possess, but it still seems to me that they fall short somewhere. After all they were the ones who gave me your address and telephone number and told me of how much others have benefitted from contacting you. It just seems to me that if others have benefitted why hasn't MTX-USA joined in the fray.

Speaking of telephone numbers. I tried calling the number they gave me (my first overseas call) and it sounded like a computer answered. I tried again using my computer and modem but all I got was garbage on my screen. Since I'm using an American modem that utilises the "Bell" standards I assume you're using a modem that utilises an European standard.

In closing I would like to say that any help you could give would be so fantastically appreciated. I have heard that you people are a users group that's about 5000 strong and I was wondering if it were possible for me to join? I subscribe to a service called Compuserve and am a member of the CP/M SIG there. If you say that it is a go, I would like to place any useful information you supply pertinent to the Memotech on line there. I am also willing to pay any reasonable remittance you request. Further I have roughly three hundred public domain programs I have acquired on line and I would be willing to share them with you, along with any that I author.

My system is a Memotech FDX1000 - dual 5.25 floppy drives configured as double sided, double density, 310k capacity - 247k silicon ram disc (that syspool.com doesn't work on!) - Promethius Products Inc.(s Promodem 1200 (300-1200 baud, Hayes instruction set compatible) modem - and a DMX80

(Panasonic) printer.

Desperately yours, KEVIN M. HOFFMAN ★

Flash! *Len Clark*

This ingenious routine solves the problem of knowing when the MTX is in lower case mode. It will display a flashing cursor at the top of the screen.

MTX 500 owners type in the listing as shown. However your addresses (numbers to the left of the listing) will start at 8007.

When you have typed it in, make sure you save it before 'Running'. If it works correctly after you have typed 'Run' <RET> and you enter lower case, behold! A flashing cursor (complete with dirty mac).★

```
10 CODE

4007          LD HL,£FA98
400A          LD (HL),£C3
400C          INC HL
400D          LD DE,£F000
4010          LD (HL),E
4011          INC HL
4012          LD (HL),D
4013          LD BC,INTRTN
4016          PUSH BC
4017          LD HL,END
401A          XOR A
401B          SBC HL,BC
401D          PUSH HL
401E          POP BC
401F          INC BC
4020          POP HL
4021          DI
4022          LDIR
4024          EI
4025          LD HL,£FD5E
4028          SET 4,(HL)
402A          RET
402B INTRTN:  LD HL,£FA91
402E          BIT 7,(HL)
4030          LD HL,STORE
4033          JR Z,OFF
4035          LD (HL),£00
4037          JR NOFLSH
4039 OFF:     INC (HL)
403A          BIT 5,(HL)
403C          JR NZ,NOFLSH
403E          BIT 4,(HL)
4040          JR Z,NOFLSH
4042 FLASH:   LD B,£7F
4044          JR OUTPUT
4046 NOFLSH:  LD B,£20
4048 OUTPUT:  LD A,£27
404A          OUT (£02),A
404C          LD A,£5C
404E          OUT (£02),A
4050          LD A,B
4051          OUT (£01),A
4053          RET
4054 STORE:   DB £00
4055 END:     RET
4056          RET

Symbols:
INTRTN  402B    END     4055
OFF     4039    STORE   4054
NOFLSH  4046    FLASH   4042
OUTPUT  4048
```

Rainbow

This short program allows you 29 colours to be used on the screen at one time. This must stimulate your imaginations as to its possibilities.

```
1 REM
2 REM  PROGRAMMED BY JOHN GRAYSON, AMPFIELD, NR. ROMSEY. HANTS.
3 REM
5 VS 4: CLS : LET V=-8: LET Y=-1: LET C=0
10 FOR U=1 TO 14: READ A,B
30 LET C=C+1: IF C>7 THEN  LET X=18 ELSE  LET X=4
40 IF C=8 THEN  LET Y=-1: LET V=-8
80 INK 14: LET Y=Y+2: CSR X,Y: PRINT A,B
85 IF C>7 THEN  LET X1=120: LET X2=140 ELSE  LET X1=10: LET X2=30
90 LET V=V+16: FOR T=195-V TO 182-V STEP -2: INK A: LINE X1,T,X2,T: INK B: LINE
X1,T-1,X2,T-1: NEXT : NEXT U
92 CSR 8,19: INK 14: PRINT "NORMAL COLOURS:";: GENPAT 0,64,127,127,127,127,127,1
27,127,127: GENPAT 0,91,254,254,254,254,254,254,254,254
93 FOR H=0 TO 29 STEP 2: INK (H/2)+1: CSR H,21: PRINT "@[": CSR H,22: PRINT "@["
: NEXT H
95 DATA   11,14,10,14,3,9,2,8,6,12,3,10,3,7,7,14,2,5,7,11,7,10,13,2,8,13,5,13
96 GOTO 96
```

Assembly Line

Terry Trotter
Editor

Billingham
Cleveland
TS23 2HW

65 Marsh House Avenue

Hello again, this month we have, as promised, some routines to create sprites from the VRAM character tables and a printer driver routine as well as a routine to extend the Front Panel to include a Fill command, and also a few useful ROM calls for you to try.

Title : VDPSPR. VDP Sprite.

This routine copies the Ascii character set from VRAM (Pattern Generator Table) into RAM to make it easy to access from BASIC. It can then be used, as shown for example to create a sprite from the data.

The routine only needs to be run once, with either a USR command or a CALL from another assembly language program.

The program below shows how to create a sprite from the character stored in the string variable A$

```
1 GOTO 8
5 CODE

400E STORE:  DS 192      ;storage for
40CE         DS 192      ;the pattern
418E         DS 192      ;generator
424E         DS 192      ;table
430E         NOP
430F START:  LD HL,£1900        ;table address
4312         LD A,L      ;access the VDP
4313         OUT (2),A
4315         LD A,H
4316         OUT (2),A
4318         NOP         ;transfer the
4319         LD HL,STORE        ;data
431C         LD D,4
431E LOOP:   LD B,192
4320         LD C,1
4322         INIR
4324         DEC D
4325         LD A,0
4327         CP D
4328         JP NZ,LOOP
432B         NOP
432C         RET

Symbols:
STORE   400E    LOOP    431E
START   430F

8 INPUT "character for sprite = ";A$
9 LET A=USR(17167)
10 LET N=16398+(8*(ASC(A$)-32)): REM will find start of chr. = A$
20 GENPAT 3,1,PEEK(N),PEEK(N+1),PEEK(N+2),PEEK(N+3),PEEK(N+4),PEEK(N+5),PEEK(N+6
),PEEK(N+7): REM reads next bytes into genpat statement
100 VS 4: CLS : CTLSPR 2,1: SPRITE 1,1,128,96,0,0,1: PAUSE 5000
900 LET A$=" "
901 PRINT ASC(A$)
```

Title: LPRINTA:

Transfers a character stored in the Accumulator A to the Centronics printer port

```
10 CODE

4007 LPRINTA:PUSH AF     ;save A on the stack
4008 LPTALP: IN A,(£4)   ;read the status of the printer
400A         AND £0F
400C         CP £0A      ;test for correct bits set ... see note
400E         JR NZ,LPTALP        ;if printer not ready re-test
4010         POP AF      ;fetch the data to send
4011         OUT (£4),A  ;send it to the printer
4013         NOP         ;wait
4014         IN A,(£0)   ;strobe the data out
4016         NOP
4017         IN A,(£4)
4019         RET

Symbols:
LPRINTA 4007     LPTALP  4008
```

Note If you have a different printer then you may need to alter the value used in the comparison, the value given is correct for the DMX80 and Epson MX80 series.

Editor's Note

If you dis- assemble the ROM code at #0CE0 you will find a routine much like this one that the MTX itself uses for Centronics I/O. The CALL at the beginning to #0CF3 checks for the BREAK key being pressed and aborts if it has.

Other useful ROM routines

For those who wish to play around with the sound generator chip there is CALL to #093A. This is a ROM routine to shut off all the sound channels, which is handy if you want to stop them all in a hurry. All registers used are preserved by the routine.

A routine to sound the "bell" is at #0953. N"B" interrupts must not have been disabled by you prior to this call.

We conclude this month with the Front Panel expansion. The purpose of the program is to add to the commands in PANEL a "FIII" command. This will allow you to fIII a specIfIed block of memory with a particular byte. To use the program load and run the code given, this will put the fIII routine in high memory and you can then delete the code in line 10. In use it acts like any of the other commands, a single key "F" will prompt you for the rest of the parameters. I've found this routine especially useful when playing around transferring VRAM to RAM.

```
10 CODE

4007 PANEXT: LD DE,£F6F0          ;point to where we will be
400A         LD (£FA92),DE        ;change stack limit
400E         LD (£FA9F),DE        ;change panel expand bytes
4012         LD HL,FILL ;point to the start
4015         LD BC,£27  ;how many bytes to move
4018         LDIR       ;move them !
401A         LD A,£C3   ;set up the jump to fill
401C         LD (£FA9E),A
401F         RET
4020 FILL:   LD A,(£FD7D)         ;was the last character a "F" ?
4023         CP "F"
4025         RET NZ     ;if not return
4026         RST 28     ;print Fill and get bytes
4027         DB £AB
4028         DB "Fil",£EC         ;last byte has bit 7 set to 1
402C         PUSH BC    ;save the start address on stack
402D         RST 28     ;print To and get bytes
402E         DB £AB
402F         DB "T",£EF ;last byte has bit 7 set to 1
4031         PUSH BC    ;save the end address on stack
4032         RST 28     ;print "With" and get byte
4033         DB £AB
4034         DB "Wit",£E8         ;last byte has bit 7 set to 1
```

```
4038            LD A,C      ;fetch the byte into A
4039            POP HL      ;get the end address
403A            POP DE      ;get the start address
403B            AND A       ;clear carry flag
403C            SBC HL,DE   ;calculate how many bytes
403E            LD B,H      ;set up length in BC
403F            LD C,L
4040            LD H,D      ;set up start address in HL
4041            LD L,E
4042            LD (DE),A   ;fill the first byte
4043            INC DE      ;point to the next with DE
4044            LDIR        ;fill the rest
4046            RET

Symbols:
FILL    4020    PANEXT  4007
```

Well that's all for this month, next month we will have a routine to run a BASIC program from assembler and lots more. Keep on sending those routines.

Contributors

P. Alkman	:	VDPSPR
P. Crighton	:	LPRINTA
T. Trotter	:	Panel expansion, ROM calls.

Disc Adaption for Edasm *G.Dunsby*

As many readers know EDASM is an excellent Macro Assembler but it does at present only exist on cassette, a shortcoming that perhaps Syntaxsoft will soon remedy. However all is not lost for SDX Users as with a bit of thought we can produce a workable disc version. The first step is to transfer EDAM from cassette to disc. To do this load EDASM as usual but instead of running it type:-

```
USER  WRITE "E .2",16475,16565
```

The next problem concerns the convenience of switching between assembler and BASIC without using PANEL. The cassette version uses the USER command but it's not possible now a the disc system is accessed with this function. However a short piece of code that neatly fits between the two disc routines at F5CA Hex. overcomes this.

With this in mind type in listing 1 and once again save it to disc with:-

```
USER  SAVE  "EDASM.BAS"
```

Now type in listing 2 and once again save it with:-

```
USER  SAVE  "E.1"
```

If everything has gone according to plan you should now have 2 programs (E.1 and EDASM) and a block of code (E.2) on disc.

The assembler can now be loaded from disc by typing:-

```
USER  LOAD  "EDASM.BAS"
```

Run it and you should get the READY message and then run it again and all being well the Yellow and Black EDASM Screen will appear before your very eyes.

At this point ^Q to exit the assembler (as per manual) and type in the short BASIC line:-

```
10 RAND USR(62922)
```

Type RUN and you will be back with the assembler and you are ready to go. The BASIC line will not b erased by any assembler operation as the Source Code starts at 4010 Hex. and just leaves enough room.

The chances are that when you have typed in and assembled your machine code masterpiece you will want to save it for future use. To do this without using cassettes is quite simple as long as you are careful.

After assembling your program with the ^A command EDASM gives out a piece of information that looks something like this:-

```
Workarea - 4B40 to 4DB1
ORG end - C4EC
LOAD end - C4EC
```

(The Hex. figures are obviously dependant upon the size of your particular program).

So to save your programme to disc first assemble it and take note of the start of the work are. (In this case it is 4B40 Hex.). Exit the assembler with ^Q and then save the program to disc as a section of bytes . In this case:-

```
USER WRITE " NAME.COD",16400,2864
```

Always use 16400 as the address (=4010 Hex which is the start of Source Code) and the length of code is the work area address minus 4010 He x. In this case it is 4B40 - 4010 Hex. = B30 Hex.

To load a program back into the assembler first load EDASM and then exit it as before and type in:-

```
USER READ "name.COD",16400
```

and then press RUN and 'O' (when asked) and your program will be back into the assembler.

The last point concerns the merging of one or more programs or subroutines an easy task with cassette but slightly more difficult with disc.

I have found that the best way to save routines to disc is to have a standard first line and two last lines, i.e:-

```
ORG C350H
, your
; program
RET
END
```

If you do this, there is an easy formula for merging two programs.

Assemble the first program in the assembler with ^A as normal (ensuring that the last two lines are RET and END) then subtract 11 (0B Hex) from the Workarea address to get the address to where the second program can be loaded.

If we take the previous example we would load the second program at 4B40-B Hex = 4B35 Hex. (19253 Dec.). So we would exit the assembler and type:-

```
USER READ "NAME2.COD",1919253
```

We have not quite finished yet unfortunately as we have corrupted the letter 'T' in RET in the first program. So enter PANEL and press D (display) and the address to where you loaded NAME2. You will see RE. In the above e xample you would type PANEL then D and 4B35 . Now type in 54 (Ascii 'T') and you should have completed the RET command. Exit panel and re-enter EDASM with RUN and 'O' and you will have your two programs merged and ready for any editing that you might wish to do.

At first you may think that it's not worth the bother but after a couple of trials you'll find it very easy and far quicker and reliable than cassette. ★

```
LISTING 1  EDAM.BAS
10 CODE
4007                LD DE,#F5CA
400A                LD HL,START
400D                LD BC,9
4010                LDIR
4012                RET
4013 START:         DI
4014                LD A ,#80
4016                OUT (0),A
4018                JP #014E
401B                RET

Symbols:
START     4013
```

```
20 USER  LOAD "E.1"

LISTING 2 'E .1'

10 GOTO 20
15  CODE

400F        DS12
401B        RET

Symbols:

20 USER READ "E.2",16475
30 RAND USR (16 475):RAND USR(16519)
35 NEW
```

Debounce those Keys *Dave Bain*

```
10 CODE

4007            LD HL,(#FD52)       ;GET CURRENT ADDRESS FROM USERIO
400A            LD (KBJMP),HL       ;INSERT IT INTO THE PROGRAM
400D            LD DE,#F800         ;ADDRESS OF DELAY ROUTINE
4010            OR A                 CLEAR CARRY FLAG
4011            SBC HL,DE           ;HAS PROGRAM ALREADY BEEN RUN ?
4013            RET Z               ;RETURN IF IT HAS
4014            LD HL,KBD           ;LOAD DELAY ROUTINE INTO FREE MEMORY
4017            LD DE,#F800
401A            LD BC,#000F
401D            LDIR
401F            LD HL,#F800         ;LOAD USERIO WITH DELAY ROUTINE
4022            LD (#FD52),HL
4025            RET
4026 KBD:       DB #CD              ;OP CODE FOR CALL
4027 KBJMP:     DS 2                ;ADDRESS FOR CALL (OLD USERIO)
4029            RET Z               ;RETURN IF NO KEY PRESSED
402A            PUSH AF             ;SAVE KEY ON STACK
402B            LD BC,#1000        ;DELAY TIME CONSTANT
402E LOOP:      DEC BC              ;DELAY LOOP
402F            LD A,B
4030            OR C
4031            JR NZ,LOOP
4033            POP AF              ;GET KEY FROM STACK
4034            RET
```

JUST TYPE IN THE PROGRAM AND IT IS INSTALLED UNTIL YOU SWITCH OFF. ★

Sound Editor P Brewer

Here is a neat little routine that should help all of you budding musicians. This routine will create sounds for you without the need to use a mainframe. Just shows you what a small program can do when you think about it.

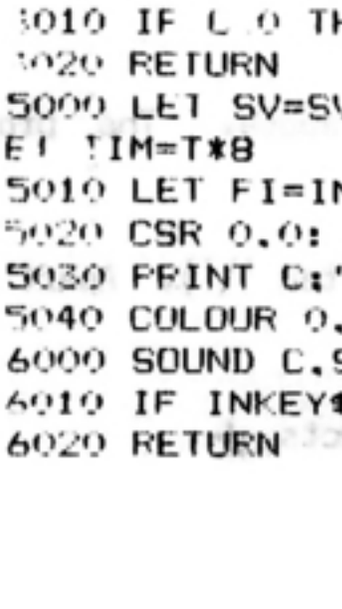
```
10 REM Sound Editor P.Brewer 1986
20 SBUF 10: PLOD "PROG"
30 VS 4: PAPER 1: CLS : PRINT CHR$(29): COLOUR 4,3
40 LET SVOL=0: LET SFR=0: LET EVOL=0: LET EFR=0: LET T=6: LET C=0: GOSUB 5000
50 LET K=ASC(INKEY$)
60 IF K=26 THEN  GOSUB 6000
70 IF K=32 THEN  SOUND C,0,0
80 IF K=11 THEN  LET C=C+1: GOSUB 3000
90 IF K=10 THEN  LET C=C-1: GOSUB 3000
100 IF K>1 THEN  GOSUB 5000
110 IF K=128 AND SVOL<23 THEN  LET SVOL=SVOL+1: CSR 0,24-SVOL: COLOUR 1,11: PRIN
T CHR$(130);
120 IF K=129 AND SVOL>0 THEN  LET SVOL=SVOL-1: CSR 0,23-SVOL: PRINT " ";
140 IF K=130 AND SFR<23 THEN  LET SFR=SFR+1: CSR 1,24-SFR: COLOUR 1,8: PRINT CHR
$(130);
150 IF K=131 AND SFR>0 THEN  LET SFR=SFR-1: CSR 1,23-SFR: PRINT " ";
160 IF K=134 AND EFR<23 THEN  LET EFR=EFR+1: CSR T,24-EFR: COLOUR 1,8: PRINT CHR
$(130);
170 IF K=135 AND EFR>0 THEN  LET EFR=EFR-1: CSR T,23-EFR: PRINT " ";
180 IF K=132 AND EVOL<23 THEN  LET EVOL=EVOL+1: CSR T+1,24-EVOL: COLOUR 1,11: PR
INT CHR$(130);
190 IF K=133 AND EVOL>0 THEN  LET EVOL=EVOL-1: CSR T+1,23-EVOL: PRINT " ";
200 IF K=8 AND T>2 THEN  GOSUB 1000: LET T=T-1: GOSUB 1050
210 IF K=25 AND T<30 THEN  GOSUB 1000: LET T=T+1: GOSUB 1050
300 GOTO 50
1000 FOR I=0 TO EVOL-1: CSR T+1,23-I: PRINT " ";: NEXT
1010 FOR I=0 TO EFR-1: CSR T,23-I: PRINT " ";: NEXT
1020 RETURN
1050 FOR I=0 TO EVOL-1: CSR T+1,23-I: COLOUR 1,11: PRINT CHR$(130);: NEXT
1060 FOR I=0 TO EFR-1: CSR T,23-I: COLOUR 1,8: PRINT CHR$(130);: NEXT
1070 RETURN
3000 IF C>3 THEN  LET C=0
3010 IF C<0 THEN  LET C=3
3020 RETURN
5000 LET SV=SVOL*41: LET SF=(24-SFR)*340: LET EV=EVOL*41: LET EF=(24-EFR)*340:
EL TIM=T*8
5010 LET FI=INT((EF-SF)/TIM): LET VI=INT((EV-SV)/TIM)
5020 CSR 0,0: COLOUR 1,15: COLOUR 0,4
5030 PRINT C;",";SF;",";SV;",";FI;",";VI;",";TIM;",1"
5040 COLOUR 0,1: RETURN
6000 SOUND C,SF,SV,FI,VI,TIM,1
6010 IF INKEY$<>"" THEN  GOTO 6010
6020 RETURN
```

PROG

```
*D PG1.
*E
*R
```

PG1

SOUND EDITOR



```
F1 ---------- Increase initial volume.
F2 ---------- Decrease initial volume.
F3 -------- Increase initial frequency.
F4 -------- Decrease initial frequency.
F5 ---------- Increase end volume.
F6 ---------- Decrease end volume.
F7 ---------- Increase end frequency.
F8 ---------- Decrease end frequency.
Left & Right cursor --------- Duration.
Up & Down cursor ------------ Channel.
HOME ------------------- Start sound.
SPACE ------------------- Stop sound.
```

Press RET to continue.

Clash of the Robots

P Crighton

Clash of the Robots is a game in which two robots with predefined programs fight each other on a grid of 10 x 15 squares. The robots have alternate turns and at each turn must do one of five things :

MV - Move forward by one square. If it reaches the edge it automatically turns round.
RL - Rotate left through 90 degrees.
RR - Rotate right through 90 degrees.
FR - Fire forwards with a range of five squares. If it lands on the enemy robot it scores one hit.
FD - Find. This causes the robot to turn towards the enemy robot, although it is only a general direction as the robot cannot face along a diagonal.

The player does not take his turn at each move but writes a program 22 instructions long consisting of the above abbreviations. When an Enemy program is chosen your program will be stepped through instruction by instruction which will control what your robot does. When one robot is destroyed then you will be told how long it took and who was destroyed. To destroy a robot the enemy must either 'ram' the other or to fire two hits.

The idea of the game is to write the optimum program to destroy the other robots in the fastest time, but with just five commands there are 2.38 x 10^15 possible combinations in the 22 instructions!

There are four built in Enemy programs and one player program. The Enemy:-

Killer - Lots of firing.
Seeker - Emphasis on finding the player before attacking.
Random - As it's name suggests 22 random commands.
Runner - Moves around a lot.

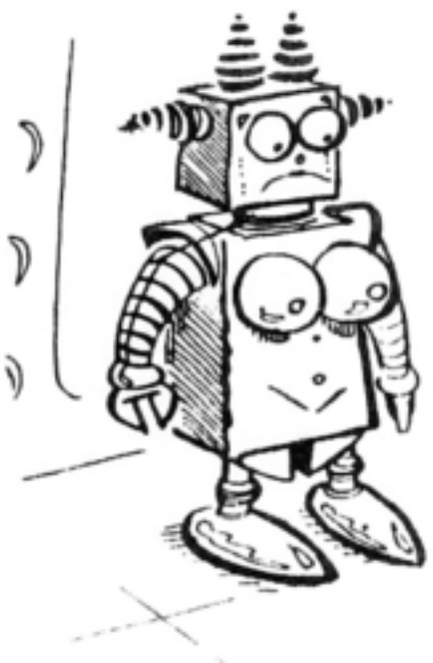
If you choose an enemy and run it against your program you will see the two robots (looking somewhat like tanks!) on the battleground performing their instructions. To interrupt a game hold down any key, except 'BRK'.

If you choose to type your own program you must use the two-letter abbreviations above. The program then returns to the menu for you to select the Enemy.

Lines 2000 & 2020 both contain the symbol '|', and this is the vertical line (shifted '\'). Apart from this entering should be straight forward.

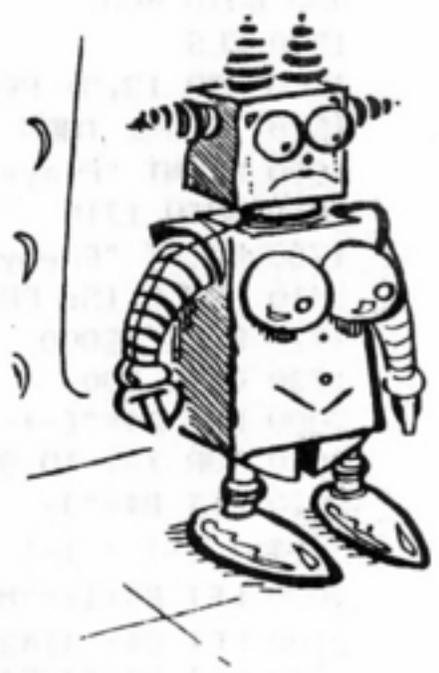
When playing the game you should turn up the TV sound as there are some sound effects.★

```
0 REM ************************************
1 REM *     C L A S H    O F    THE     *
2 REM *          R O B O T S            *
3 REM *              B Y                *
4 REM *     P .    C R I G H T O N      *
5 REM ************************************
8 SAVE "ROBOT CLASH"
9 VS 5: CLS
10 PAPER 6: CSR 9,5: PRINT "Clash of the Robots": CSR 17,9: PRINT "By": CSR 11,13: PRINT "Peter
Crighton"
20 GENPAT 0,43,16,16,16,16,252,16,16,16
30 GENPAT 0,45,0,0,0,0,252,0,0,0
40 GENPAT 0,47,16,16,16,16,240,0,0,0
50 GENPAT 0,91,16,16,16,16,28,16,16,16
60 GENPAT 0,92,0,0,0,0,240,16,16,16
70 GENPAT 0,93,0,0,0,0,28,16,16,16
80 GENPAT 0,94,0,0,0,0,252,16,16,16
90 GENPAT 0,95,16,16,16,16,28,0,0,0
100 GENPAT 0,123,16,16,16,16,240,16,16,16
110 GENPAT 0,124,16,16,16,16,16,16,16,16
120 GENPAT 0,125,16,16,16,16,252,0,0,0
130 GENPAT 1,129,0,48,120,252,204,204,204,0
140 GENPAT 1,130,0,60,124,224,224,124,60,0
150 GENPAT 1,131,0,204,204,204,252,120,48,0
160 GENPAT 1,132,0,240,248,28,28,248,240,0
```



```
170 GENPAT 1,133,0,0,0,48,48,0,0,0
180 GENPAT 1,134,0,48,120,252,252,252,252,0
190 GENPAT 1,135,0,252,252,252,252,120,48,0
200 GENPAT 1,136,0,240,248,252,252,248,240,0
210 GENPAT 1,137,0,60,124,252,252,124,60,0
220 DIM E$(5,2),D(22,5),A$(80),B$(840)
230 LET C$=CHR$(129)+CHR$(130)+CHR$(131)+CHR$(132): LET D$=CHR$(134)+CHR$(137)+CHR$(135)+CHR$(13
6)
240 GOSUB 2000
300 LET V=0: LET A=1: LET B=1: LET C=1: LET D=0: LET E=2: LET F=2
310 CLS
320 GOSUB 7000
330 IF X=49 THEN  GOTO 8500
340 LET G=X-48: LET H=G
350 CLS
360 PRINT B$: CSR 31,5: PRINT CHR$(129);" Your": CSR 33,7: PRINT "Robot": CSR 31,13: PRINT CHR$(
134);" Enemy": CSR 33,15: PRINT "Robot": CSR 6,22: PRINT "Number of moves :"
370 GOSUB 5000
400 CSR 23,22: PRINT V
410 GOSUB 8000
420 IF INKEY$>"" THEN  GOTO 300
430 IF D>0 THEN  GOTO 1500
440 IF H=1 THEN  LET V=V+1: LET H=G: LET C=B: LET J=L: LET M=P: LET Q=S ELSE  LET H=1: LET C=A:
LET J=K: LET M=N: LET Q=R
450 GOTO 400
1500 CLS
1510 CSR 13,5: PRINT "End of game": CSR 8,10
1520 IF D=2 THEN  GOTO 1700
1600 PRINT "Player robot destroyed"
1610 GOTO 1710
1700 PRINT "Enemy robot destroyed"
1710 CSR 8,15: PRINT "Number of moves :";V
1720 PAUSE 6000
1730 GOTO 300
2000 LET A$="[-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-{        ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !        "
2010 FOR X=1 TO 9: LET B$=B$+A$: NEXT
2020 LET B$="]-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-^-\        ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !        "+B$+
"_-)-)-)-)-)-)-)-)-)-)-)-)-)-)-)-)-)-)-/        "
2030 LET E$(1)="MV": LET E$(2)="RL": LET E$(3)="RR": LET E$(4)="FR": LET E$(5)="FD"
2100 LET Q$="414311431413141434144 1": LET G=2: GOSUB 3000
2110 LET Q$="425434111134345454354 5": LET G=3: GOSUB 3000
2120 LET Q$="523123313454243225112 3": LET G=4: GOSUB 3000
2130 LET Q$="411121141211412111121 1": LET G=5: GOSUB 3000
2140 LET Q$="545415415415442541541 1": LET G=1: GOSUB 3000
2150 RETURN
3000 FOR X=1 TO 22: LET D(X,G)=VAL(Q$(X,1)): NEXT
3010 RETURN
5000 LET N=(INT(RND*15)*2)+1: LET R=(INT(RND*10)*2)+1: LET K=INT(RND*4)+1: LET J=K: LET M=N: LET
 Q=P: LET H=1: GOSUB 6000
5010 LET P=(INT(RND*15)*2)+1: IF P=N THEN  GOTO 5010
5020 LET S=(INT(RND*10)*2)+1: IF S=R THEN  GOTO 5020
5030 LET L=INT(RND*4)+1: LET J=L: LET M=P: LET Q=S: LET H=G: GOSUB 6000
5040 RETURN
6000 CSR M,Q: IF H=1 THEN  PRINT C$(J) ELSE  PRINT D$(J)
6010 RETURN
6200 CSR M,Q: PRINT " ": RETURN
7000 CLS : CSR 14,0: PRINT "Menu": CSR 0,3: PRINT "Write new program": CSR 29,3: PRINT "1": CSR
0,5: PRINT "Test current program against :"
7010 CSR 0,7: PRINT "Killer": CSR 29,7: PRINT "2": CSR 0,9: PRINT "Seeker": CSR 29,9: PRINT "3":
 CSR 0,11: PRINT "Random": CSR 29,11: PRINT "4"
7020 CSR 0,13: PRINT "Runner": CSR 29,13: PRINT "5": PRINT : PRINT : PRINT : PRINT "Please give
a choice from 1 to 5"
7030 LET X=ASC(INKEY$)
7040 IF X<49 OR X>53 THEN  GOTO 7030
7050 PRINT : PRINT "Choice : ";CHR$(X)
7060 PAUSE 400: RETURN
8000 LET X=D(C,H)
8100 ON X-1 GOSUB 8150,8200,8250,8300,8350
8110 IF H=1 THEN  LET A=A+1 ELSE  LET B=B+1
8115 IF A=23 THEN  LET A=1 ELSE  IF B=23 THEN  LET B=1
8120 RETURN
8150 GOSUB 6200
```

```
8155 IF J=4 AND M<29 THEN  LET M=M+2 ELSE  IF J=2 AND M>1 THEN  LET M=M-2
8160 IF J=1 AND Q>1 THEN  LET Q=Q-2 ELSE  IF J=3 AND Q<19 THEN  LET Q=Q+2
8165 IF M=1 AND J=2 THEN  LET J=4
8170 IF M=29 AND J=4 THEN  LET J=2
8175 IF Q=19 AND J=3 THEN  LET J=1
8180 IF Q=1 AND J=1 THEN  LET J=3
8185 GOSUB 8475: GOSUB 6000: IF H=1 THEN  LET N=M: LET R=Q ELSE  LET P=M: LET S=Q
8190 IF N<>P OR R<>S THEN  RETURN
8191 IF H=1 THEN  LET D=2: LET AA=6: LET AB=1000 ELSE  LET D=1: LET AA=5: LET AB=520
8192 FOR AC=15 TO -.5 STEP -1: SOUND 0,AB,AC: SOUND 3,AA,15: PAUSE 25: NEXT : SOUND 3,0,0
8195 IF H=1 THEN  LET D=2 ELSE  LET D=1
8196 RETURN
8200 GOSUB 6200: IF J=4 THEN  LET J=0
8210 LET J=J+1: GOSUB 6000: GOSUB 8475: RETURN
8250 GOSUB 6200: IF J=1 THEN  LET J=5
8260 LET J=J-1: GOSUB 6000: GOSUB 8475: RETURN
8300 LET W=M: LET Z=Q: GOSUB 8450: FOR X=1 TO 5: LET W=W+((J=2)-(J=4))*2: LET Z=Z+((J=1)-(J=3))*
2: IF H=1 THEN  LET AB=6 ELSE  LET AB=5
8305 IF W>29 OR W<1 OR Z>19 OR Z<1 THEN  RETURN
8310 CSR W,Z: PRINT CHR$(133);: SOUND 3,AB,16-X: PAUSE 100: PRINT CHR$(8);" ": SOUND 3,0,0: PAUS
E 100: IF T=W AND U=Z THEN  GOTO 8320 ELSE  NEXT : RETURN
8320 IF H=1 THEN  LET AB=1000 ELSE  LET AB=520
8325 FOR AA=15 TO -.5 STEP -1: SOUND 0,AB,AA: PAUSE 20: NEXT AA
8330 IF H=1 THEN  LET F=F-1 ELSE  LET E=E-1
8335 LET D=D-((F=0)*2)-(E=0)
8340 IF H=1 THEN  LET H=G: LET M=P: LET Q=S: LET J=L ELSE  LET H=1: LET M=N: LET Q=R: LET J=K
8342 GOSUB 6000: IF H=1 THEN  LET H=G ELSE  LET H=1
8345 RETURN
8350 GOSUB 6200: GOSUB 8450: IF M<T THEN  LET J=4 ELSE  IF M>T THEN  LET J=2
8360 IF Q<U THEN  LET J=3 ELSE  IF Q>U THEN  LET J=1
8370 GOSUB 8475: GOSUB 6000: RETURN
8450 IF H=1 THEN  LET T=P: LET U=S ELSE  LET T=N: LET U=R
8460 RETURN
8475 IF H=1 THEN  LET K=J ELSE  LET L=J
8480 RETURN
8500 CLS
8510 FOR X=0 TO 21: PRINT "Command";X;": ": NEXT X
8520 FOR X=1 TO 22: CSR 13,X-1
8525 INPUT "";S$: GOSUB 8700
8530 IF Y=6 THEN  GOTO 8800
8540 LET D(X,1)=Y: NEXT X: LET G=2: LET H=2: GOTO 300
8700 FOR Y=1 TO 5: IF S$=E$(Y) THEN  RETURN
8710 NEXT Y: RETURN
8800 CSR 20,X-1: PRINT "Invalid": SOUND 0,500,15: PAUSE 900: SOUND 0,0,0: CSR 20,X-1: PRINT "
    ": CSR 13,X-1: GOTO 8525
```



Books Now Available To Club Members

UNDERSTANDING C	11.95p
USING DBASEII	14.00p
CP/M DATABASE MANAGEMENT SYSTEMS	15.25p
SECRETS OF BETTER BASIC	9.95p
C PROGRAMMERS LIBRARY	13.95p
C MADE EASY	13.25p
DBASEII FOR FIRST TIME USER	14.95p
DOING BUSINESS WITH SUPERCALC	11.25p
UNDERSTANDING DBII	16.25p
PROGRAMMING IN C	13.25p
HOW TO BUILD A PROGRAM	14.25p

2 New Utilities

SDX UTILIES £9.95

SAVE AND LOAD VRAM

THIS PART OF THE UTILITY WILL SAVE VRAM TO A DISC FILE WHICH CAN BE RELOADED. SPRITE DATA IS ALSO SAVED. BY USING A COUPLE OF POKES ANY AREA OF VRAM CAN SAVED AND LOADED BACK.

DISC TO DISC BACKUP

THIS COMMAND ELIMINATES THE NEED FOR "USER COPY" Filename = Filename. YOU CAN BACKUP ALL FILES FROM THE DISC OR JUST THE ONES YOU SPECIFY.

RECOVER ERASED FILES.

WE HAVE ALL, AT SOMETIME OR OTHER, ERASED A FILE BY MISTAKE. THIS PART OF THE UTILITY WILL, IN MOST CASES, GET YOU OUT OF TROUBLE.

HEX TO ASCII DUMP

WILL PRODUCE AN HEXIDECIMAL & CHARACTER LISTING OF ANY NAMED FILE. WILL LIST TO PRINTER OR SCREEN.

JOIN BASIC PROGRAMS.

WITH THIS UTILITY YOU CAN JOIN TWO BASIC PROGRAMS ... YOU CAN NOW TRANSFER NODDY PAGES OR SUBROUTINES FROM ONE PROGRAM TO ANOTHER.

DISASSEMBLE.

THIS IS A FULL Z80 BASED DISASSEMBLER WHICH USES STANDARD Z80 MNEMONICS AND IS SUPPLEMENTED BY A SPECIAL SET OF MNEMONICS THAT ARE EXCLUSIVE TO THE MEMOTECH RANGE OF COMPUTERS. (SEE LISTING EXAMPLE)

A SPECIAL COMMAND IS INCORPORATED THAT ALLOWS YOU TO DISASSEMBLE ANY ROM BY SPECIFICATION.

THIS IS A UNIQUE UTILITY AND ONE THAT WE HAVE ALL BEEN WAITING FOR.

TAPE BASED SYSTEMS CAN PURCHASE THE DISASSEMBLER FOR £6.95

EDASM SDX VERSION £8.95

THIS IS THE DISC BASED VERSION OF EDASM WHICH HAS BEEN REWRITTEN TO INCLUDE A FEW MORE COMMANDS SUCH AS JOINING A SOURCE FILE FROM DISC TO THE SOURCE FILE IN MEMORY A MORE.

PLEASE NOTE THESE UTILITIES DO NOT WORK WITH CP/M BASED SYSTEMS. ★

```
1CFB D7          RST 10
1CFC 6F
1CFD 8A
1CFE 1E .
1CFF 41 A
1D00 73 s
1D01 73 s
1D02 65 e
1D03 6D m
1D04 62 b
1D05 6C l
1D06 65 e
1D07 3E >
1D08 EF          RST 28
1D09 E9 JP adr.#1429 = EDITOR
1D0A AA JP adr.#1448 = EDEND
1D0B D7          RST 10
1D0C 4F
1D0D CA971E      JP Z,#1E97
1D10 FE09        CP #09
```

```
13B7 EF          RST 28
13B8 C5 JP adr.#1286 = STACKACC
13B9 D7 JP adr.#1382 = SINE
13BA C6 JP adr.#1183 = EXCHANGE
13BB C1 JP adr.#128F = ACCSTACK
13BC D6 JP adr.#1298 = STACKOP1
13BD D8 JP adr.#137D = COSINE
13BE C4 JP adr.#129E = OP1STACK
13BF 8A JP adr.#109E = DIVIDE
13C0 C9          RET
```

Shared Bank Account

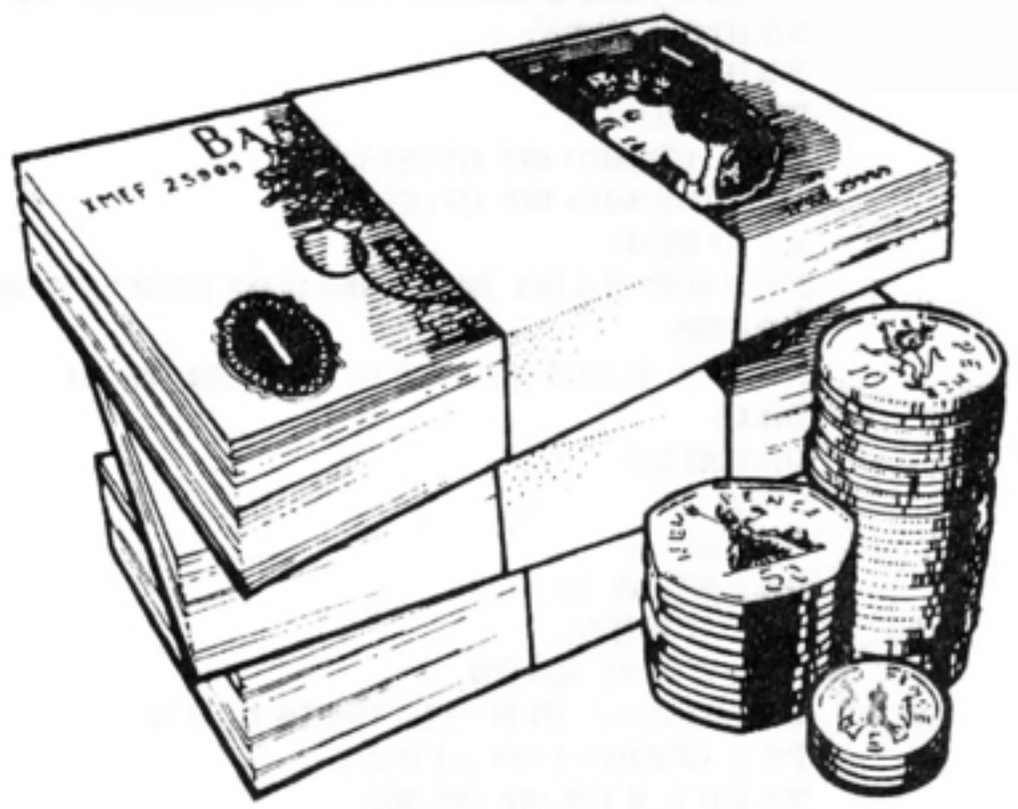
Bob Robinson

Part 3

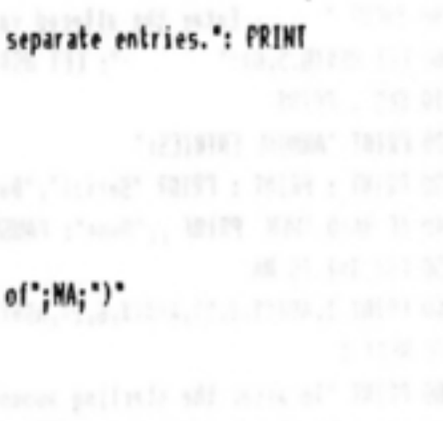
```
4590 IF LEFT$(J$(Z),2)="C"+NA$(1) THEN  PRINT " !";LEFT$(C$(Z),5);"!";NA$;"!  !";RIGHT$(J$(Z),8)+Q$;"!";L(Z);QQ$;"!"
4600 IF LEFT$(J$(Z),2)="W"+NM$(1) THEN  PRINT " !";LEFT$(C$(Z),5);"!";NM$;"!";RIGHT$(J$(Z),8)+Q$;"  !";L(Z);QQ$;"!"
4610 IF LEFT$(J$(Z),2)="W"+NA$(1) THEN  PRINT " !";LEFT$(C$(Z),5);"!";NA$;"!";RIGHT$(J$(Z),8)+Q$;"  !";L(Z);QQ$;"!"
4620 LET KK$=C$(Z): LET Q$="": LET QQ$="": LET II=0
4630 FOR I=1 TO 9-LEN (STR$(M(Z))): LET QQ$=QQ$+" ": NEXT I
4640 IF LEN (STR$(M(Z)))=9 THEN  LET QQ$=""
4650 FOR I=1 TO 10
4660 IF ASC(KK$(I))=0 THEN  LET II=II+1
4670 NEXT I: IF II=0 THEN  GOTO 4690
4680 FOR I=1 TO II: LET Q$=Q$+" ": NEXT I
4690 PRINT CHR$(28);" !";RIGHT$(C$(Z),5)+Q$;"!JNT.!  !       !  !";LEFT$(STR$(M(Z)),9);QQ$;"!"
4695 IF INKEY$=CHR$(13) THEN  LET R$(1)="         ": GOTO 1160
4700 NEXT Z
4710 PRINT " ------------------------------------"
4720 IF INKEY$()CHR$(13) THEN  GOTO 4720
4730 IF INKEY$=CHR$(13) THEN  PRINT CHR$(30): CSR 0,23: PRINT CHR$(30);CHR$(31): CLS : LET R$(1)="         ": GOTO 1160
4740 REM $$$$ (FROM 4500) $$$$
4750 IF INKEY$=" " THEN  GOTO 4750
4760 CLS : GOSUB 4770: GOSUB 4020: GOTO 4510
4770 REM  $$ (FROM 4480,4760) $$$
4780 CSR 4,0: PRINT "List of Entries from ";R$(1)
4790 PRINT "Cont.with SPACE BAR:To Finish,hold'RET'"
4800 RETURN
4810 REM  $$ PRINTER WITH )= 60 CHR$     per line. (from 4470) $$
4820 LPRINT "      ";COMP$;" -- Statement of Account"
4830 LPRINT "            List of Entries from ";R$(1)
4840 LPRINT "  ---------------------------------------------------"
4850 LPRINT " !DATE !AC.  !     £ ENTRY       !     £ BALANCE      !"
4860 LPRINT " ! +   !";NM$;"/!         !         !         ! JOINT   !"
4870 LPRINT " !REF. ! ";NA$;"!WITHDRAWAL!  CREDIT  !INDIVIDUAL!AVAILABLE !"
4880 LPRINT " !-----+-----+----------+----------+----------+----------!"
4890 FOR Z=Y TO E
4900 IF INKEY$=" " THEN  GOTO 4900
4910 LET KK$=J$(Z): LET Q$="": LET QQ$="": LET II=0: FOR I=1 TO 10
4920 IF ASC(KK$(I))=0 THEN  LET II=II+1
4930 NEXT I
4940 FOR I=1 TO 9-LEN (STR$(L(Z))): LET QQ$=QQ$+" ": NEXT I
4950 IF II=0 THEN  GOTO 4970
4960 FOR I=1 TO II: LET Q$=Q$+" ": NEXT I
4970 IF LEFT$(J$(Z),2)="C"+NM$(1) THEN  LPRINT " !";LEFT$(C$(Z),5);"!";NM$;" !         !";RIGHT$(J$(Z),8)+Q$;"  !";L(Z);QQ$;" !         !"
4980 IF LEFT$(J$(Z),2)="C"+NA$(1) THEN  LPRINT " !";LEFT$(C$(Z),5);"!";NA$;" !         !";RIGHT$(J$(Z),8)+Q$;"  !";L(Z);QQ$;" !         !"
4990 IF LEFT$(J$(Z),2)="W"+NM$(1) THEN  LPRINT " !";LEFT$(C$(Z),5);"!";NM$;" !";RIGHT$(J$(Z),8)+Q$;"  !         !";L(Z);QQ$;" !         !"
5000 IF LEFT$(J$(Z),2)="W"+NA$(1) THEN  LPRINT " !";LEFT$(C$(Z),5);"!";NA$;" !";RIGHT$(J$(Z),8)+Q$;"  !         !";L(Z);QQ$;" !         !"
5010 LET KK$=C$(Z): LET Q$="": LET QQ$="": LET II=0
5020 FOR I=1 TO 9-LEN (STR$(M(Z))): LET QQ$=QQ$+" ": NEXT I
5030 IF LEN (STR$(M(Z)))=9 THEN  LET QQ$=""
5040 FOR I=1 TO 10
5050 IF ASC(KK$(I))=0 THEN  LET II=II+1
5060 NEXT I: IF II=0 THEN  GOTO 5080
5070 FOR I=1 TO II: LET Q$=Q$+" ": NEXT I
5080 LPRINT " !";RIGHT$(C$(Z),5)+Q$;"!JOINT!         !         !         !";LEFT$(STR$(M(Z)),9);QQ$;" !"
5090 IF SP$="1" THEN  LPRINT " !     !     !         !         !         !         !"
5100 NEXT Z
5110 LPRINT "  ---------------------------------------------------"
5120 GOTO 1160
5130 REM $$ (FROM 1880,8440) $$
5140 PRINT "      LIMIT OF FILE ALLOCATION."
5150 PRINT "Final Individual Balances that apply onlast date as shown above are:            ";NM$;":-  £";B1
5160 PRINT "    ";NA$;":-  £";B2
5170 PRINT "  Write down these two Individual      Balances. Enter NEW Account.  Carry    Forvard the two balances and use '$' aslst. Reference Chr. after ";
5180 PRINT "date to give immediate entry."
5190 PRINT "  Press (RET)"
5200 INPUT O$: GOTO 1160
5210 REM $$ (FROM 1800,5720) $$$
5220 CSR 0,0: PRINT COMP$;" ACCOUNT STATEMENT": REM$$$$$$$$$ HAVE VARIABLES BEEN ENTERED?  IF NOT,PRESS'EDL', ENTER'RUN' THEN PRESS 'N'
5230 PRINT
5240 PRINT "All Withdravals are entered at once as vell as Credits to ";NM$;" or ";NA$;".Creditsare passed to JOINT balance only after";CL;" days to ";
5250 PRINT "allov for clearance."
5260 PRINT "  Up to 5 reference Chr$ may be added  immediately after the  date: (1st. Ref.Chr.='$' will avoid the";CL;" day delay)."
5270 PRINT "  Auto-credits operate without a delay."
5280 PRINT "  Entries  can only be made strictly indate order.  Corrections  can only be  made by converse  entries."
5290 PRINT "  ";K-Q-2,"entries (including Auto-)"
5300 PRINT "remain before file limit is reached."
```

```
5310 PRINT "  To show present position, Enter      today's date in statement display      followed by 'W';NM$(I);'0' or 'W';NA$(I);'0'."
5320 PRINT
5330 PRINT "When  ready, press <RET>."
5340 INPUT O$
5350 CLS
5360 RETURN
5370 REM ## ADDITION OF ENTRIES INTO     BALANCES (FROM 470) ##
5380 LET B1=B1+BB1: LET B2=B2+BB2: LET B3=B3+BB3
5390 LET B1=INT(B1*100+SGN(B1)*0.49)/100: LET B2=INT(B2*100+SGN(B2)*0.49)/100: LET B3=INT(B3*100+SGN(B3)*0.49)/100
5400 LET BB1=INT(BB1*100+SGN(BB1)*0.49)/100: LET BB2=INT(BB2*100+SGN(BB2)*0.49)/100: LET BB3=INT(BB3*100+SGN(BB3)*0.49)/100
5410 CSR 1,21: PRINT "!";PIN$(I,1,5);"!";NM$;"!  !";BB1: CSR 24,21: PRINT "!";LEFT$(STR$(B1),9): CSR 38,21: PRINT "!"
5420 CSR 0,23: PRINT
5430 CSR 1,21: PRINT "! Int.!JNT.!  !        !  !";LEFT$(STR$(B3),9): CSR 38,21: PRINT "!"
5440 CSR 0,23: PRINT
5450 LET A(Q+1)=A(Q)
5460 LET A(Q)=P
5470 LET B$(Q,11)="F"
5480 LET Q=Q+1
5490 LET R$(1)=PIN$(I,1,5)+" Int.": LET F$="C"+NM$(1)+STR$(BB1): GOSUB 5610
5500 CSR 1,21: PRINT "!";PIN$(I,1,5);"!";NA$;"!  !";BB2: CSR 24,21: PRINT "!";LEFT$(STR$(B2),9): CSR 38,21: PRINT "!"
5510 CSR 0,23: PRINT
5520 CSR 1,21: PRINT "! Int.!JNT.!  !        !  !";LEFT$(STR$(B3),9): CSR 38,21: PRINT "!"
5530 CSR 0,23: PRINT
5540 LET A(Q+1)=A(Q)
5550 LET A(Q)=P
5560 LET B$(Q,11)="F"
5570 LET Q=Q+1
5580 LET R$(1)=PIN$(I,1,5)+" Int.": LET F$="C"+NA$(1)+STR$(BB2): GOSUB 5610
5590 LET BB1=0: LET BB2=0: LET BB3=0
5600 RETURN
5610 REM ## STORAGE OF DATA (FROM 3740,  3880,5490,5580,8430) ###
5620 LET E=E+1
5630 LET C$(E)=R$(1)
5640 LET J$(E)=F$
5650 IF F$(2)=NM$(1) THEN  LET L(E)=B1
5660 IF F$(2)=NA$(1) THEN  LET L(E)=B2
5670 LET M(E)=B3
5680 IF B1+B2)=1E+8 THEN  PRINT "MAXIMUM BALANCE EXCEEDED: PROGRAMME NOW WOULD PRINT ERRONEOUSLY": PRINT : PRINT : PRINT : PRINT : STOP
5690 RETURN
5700 REM ##  RECALL OF SCREEN DISPLAY    (FROM 1310,3410) ###
5710 CLS
5720 GOSUB 5210
5730 CLS
5740 GOSUB 6050
5750 IF E>9 THEN  LET B4=E-6
5760 FOR S=B4 TO E
5770 IF B4>E THEN  GOTO 5910
5780 LET KK$=J$(S): LET Q$="": LET II=0: FOR I=1 TO 10
5790 IF ASC(KK$(I))=0 THEN  LET II=II+1
5800 NEXT I: IF II=0 THEN  GOTO 5820
5810 FOR I=1 TO II: LET Q$=Q$+" ": NEXT I
5820 IF LEFT$(J$(S),2)="C"+NM$(1) THEN  PRINT " !";LEFT$(C$(S),5);"!";NM$;"!  !";RIGHT$(J$(S),8)+Q$;"!";LEFT$(STR$(L(S)),9)
5830 IF LEFT$(J$(S),2)="C"+NA$(1) THEN  PRINT " !";LEFT$(C$(S),5);"!";NA$;"!  !";RIGHT$(J$(S),8)+Q$;"!";LEFT$(STR$(L(S)),9)
5840 IF LEFT$(J$(S),2)="W"+NM$(1) THEN  PRINT " !";LEFT$(C$(S),5);"!";NM$;"!";RIGHT$(J$(S),8)+Q$;"   !";LEFT$(STR$(L(S)),9)
5850 IF LEFT$(J$(S),2)="W"+NA$(1) THEN  PRINT " !";LEFT$(C$(S),5);"!";NA$;"!";RIGHT$(J$(S),8)+Q$;"   !";LEFT$(STR$(L(S)),9)
5860 LET KK$=C$(S): LET Q$="": LET II=0: FOR I=1 TO 10
5870 IF ASC(KK$(I))=0 THEN  LET II=II+1
5880 NEXT : IF II=0 THEN  GOTO 5900
5890 FOR I=1 TO II: LET Q$=Q$+" ": NEXT I
5900 PRINT " !";RIGHT$(C$(S),5)+Q$;"!JNT.!  !        !  !";LEFT$(STR$(M(S)),9)
5910 REM ### (FROM 5770) ####
5920 NEXT S
5930 FOR I=8 TO 22: CSR 38,I: PRINT "!": NEXT I
5940 PRINT
5950 GOSUB 3950: GOSUB 4020
5960 LET R$(1)="          "
5970 GOTO 1860
5980 STOP
5990 REM ## (FROM 1830,2130) ##
6000 FOR I=7 TO 21
6010 CSR 1,I: PRINT "!": CSR 7,I: PRINT "!": CSR 12,I: PRINT "!": CSR 24,I: PRINT "!": CSR 38,I: PRINT "!"
6020 NEXT
6030 RETURN
6040 REM ## (FROM 1810) ##
6050 REM ## (FROM 5740) ##
6060 FOR I=7 TO 22
6070 CSR 0,I: PRINT " !     !   ! !       ! !         !"
6080 NEXT I
```

```
6090 RETURN
6100 REM ## INPUT OF DIVIDED INTEREST     RATES (FROM 2810) AND CALCULATION ##
6110 CSR 2,4
6120 PRINT "Some accounts provide interest of   different values depending on the     Balance level of the account. E.g, the first £100 may provide";
6130 PRINT " no interest(i.e.interest = 0%) and from £100 upward    interest = 7%.   The present programme provides for up to 5 values of interestat ";
6140 PRINT "corresponding credit levels."
6150 CSR 0,15: PRINT "Enter the number of different interest"
6160 PRINT "Values that are provided by the       ";COMP$;" Account at pre-   specified Deposit Levels:-   ";: INPUT "";N2
6170 IF N2>=1 AND N2<=5 THEN  GOTO 6190
6180 PRINT : PRINT "NUMBER UNACCEPTABLE: Try again from 1-5": PAUSE 3000: CLS : GOTO 6110
6190 CSR 0,15: PRINT "                                    "
6200 CSR 0,16: INPUT "Enter Starting Date for operation of   interest rates, including year (FOR    REFERENCE ONLY, NOT >8 CHR$):  ";D$
6210 IF LEN (D$)>8 THEN  CSR 0,18: PRINT "                                                    ": CSR 0,18: PRINT "TOO MANY CHR$": PAUSE 2000: GOTO 6190
6220 IF LEN (D$)=8 THEN  GOTO 6240
6230 FOR I=LEN (D$) TO 7: LET D$=D$+CHR$(32): NEXT I
6240 LET DATE$(ZZ)=D$: LET D$=""
6250 PRINT
6260 FOR I=1 TO N2
6270 CSR 0,15: PRINT "
6280 PRINT "
6290 CSR 0,15: PRINT "Enter the value of Interest rate (as I)paid on deposit level No.";I;: INPUT ":-  ";RATE(ZZ,I): PRINT
6300 IF LEN (STR$(RATE(ZZ,I)))>6 THEN  PRINT "TOO MANY DIGITS !": PAUSE 2000: GOTO 6270
6310 IF N2=1 THEN  PRINT : PRINT "All deposit amounts will attract";RATE(ZZ,I);"%     interest.": GOTO 6330
6320 IF I=N2 THEN  PRINT : PRINT "Deposit amounts above £";RIGHT$(STR$(CAP(I-1)),LEN (STR$(CAP(I-1)))-1);" will          attract";RATE(ZZ,I);"% interest"
6330 IF N2=1 OR I=N2 THEN  LET CAP(I)=999999: GOTO 6360
6340 PRINT "Enter the highest deposit level (to    nearest £) pertaining to an interest of ";RATE(ZZ,I);"% :- ";: INPUT "£";CAP(I)
6350 IF LEN (STR$(CAP(I)))>6 THEN  CSR 0,20: PRINT "                                    ": CSR 0,20: PRINT "TOO MANY DIGITS !": PAUSE 2000: GOTO 6270
6360 PRINT : PRINT
6370 NEXT I
6380 REM ## DISPLAY OF INTERESTS
6390 REM ## AND DEPOSIT LEVELS  ###
6400 FOR I=1 TO 5
6410 IF I=N2 THEN  LET DIS$(I)="ABOVE"
6420 IF I<N2 THEN  LET DIS$(I)="UP TO"
6430 IF I>N2 THEN  LET DIS$(I)=" NONE"
6440 IF I=N2 AND I>1 THEN  LET CAP$(I)=CAP$(I-1): GOTO 6510
6450 IF N2=1 THEN  LET CAP$(I)="0    ": GOTO 6500
6460 LET D$=LEFT$(STR$(CAP(I)),6)
6470 LET D$=RIGHT$(D$,LEN (D$)-1)
6480 IF LEN (D$)=5 THEN  GOTO 6500
6490 FOR Y=LEN (D$) TO 4: LET D$=D$+CHR$(32): NEXT Y
6500 LET CAP$(I)=D$
6510 NEXT I
6520 FOR I=1 TO 5
6530 LET LEV(I)=RATE(ZZ,I)
6542 IF I=N2 THEN  LET Z=RATE(ZZ,I)
6550 IF I>N2 THEN  LET LEV(I)=Z
6560 NEXT I
6570 INPUT "   Press <RET)";D$
6580 REM ## DISPLAY FOR INTEREST DATES    AND DEPOSIT LEVELS (FROM 2980) ##
6590 CLS
6600 CSR 0,0: PRINT "-------------------------------------"
6610 PRINT "!STARTING!RATE1!RATE2!RATE3!RATE4!RATE5"
6620 PRINT "!DATE FOR!     !     !     !     !"
6630 PRINT "!INTEREST!";DIS$(1);"!";DIS$(2);"!";DIS$(3);"!";DIS$(4);"!";DIS$(5)
6640 PRINT "! RATES  !  £  !  £  !  £  !  £  !  £"
6650 PRINT "! (NET%) !";CAP$(1);"!";CAP$(2);"!";CAP$(3);"!";CAP$(4);"!";CAP$(5)
6660 PRINT "+--------+-----+-----+-----+-----+-----"
6670 IF K$="C" THEN  GOTO 3000
6680 FOR I=1 TO 5
6690 IF I>N2 THEN  LET RATE$(I)="0": GOTO 6720
6700 LET D$=RIGHT$(STR$(RATE(ZZ,I)),LEN (STR$(RATE(ZZ,I)))-1)
6710 LET RATE$(I)=D$
6720 NEXT I
6730 PRINT "!";DATE$(ZZ): CSR 9,7: PRINT "!";RATE$(1): CSR 15,7: PRINT "!";RATE$(2): CSR 21,7: PRINT "!";RATE$(3): CSR 27,7: PRINT "!";RATE$(4);
6740 CSR 33,7: PRINT "!";RATE$(5)
6750 FOR I=1 TO 5: LET RATE$(I)="     ": NEXT I
6760 PRINT "-------------------------------------"
6770 CSR 0,20: INPUT "Press <RET)";D$
6780 CLS
6790 PRINT "       DATES FOR INTEREST PAYMENTS"
6800 PRINT "       --------------------------"
6810 CSR 2,4: INPUT "On how many different dates during   the year are interest payments made    into the Account:  ";N4
6820 DIM PIN$(N4,11)
6830 CSR 0,11: PRINT "  Enter the dates in turn for each     payment -- Dates must be entered as    2 digits followed by 3 Capital letters for the ";
6840 PRINT "first three letters of the     month:  E.g., 03JAN."
6850 FOR I=1 TO N4
6860 CSR 0,(16+I): PRINT " Enter date no.";I;":                    ";: CSR 30,(16+I): INPUT D$: IF LEN (D$)<>5 THEN  GOTO 6860
```



```
6870 IF VAL(O$(1,2)))31 THEN  GOTO 6860
6880 LET PIN$(X,1,5)=O$
6890 LET O$=PIN$(X,3,3)
6900 GOSUB 1590
6910 LET PIN$(X,6,3)=P$
6920 LET PIN$(X,9,3)=Q$
6930 NEXT X
6940 RETURN : REM ### (TO 2980) ##
6950 REM ### INPUT OF AUTO-PAYMENTS       (FROM 3400) ####
6960 CLS : PRINT "  REGULAR PERIODIC STANDING ORDERS AND ": PRINT "  ----------------------------------"
6970 PRINT "      AUTOMATIC CREDITS & DEBITS": PRINT "     -------------------------": PAUSE 1000
6980 PRINT : PRINT : PRINT
6990 INPUT "   If none are required, Enter '0':    Otherwise press (RET) ";AP$: IF AP$="0" THEN  RETURN
7000 PRINT
7010 PRINT
7020 PRINT "   If Standing Orders or Auto-credits  to be repeated indefinitely every      CALENDAR Month are required, Enter how many there are to be ";
7030 PRINT "of such sequences (If none, Enter '0'): ";
7040 INPUT NM: IF NM=0 THEM  GOTO 7270
7050 DIM MS$(NM,17)
7060 CLS
7070 FOR I=1 TO NM
7080 PRINT "  Particulars for MONTHLY Transaction  number";I;"( of";NM;")"
7090 PRINT
7100 PRINT "  Enter the required DAY NUMBER in the month (Note that the highest Day Numberto include every month is 28): ";
7110 INPUT O$: IF VAL(O$))31 THEM  GOTO 7110
7120 IF LEN (O$)=1 THEN  LET O$="0"+O$
7130 LET MS$(I,1,2)=O$
7140 PRINT
7150 INPUT "  If required, Enter up to 5 charactersto identify the entry: ";REF$
7160 IF LEM (REF$))5 THEM  GOTO 7150
7170 LET MS$(I,13,5)=REF$
7180 PRINT : INPUT "  Enter 'C' (Credit) or 'W' (Withdravalor Debit):  ";O$: IF O$()"C" AND O$()"W" THEN  GOTO 7180
7190 LET MS$(I,3,1)=O$
7200 PRINT : PRINT "  Enter '";NM$(1);"' (";NM$;") or '";NA$(1);"' (";NA$;"):  ";: INPUT O$: IF O$()NM$(1) AND O$()NA$(1) THEN  GOTO 7200
7210 LET MS$(I,4,1)=O$
7220 PRINT : INPUT "  Enter Sterling Amount--                (Not ) 8 Numerics):  £";O$: IF LEN (O$))8 THEN  GOTO 7220
7230 LET MS$(I,5,8)=O$
7240 PRINT "-----------------------------------": PAUSE 800
7250 NEXT I
7260 CLS
7270 PRINT "  If WEEKLY or MULTI-WEEKLY Entries,   repeated on the same day of the week   each time, are required, ";
7280 INPUT "then Enter thenumber of such sequences             (If none, Enter '0')  ";NW: IF NW=0 THEN  GOTO 7300
7290 GOSUB 7580
7300 CLS
7310 PRINT "  If ANNUAL Standing Orders or Auto-   Credits (to be repeated on an Annual   basis) are required, Input total numberof separate entries.": PRINT
7320 PRINT "  QUARTERLY (requiring 4 entries each) and HALF-YEARLY (requiring 2 entries   each) are to be included here."
7330 INPUT "(If none, Enter '0'):  ";NA: IF NA=0 THEN  GOTO 500
7340 PRINT : PRINT
7350 DIM AS$(NA,26)
7360 FOR I=1 TO NA
7370 PRINT : PRINT "  Input particulars for each ANNUAL    Auto-Instruction in turn.            Present transaction number=";I;"( of";NA;")"
7380 PRINT : INPUT "  Enter the day number in the month: ";O$: IF VAL(O$))31 THEN  GOTO 7380
7390 IF LEN (O$)=1 THEN  LET O$="0"+O$
7400 LET AS$(I,1,2)=O$
7410 PRINT : INPUT "  Enter name of month (first 3 letters only, in Capitals) for required AUTO   Entry:  ";O$: IF LEN (O$)()3 THEN  GOTO 7410
7420 LET AS$(I,3,3)=O$
7430 PRINT : INPUT "  If required, Enter up to 5 charactersto identify the entry: ";REF$
7440 IF LEN (REF$))5 THEN  GOTO 7430
7450 LET AS$(I,22,5)=REF$
7460 PRINT : INPUT "Enter 'C' (Credit) or 'W' (Withdrawal or Debit):  ";O$: IF O$()"C" AND O$()"W" THEN  GOTO 7460
7470 LET AS$(I,6,1)=O$
7480 PRINT : PRINT "Enter '";NM$(1);"' (";NM$;") or '";NA$(1);"' (";NA$;"):  ";: INPUT O$: IF O$()NM$(1) AND O$()NA$(1) THEN  GOTO 7480
7490 LET AS$(I,7,1)=O$
7500 PRINT : INPUT "Enter Sterling Amount--                (Not ) 8 Numerics):  £";O$: IF LEN (O$))8 THEN  GOTO 7500
7510 LET AS$(X,8,8)=O$: LET O$=AS$(I,3,3)
7520 GOSUB 1590
7530 LET AS$(I,16,3)=P$: LET AS$(I,19,3)=Q$
7540 PRINT "-----------------------------------": PAUSE 800
7550 NEXT I
7560 GOTO 500
7570 CLS
7580 REM ## WEEKLY & MULTI-WEEKLY        ENTRIES  (FROM 7290) ##
7590 DIM WS$(NW,19)
7600 CLS
7610 FOR I=1 TO NW
7620 PRINT "  Multiple weekly entry no.";I;" (of";NW;")."
7630 PRINT
7640 INPUT "  Enter required number of weeks withinthis cycle (e.g., Enter '4' for a      4-weekly repetition):-  ";O$: LET WS$(X,13,2)=O$
```

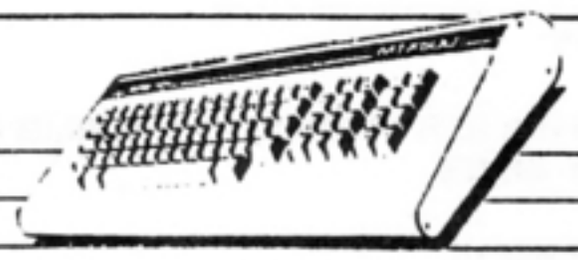


```
7650 PRINT
7660 PRINT "  Enter day number for start of first  entry in ";STMON$(1,3);":  ";: INPUT D$: IF VAL(D$)<1 OR VAL(D$)>31 THEN  GOTO 7660
7670 IF LEN (D$)=1 THEN  LET D$="0"+D$
7680 LET WS$(X,1,2)=D$
7690 PRINT
7700 INPUT "  If required, Enter up to 5 charactersto identify the entry: ";REF$
7710 IF LEN (REF$)>5 THEN  GOTO 7700
7720 LET WS$(X,15,5)=REF$
7730 PRINT : INPUT "  Enter 'C' (Credit) or 'W' (Withdravalor Debit):  ";D$: IF D$<>"C" AND D$<>"W" THEN  GOTO 7730
7740 LET WS$(X,3,1)=D$
7750 PRINT : PRINT "  Enter '";NM$(1);"' (";NM$;") or '";NA$(1);"' (";NA$;"):  ";: INPUT D$: IF D$<>NM$(1) AND D$<>NA$(1) THEN  GOTO 7750
7760 LET WS$(X,4,1)=D$
7770 PRINT : INPUT "  Enter Sterling Amount--                    (Not > 8 Numerics):  £";D$: IF LEN (D$)>8 THEN  GOTO 7770
7780 LET WS$(X,5,8)=D$
7790 PRINT "-------------------------------------": PAUSE 800
7800 NEXT X
7810 RETURN
7820 REM ## REVIEW OR CHANGING OF       'AUTO-'ENTRIES  (FROM 1350) ##
7830 CLS
7840 IF AP$="0" THEN  GOTO 2930
7850 PRINT "      REVIEW OF 'AUTO' ENTRIES"
7860 PRINT "      ----------------------"
7870 PRINT "MONTHLY  ENTRIES:"
7880 PRINT : PRINT : PRINT "Serial","Date","Entry","Amount": PRINT "No.",,,"  £": PRINT
7890 IF NM=0 THEN  PRINT ,,"None": PAUSE 4000: GOTO 7980
7900 FOR I=1 TO NM
7910 PRINT I,MS$(I,1,2),MS$(I,3,2),MS$(I,5,8)
7920 NEXT I
7930 PRINT : PRINT "To alter the sterling amount, Enter the indicated SERIAL No.: If no alteration is required, Enter '0'."
7940 INPUT O: IF O=0 THEN  GOTO 7980
7950 IF O>NM THEN  GOTO 7940
7960 INPUT "       Enter the altered value for the sterling amount:-  £";S$
7970 LET MS$(O,5,8)="        ": LET MS$(O,5,8)=S$
7980 CLS : PRINT
7990 PRINT "WEEKLY or MULTI-WEEK 'AUTO' ENTRIES:"
8000 PRINT : PRINT : PRINT "Serial","Weekly","Entry","Amount": PRINT "No.","Cycle",,"  £": PRINT
8010 IF NW=0 THEN  PRINT ,,"None": PAUSE 4000: GOTO 8110
8020 FOR I=1 TO NW
8030 PRINT I,WS$(I,13,2),WS$(I,3,2),WS$(I,5,8)
8040 NEXT I
8050 PRINT "To alter the sterling amount, Enter the indicated SERIAL No.: If no alteration is required, Enter '0'."
8060 INPUT O: IF O=0 THEN  GOTO 8110
8070 IF O>NW THEN  GOTO 8060
8080 PRINT
8090 INPUT "       Enter the altered value for the sterling amount:-  £";S$
8100 LET WS$(O,5,8)="        ": LET WS$(O,5,8)=S$
8110 CLS : PRINT
8120 PRINT "ANNUAL ENTRIES:"
8130 PRINT : PRINT : PRINT "Serial","Date","Entry","Amount": PRINT "No.",,,"  £": PRINT
8140 IF NA=0 THEN  PRINT ,,"None": PAUSE 4000: GOTO 8230
8150 FOR I=1 TO NA
8160 PRINT I,AS$(I,1,5),AS$(I,6,2),AS$(I,8,8)
8170 NEXT I
8180 PRINT "To alter the sterling amount, Enter the indicated SERIAL No.: If no alteration is required, Enter '0'."
8190 INPUT O: IF O=0 THEN  GOTO 8230
8200 IF O>NA THEN  GOTO 8190
8210 INPUT "       Enter the altered value for the sterling amount:-  £";S$
8220 LET AS$(O,8,8)="        ": LET AS$(O,8,8)=S$
8230 GOTO 2930
8240 REM ## AUTO PAYMENT CALCULATIONS     AND DISPLAY PRINTING  (FROM 710,820,   1030) ##
8250 IF ID$="W" THEN  LET AUB=-AUB
8260 IF PER$=NM$ THEN  LET B1=B1+AUB: LET BX=B1: LET B3=B3+AUB
8270 IF PER$=NA$ THEN  LET B2=B2+AUB: LET BX=B2: LET B3=B3+AUB
8280 LET B1=INT(B1*100+SGN(B1)*0.49)/100: LET B2=INT(B2*100+SGN(B2)*0.49)/100: LET B3=INT(B3*100+SGN(B3)*0.49)/100
8290 LET AUB=INT(AUB*100+SGN(AUB)*0.49)/100: LET BX=INT(BX*100+SGN(BX)*0.49)/100
8300 FOR Y=5 TO 1 STEP -1: IF ASC(REF$(Y))=0 THEN  LET REF$(Y)=CHR$(32) ELSE  GOTO 8320
8310 NEXT Y
8320 IF ID$="C" THEN  CSR 1,21: PRINT "|";AD$;"|";PER$;"|  |";RIGHT$(STR$(AUB),LEN (STR$(AUB))-1): CSR 24,21: PRINT "|";LEFT$(STR$(BX),9)
8330 CSR 38,21: PRINT "|"
8340 IF ID$="W" THEN  CSR 1,21: PRINT "|";AD$;"|";PER$;"|";RIGHT$(STR$(AUB),LEN (STR$(AUB))-1): CSR 24,21: PRINT "|";LEFT$(STR$(BX),9): CSR 38,21: PRINT "|"
8350 CSR 0,23: PRINT
8360 CSR 1,21: PRINT "|";REF$;"|JNT.|  |        |   |";LEFT$(STR$(B3),9): CSR 38,21: PRINT "|"
8370 GOSUB 3760
8380 CSR 0,23: PRINT
8390 LET A(Q+1)=A(Q)
```

```
8400 LET A(Q)=P
8410 LET B$(Q,11)="F"
8420 LET Q=Q+1
8430 LET R$(I)=AD$+REF$: LET F$=ID$+PER$(I)+RIGHT$(STR$(AUB),LEN (STR$(AUB))-1): GOSUB 5610
8440 IF Q=K-1 THEN  GOTO 5130
8450 RETURN
8460 REM ### (FROM 1320) ###
8470 CLS : CSR 1,2: PRINT "In order to provide more reliable     Saving, Verifying & Loading, use the   following procedure:": PRINT
8480 PRINT "1. Clear Screen (Enter 'CLS')": PRINT : PRINT "2. Enter 'SAVE ";CHR$(34);CHR$(34);" '  (with Title, if   required)": PRINT
8490 PRINT "3. After Saving, Enter 'VERIFY ";CHR$(34);CHR$(34);" '    (with same Title, or no Title)": PRINT
8500 PRINT "4. After Loading, always re-start      programme with 'GOTO 10'.  If 'RUN' is used, all variables will be lost."
8510 STOP
      1 REM ########## THIS LISTING UP-DATES 'SHARED BANK ACCOUNT (AUG.1985)'  Ref:  MEMOPAD,Vol.10,No.3,p.86  TO  PROVIDE  DEC.1985 VERSION.  ##################
      2 REM ########  IT LISTS AMENDMENTS AND ADDITIONS SUPPLEMENTARY TO  THE  MAIN  PROGRAMME.  ##########################################################
      3 REM ##   THIS PROGRAMME IS NON-ACTIVE AND WILL NOT RESPOND TO 'RUN'   ##########################################################################
      4 REM ####    INSTRUCTIONS:  LOAD MAIN PROGRAMME THEN TYPE IN ALL FOLLLOWING  LINES FROM  No. 10 (incl.) ONWARD  ##################################
      5 REM ## N.B. MODDY INTRODUCTION UNDER 'LIMITATIONS'CONTAINS ERROR; QUOTE LINE 1450 SHOULD BE LINE 1410 ##########################################
      6 REM ##################################################################################################################################
     10 REM ### 'SHARED BANK ACCOUNT' ###        ### MEMOTECH MTX512 (Dec.1985) ###
    405 IF INT3<0 THEN  LET INT3=0
   1200 REM   [ DELETE ] i.e.,Type: 1200
   1265 PRINT "  I","  Introductory Summary.": PRINT
   1270 PRINT "  T","  Terminate the Account."
   1280 PRINT : PRINT " (N)","  Start NEW Account-"
   1310 IF K$="A" OR K$="a" THEN  GOTO 5700
   1320 IF K$="S" OR K$="s" THEN  GOTO 8470
   1330 IF K$="L" OR K$="l" THEN  GOTO 4160
   1340 IF k$="P" OR K$="p" THEN  GOTO 4160
   1350 IF K$="C" OR k$="c" THEN  GOTO 7820
   1355 IF K$="I" OR K$="i" THEN  GOTO 2390
   1360 IF K$="T" OR k$="t" THEN  GOTO 5122
   1370 IF k$="N" OR K$="n" THEN  GOTO 90
   1775 LET FL2=0: LET BB4=0: LET BB5=0: LET BB6=0
   2105 IF FL2=0 THEN  LET BB1=BB1+BB4: LET BB2=BB2+BB5: LET BB3=BB3+BB6: LET FL2=1
   2685 IF LEN (STMON$)<>3 THEN  CSR 20,10: PRINT "          "
   2690 IF LEN (STMON$)<>3 THEN  CSR 20,10: PRINT "          ": GOTO 2680
   2801 CSR 2,4: PRINT "If this account is being Carried     Forward from an earlier series of      entries which included Accrued Interestfor ";NM$;" and ";
   2802 PRINT NA$;" then enter here:"
   2803 PRINT "    (If condition does not apply,           Enter '0' for each)"
   2804 PRINT : PRINT : PRINT "           Amount of accrued interest for ";NM$;": £";: INPUT "";BB4
   2805 PRINT : PRINT "          Amount of accrued interest for ";NA$;": £";: INPUT "";BB5
   2806 LET BB6=BB4+BB5
   2807 CLS : CSR 10,0: PRINT "INTEREST RATES": CSR 10,1: PRINT "--------------"
   3360 IF K$="C" OR K$="c" THEN  GOTO 1160
   4170 CLS : IF K$="L" OR K$="l" THEN  GOTO 4200
   4470 IF K$="P" OR K$="p" THEN  GOTO 4810
   4945 IF LEN (STR$(L(Z)))=9 THEN  LET QQ$=""
   4970 IF LEFT$(J$(Z),2)="C"+NM$(1) THEN  LPRINT " ";LEFT$(C$(Z),5);"";NM$;" ¦         ";RIGHT$(J$(Z),8)+Q$;"  ";LEFT$(STR$(L(Z)),9);QQ$;" ¦         "
   4980 IF LEFT$(J$(Z),2)="C"+NA$(1) THEN  LPRINT " ";LEFT$(C$(Z),5);"";NA$;" ¦         ";RIGHT$(J$(Z),8)+Q$;"  ";LEFT$(STR$(L(Z)),9);QQ$;" ¦         "
   4990 IF LEFT$(J$(Z),2)="W"+NM$(1) THEN  LPRINT " ";LEFT$(C$(Z),5);"";NM$;" ";RIGHT$(J$(Z),8)+Q$;"  ¦         ";LEFT$(STR$(L(Z)),9);QQ$;" ¦         "
   5000 IF LEFT$(J$(Z),2)="W"+NA$(1) THEN  LPRINT " ";LEFT$(C$(Z),5);"";NA$;" ";RIGHT$(J$(Z),8)+Q$;"  ¦         ";LEFT$(STR$(L(Z)),9);QQ$;" ¦         "
   5122 REM ## (FROM 1355) ##
   5124 CLS : PRINT "      TERMINATION OF THIS ACCOUNT."
   5125 PRINT "      ------------------------": PRINT
   5126 PRINT "Individual Balances that apply on last date shown in account are:"
   5128 GOTO 5151
   5150 PRINT "Final Individual Balances that apply onlast date as shown above are:"
   5151 PRINT "     For ";NM$;" = £";B1
   5152 PRINT " and for ";NA$;" = £";B2
   5155 PRINT "Any additional accrued interest equals:"
   5156 PRINT "     For ";NM$;" = £";INT(100*BB1+0.5)/100
   5160 PRINT " and for ";NA$;" = £";INT(100*BB2+0.5)/100
   5170 PRINT "  Write down these Individual Amounts. Press <RET> & then'N'.Re-enter Accrued Interest (if any) when requested. CarryForward the two balances ";
   5180 PRINT "and use '#' asfirst Chr.after date to give immediate entry."
   5190 INPUT "  Press <RET)";O$: GOTO 1160
   5200 REM   [ DELETE ] i.e.,Type: 5200
   6110 CSR 2,2
   6145 PRINT : PRINT " N.B. Interest Accumulation becomes    zero during any period while AVAILABLE Balance is overdrawn."
   6670 IF K$="C" OR K$="c" THEN  GOTO 3000
   6940 RETURN : REM ### (TO 2810) ##
   7660 PRINT "  Enter day number for start of first  entry in ";STMON$(1,3);":  ";: INPUT O$: IF LEN (O$)>2 THEN  GOTO 7660
   7665 IF VAL(O$)<1 OR VAL(O$)>31 THEN  GOTO 7660
   8245 IF FL2=0 THEN  LET BB1=BB1+BB4: LET BB2=BB2+BB5: LET BB3=BB3+BB6: LET FL2=1
```

Syntaxsoft
UNIT 109, GLENFIELD PARK, GLENFIELD ROAD, NELSON BB9 8AR.
TELEPHONE [0282] 698849
TELEX:635480 CENTIM.G. ANSWERBACK SYNTAX

HARDWARE



PRICES EFFECTIVE FROM 1ST DECEMBER 1985

MTX 512 ..... #119.00 MEMBERSHIP EXTRA
MTX 500 ..... # 69.95 MEMBERSHIP EXTRA

32K MEMORY EXPANSION .....#36.74
64K MEMORY EXPANSION ..... #45.43
128K MEMORY EXPANSION ..... #69.52
**** PLEASE STATE FOR WHICH MODEL ******

NEWWORD 32K ON ROM .....#36.74
PASCAL 16K ON ROM .....#36.74
SPECULATOR .....#36.95

DMX80 PRINTER .....#179.95
PRINTER CABLE .....# 8.95

SDX 500K DRIVE + INTERFACE .....#222.50
SDX 1MEG DRIVE + INTERFACE .....#265.83

2ND 250K SDX DRIVE .....# 89.00
2ND 500K SDX DRIVE .....#169.00
2ND 1MEG SDX DRIVE .....#203.00

FDX 2ND DRIVE 500K .....#141.00
FDX 2ND 1MEG DRIVE .....#163.00

DUST COVER .....#3.50
DMX80 PRINTER RIBBON .....#8.98

FLOPPY DISCS (BOX 10) GUARANTEED ..... #18.95
PACE NIGHTINGALE MODEM .....#119.00 + #5.00 P&P
250K DISC DRIVE + INTERFACE .....#199.00 + #5.00 P&P

DISC CASES HOLD 10 DISC ..... # 2.55
FLOPPICLENE DISC CLEANING KIT .....#17.20
ANTISTATIC SCREEN WIPES ...(10)....# 1.50
DISC CABINETS (LOCKABLE) 110 DISCS #36.95

CRIB CARD ..... # 2.16
ROM LISTINGS ..... #45.00
SDX CONTROLLER LISTINGS .#20.00
ROM CALLS INFO SHEET ... 50p
RST 10 CALLS INFO SHEET 50p
INTERRUPTS INFO SHEET 80p
DDT INFO BOOK .....#2.00
VDP TECHNICAL MANUAL ....#5.95
MTX SERVICE MANUAL .....#6.95

UPGRADE PACKAGE 1 ..... £198.00
UPGRADE PACKAGE 2 ..... £223.39
The above require factory fitting so add an extra £25 to cover cost of this service.

80 COL CARD + CPM +NW/SC .....#180.95
80 COL UPGRADE KIT (RS232) .....# 27.00

PACKAGE ONE
SDX 500K DRIVE + INTERFACE
+80 COL BOARD +CPM + NW + SC .....#355.00

PACKAGE TWO
AS ABOVE BUT WITH 1MEG DRIVE .....#395.00

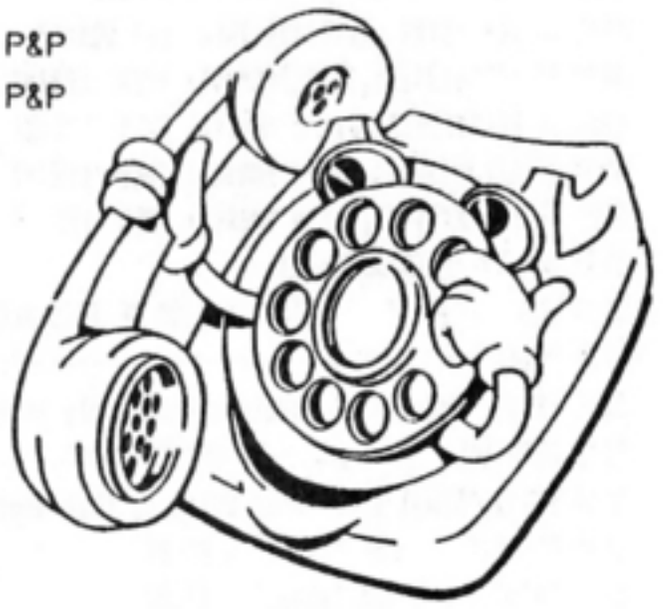
PACKAGE THREE
SDX 500K DRIVE : INTERFACE:
COMPUTER MTX512: 80 COL PCB:CPM .....#449.00

PACKAGE FOUR
AS ABOVE BUT WITH 1MEG DRIVE .....#489.00

FDX SINGLE 500K CPM SYSTEM .....#539.00
FDX SINGLE 1MEG CPM SYSTEM .....#675.00

FDX TWIN 500K CPM SYSTEM .....#569.00
FDX TWIN 1MEG CPM SYSTEM .....#740.00
(REquires RS232 comms board)

SILICON DISCS
500K ..... #145.90
1MEG ..... #441.00



THE ABOVE PRICES ONLY APPLY TO U.K SALES & BFPO SALES
BFPO SHOULD ADD AN EXTRA £30.00 TO COVER ADMINISTRATION BY MEMOTECH

ĐON'T FORGET IF YOU HAVE A DISC DRIVE YOU SHOULD OWN A HIGH QUALITY HEAD CLEANER see FLOPPICLENE
.... over half the problems handled by us are due to dirty disc heads.

SOFTWARE

00106	26X26 SPREAD SHEET	UTIL	SYNT	7.95	I	ANY
00057	3D TACHYON FIGHTER	ARC	CONT	6.95	I	ANY
00135	9 ELECTRIC. PROGS	EDUC	SSFT	13.95	I	ANY
00062	ADVENTURE QUEST	ADV	LVL9	8.75	I	ANY
00033	AGROVATOR	ARC	SYNT	5.95	I	512
00125	AIRCRAFT MAGNETISM	FLGT	AVTN	?	I	ANY
00120	AIRCRAFT PAYLOADS	FLGT	AVTN	?	I	ANY
00122	AIRLAW 2	FLGT	AVTN	?	I	ANY
00123	AIRSPEED INDICATOR	FLGT	AVTN	?	I	ANY
00071	ALICE IN WONDER.	ADV	CONT	6.02	I	ANY
00121	ALTIMETER	FLGT	AVTN	?	I	ANY
00008	ASTROMILON	ARC	CONT	6.02	I	ANY
00047	ASTROPAC	ARC	CONT	6.02	I	ANY
00058	BACKGAMMON	BRD	CONT	7.95	I	ANY
00041	BASIC BUSINESS	BS	CONT	5.95	I	ANY
00043	BLOBBO	ARC	CONT	6.02	I	ANY
00073	BOUNCING BILL	ARC	SYNT	4.95	I	ANY
00074	BRIDGE	CARD	CONT	6.95	I	512
00130	BUSINESS GAME	BRD	SSFT	15.95	I	ANY
00077	CANVAS	UTIL	CONT	7.95	I	ANY
00085	CAVES OF ORB	ADV	SYNT	5.95	I	512
00137	CESIL INTERPRETER	LANG	SSFT	5.95	I	ANY
00094	CHAMBEROIDS	ARC	MEGA	5.95	I	ANY
00059	CHESS	BRD	CONT	8.75	I	ANY
00053	COBRA	ARC	CONT	6.02	I	ANY
00025	COLOSSAL ADVENTURE	ADV	LVL9	8.75	I	ANY
00098	COMBAT	ARC	PANS	2.95	I	512
00028	COMPOSER	UTIL	XAV	13.00	I	ANY
00046	CONT RAIDERS	ARC	CONT	6.02	I	ANY
00099	CRIBBAGE	CARD	SCRP	2.95	E	ANY
00110	CRYSTAL	ARC	MEGA	5.95	I	ANY
00050	DEN.GOES BANANAS	ARC	SCRP	2.95	I	ANY
00011	DENNIS & CHICKEN	ARC	SCRP	2.95	I	ANY
00103	DENNIS AND CIRCUS	ARC	SCRP	2.95	I	ANY
00068	DOODLEBUG	ARC	SYNT	4.95	I	ANY
00108	DOWNSTREAM DANGER	ARC	MEGA	5.95	I	ANY
00096	DR. FRANKIE	ARC	SYNT	5.95	I	512
00056	DRAUGHTS	BRD	CONT	6.95	I	ANY
00111	DRIVE THE CEE 5	ARC	MEGA	5.95	I	ANY
00063	DUNGEON ADVENTURE	ADV	LVL9	8.75	I	ANY
00067	EDASM	UTIL	SYNT	7.95	I	512
00066	EMERALD ISLE	ADV	LVL9	5.95	I	ANY
00038	ESCAPE FROM ZARKOS	ARC	MEGA	5.95	I	ANY
00081	EXTENDED BASIC	6.95	SENT	6.95	I	ANY
00082	FATHOMS DEEP	ARC	MEGA	5.95	I	ANY
00090	FIG FORTH	LANG	SYNT	15.75	I	512
00055	FIREHOUSE FREDDIE	ARC	CONT	6.02	I	ANY
00021	FIRST LETTERS 1	EDUC	CONT	8.75	I	ANY
00092	FKEY DEFINER	UTIL	MEMB	6.95	I	ANY
00037	FLUMMOX	ARC	SYNT	5.95	I	512
00132	FRACT, PERCENTAGES	EDUC	SSFT	5.95	I	ANY
00052	GAUNTLET	ARC	CONT	6.02	U	ANY
00102	GHOSTLY CASTLE	ADV	PANS	2.95	I	ANY
00031	GOLDMINE	ARC	CONT	6.02	I	ANY
00069	GRAPHICS	UTIL	CONT	5.95	I	ANY
00087	H & L DUMP	UTIL	MEM	4.95	I	ANY
00072	HAWKWARS	ARC	SYNT	4.95	I	ANY
00065	HELI-MATHS	EDUC	CONT	5.95	I	ANY
00034	HUNCHY	ARC	SYNT	4.95	I	ANY
00083	ICEBERG	ARC	SYNT	4.95	I	ANY
00105	JET SET WILLY	ARC	SPRJ	6.95	I	ANY
00015	JOHNNY REB	WAR	LOTH	6.02	I	ANY
00097	JUMPING JACK FLASH	ARC	SYNT	5.95	I	512
00115	KARATE KING	ARC	MEGA	5.95	I	ANY
00016	KEY TO TIME	ADV	LUPP	6.02	I	ANY
00042	KILOPEDE	ARC	CONT	6.02	I	ANY
00019	KNUCKLES	ARC	CONT	7.95	I	ANY
00078	LES FLICS	ARC	PSS	6.95	E	ANY
00032	LITTLE DEVILS	ARC	SYNT	4.95	I	ANY
00024	LORDS OF TIME	ADV	LVL9	8.75	I	ANY
00035	M COMMAND & ARCAD.	ARC	SYNT	4.95	I	ANY
00070	MAN FROM GRANNY	ADV	SYNT	4.95	I	512
00104	MANIC MINER	ARC	SPRJ	6.95	I	ANY
00119	MAPS AND CHARTS	FLGT	AVTN	?	I	ANY
00126	MAPS AND CHARTS 1	FLGT	AVTN	?	I	ANY
00022	MATHS 1	EDUC	CONT	8.75	I	ANY
00013	MAXIMA	ARC	CONT	6.02	E	ANY
00086	MEMOCHEQUE	UTIL	SYNT	6.95	I	ANY
00075	MEMOSKETCH	UTIL	SYNT	7.95	I	ANY
00089	MINER DICK	ARC	XAV	6.95	I	ANY
00044	MISSION ALPHATRON	ARC	CONT	6.02	I	ANY
00030	MISSION OMEGA	ARC	SYNT	4.95	I	ANY
00054	MURDER AT MANOR	ADV	LUPP	6.02	I	ANY
00010	MUSIC PAD	UTIL	CONT	6.02	I	ANY
00003	NEMO	ARC	CONT	6.00	I	ANY
00131	NETWORK LOADER	UTIL	SSFT	8.95	I	ANY
00112	OBLITERATION ZONE	ARC	MEGA	5.95	I	ANY
00045	OBLOIDS	ARC	CONT	6.02	I	ANY
00129	PAINTBOX	UTIL	SYNT	5.95	I	ANY
00001	PAYROLL	UTIL	CONT	21.25	I	512
00005	PHAID	ARC	CONT	6.02	I	ANY
00061	PHYSICS 1	EDUC	CONT	8.75	I	ANY
00124	PILOT NAVIGATION	FLGT	AVTN	?	I	ANY
00012	PONT & BLACKJACK	CARD	CONT	6.02	I	ANY
00009	POT HOLE PETE	ARC	CONT	6.02	I	ANY
00040	PURCHASE LEDGER	BN	CONT	12.75	I	512
00048	QOGO	ARC	CONT	6.02	I	ANY
00076	QOGO 2	ARC	MEGA	5.95	I	ANY
00095	QUANTUM	ARC	SYNT	5.95	I	ANY
00109	QUAZZIA	ARC	MEGA	5.95	I	ANY
00107	RED MOON	ADV	LVL9	?	U	ANY
00127	RELATIVE VELOCITY	FLGT	AVTN	?	I	ANY
00064	RETURN TO EDEN	ADV	LVL9	8.75	I	ANY
00020	REVERSI	BRD	CONT	7.95	I	ANY
00114	ROLLA BEARING	ARC	MEGA	5.95	I	512
00100	RUTHLESS BASTARD	ARC	LSFT	2.50	I	512
00002	SALES LEDGER	UTIL	SYNT	15.75	I	512
00029	SALTY SAM	ARC	SYNT	4.95	I	ANY
00113	SEPULCRI SCELERATI	ARC	MEGA	5.95	I	512
00101	SLOOPY'S CHRISTMAS	ARC	PANS	2.95	I	ANY
00116	SMG	ARC	MEGA	5.95	I	ANY
00049	SNAPPO	ARC	CONT	6.02	I	ANY
00023	SNOWBALL	ADV	LVL9	8.75	I	ANY
00036	SON OF PETE	ARC	MEGA	5.95	I	ANY
00136	SOUND & RESISTORS	EDUC	SSFT	5.95	I	ANY
00026	SPELLI-COPIER	EDV	CONT	5.95	I	ANY
00080	SPOOLER	UTIL	MEM	4.95	I	ANY
00017	STAR COMMAND	ARC	CONT	6.95	I	ANY
00014	SUPA CODER	UTIL	SYNT	7.95	I	ANY
00084	SUPER BIKE	ARC	SYNT	4.95	I	ANY
00004	SUPER MINEFIELD	ARC	CONT	6.02	I	ANY
00093	SURFACE SCANNER	ARC	MEGA	5.95	I	ANY
00133	SYMMETRY & GLASS	EDUC	SSFT	5.95	I	ANY
00039	TAPE TO DISC	UTIL	MEM	6.95	I	ANY
00007	TAPEWORM	ARC	CONT	6.02	I	ANY
00088	TARGET ZONE	ARC	SYNT	6.95	I	ANY
00118	THE DESIGNER	UTIL	HALT	8.95	I	ANY
00128	THE WALL	ARC	SYNT	4.95	I	512
00051	THE ZOO GAME	ADV	CONT	6.02	I	512
00134	TITRATION,CHROMATO	EDUC	SSFT	5.95	I	ANY
00006	TOADO	ARC	CONT	6.02	I	ANY
00018	TURBO	ARC	CONT	6.95	I	ANY
00117	USER BASIC	UTIL	SYNT	8.95	I	ANY
00079	USER EXTEND	UTIL	MEM	7.95	I	ANY
00027	UTILITIES 1	UTIL	CONT	4.95	I	ANY
00091	VERNON & VAMPIRES	ARC	SYNT	5.95	I	ANY
00138	WOOD SIMULATION	EDUC	SSFT	5.95	I	ANY
00060	WORD & PICTURE	EDUC	CONT	8.75	I	ANY

PLEASE ONLY ORDER THOSE MARKED " I "

KEY:- STOCK NUMBER TITLE TYPE HOUSE MEMBER PRICE STOCK MACHINE

E=EXPECTED SOON I = IN STOCK U = UNAVAILABLE AT PRESENT