SERVICE -
THE SAGA CONTiNUES

My Minisette cassette-recorder was repaired, but not - alas -
in one week, nor even in two. TANDY explained that it had to
go to the Midlands: whereas computers are mended in London,
and had it been a computer it would probably have been back in
a week. How this may be, I do not know, since the effect of
marking it URGENT was that they forgot for a week to send it
to the repair workshop. However , although I cannot pay TANDY
the tribute I had hoped,for streamlined efficiency, I can
at least admire their unfailing courtesy, which survived a
torrent of frantic enquiries from your editor. Perhaps my
favorite comment on the whole affair came from a junior member
of TANDY's staff, who happily responded to an anxious enquiry
with the classic words: "I haven't got a clue!"

* * * * * * *

While SHARP solemnly announce that details of machine-code will
be released "this year, next year, sometime, never ... " other
exciting things are happening in the world of the PC 1500. On another
page of this issue the first of a series of articles introducing
assembly language will break the ice, and on page 41 a reader
in Istanbul describes how he now has 28K.

* * * * * * *

Barry ELKAN reports an interesting use of the PC 1500 in the USA.
On the "Peoples Express" - a bus-stop flight between Washington
and New York - tickets are issued in flight on the PC 1500,
using specially preprinted rolls of paper. He is at present
investigating the possibilities of making available,in this country,
the various add-ons produced overseas. If he is successful,
we hope to bring you the first news.

* * * CONTENTS * * *

Istanbul, May 9th, 1983

Dear Sirs,

RE: Monthly Newsletter !STATUS-1500'

In case this may be of interest to the newsletter readers:

My PC-1500 just recently underwent a bit of internal surgery
in W.Germany.Within the PC-1500 I now have available a BASIC
working area of 28,475 Bytes,this with the module box still
free.An additional module inserted may no longer be used for
BASIC memory add on.However,it can be used to store machine
code programs.For example was my CE-159 very slightly modified
thus enableing me to store a built in control program,the
RVS-Hexmonitor plus 'TOOL 1' which gives me access to 11 new
BASIC commands mainly aimed at programming and editing in
modules.(APPEND,RENUMBER,CHANGE,DELETE,ERASE,FIND,KEEP etc.)
In the very near future 'TOOL 2' will come out with the most
helpful features FSAVE,FLOAD,FMERGE securing 25-fold speed in
all tape operations.
Above programs I have secured within the modified CE-159 still
leaving me free access for my own machine programs between
&9800 - &9FFF.

The most exciting thing about the modification of my PC-1500
was the fact that I had the machine back within a week with
everything working the way it was announced!Really a rarity
these days!

Yours faithfully

Hans-Henning Reine

HEX & DEC

The program on the left will turn a series of hex numbers into
decimal figures; the & symbol is not necessary. More usefully,
the program on the right turns a series of decimal numbers into hex.

```
10: INPUT A$
15: S=0
20: FOR F=0TO 3
30: R$=MID$ (A$, 4-
    F, 1)
32: IF ASC R$>64
    GOTO 100
34: R=VAL R$
35: R=R*16^F
40: S=S+R
50: NEXT F
60: LPRINT A$;S
70: GOTO 10
100: R=ASC R$-55
101: GOTO 35
```

```
10: INPUT "dec ";D
15: U=1+2*(D>255)
20: LPRINT D;" ";
30: FOR L=UTO 0
    ,STEP -1
40: J=INT (D/16^L)
50: K$=CHR$ (J+55*
    (J>9)+48*(J<10
    ))
55: LPRINT K$;
60: D=D-J*16^L
70: NEXT L
80: LPRINT
90: CLEAR :GOTO 10
```

41

Here is a handy little routine for taking a look at the contents of
memory, without yards of printout. Pressing any of the number keys will
alter the WAIT time (wait for the BEEP). Pressing the CLEAR key will
back up by steps of 10, and the MODE key will go forward by steps of 10.
Any other key will make the WAIT time too long to be useful.

```
  2: ON ERROR GOTO 100
 10: INPUT "start-"; F
 15: WAIT 0
 20: A$=INKEY$: PRINT F; PEEK F;" ";CHR$ PEEK F
 24: F=F+1
 25: GOTO 20+ASC A$
 33: GOTO 20
 44: F=F-12: BEEP 1,20,900: GOTO 20
 51: F=F+9: BEEP 1, 50, 500: GOTO 20
100: WAIT 10*ABS (ASC A$-48):BEEP 7: GOTO 20
```

Addresses 30286 and 30287 hold the display indicators and the modes
themselves. If you POKE 30286, 127, 127 you will get all the indicators
(except BUSY) lit up. For RUN mode, POKE 30287, 67. For PRO mode
POKE 30287, 35. And if you want to experiment with *combined* RUN and PRO
modes, POKE 30287, 99. But 67 and 35 make 102: why then 99 ?

Here we must think of the binary equivalents. 127 would be 1111111.
67 would be 1000011. 35 would be 0100011. Imagine these noughts and ones
as a row of switches: 1 for on, 0 for off. So to combine the 'on' switches
of 67 and 35 gives 1100011, and this is 99.

You can work out for yourself the exact numbers for each mode and
indicator. Take 30286 and 30287 separately, and into each POKE successively
0, 1, and ascending powers of 2, +1. (i.e. 2,3,5,9 etc). In case you
forgot to note the original contents of these addresses before changing
them, the normal display may be restored by POKE 30286,65,67.

A practical use for the POKE technique occurs with Mike O'Regan's
PiCoWRITER program on page 44 .He advises the user to prime the Reserve
keys and Reserve template. The following addition to the program would
enable you to do this priming in 1 second:

```
400: "V" POKE 14422, 1,241,146,51,48,64,2,241,146,57,48,64,3,241,
       146,49,50,48,64
410: POKE 14441,4,241,146,49,49,48,64,5,241,146,50,54,48,64,6,241,146,
       50,55,48,64,0
420: POKE 14344,32,87,82,32,32,82,66,32,32,69,68,32,32,80,82,32,32
       87,84,32,32
430: POKE 14365,82,84,0: BEEP 3
```

Two other useful addresses are concerned with text graphics. 31218 holds
the ROTATE value (normally 0). By POKEing values 0 to 4 into this address
you can ROTATE in TEXT mode. The only use for this is to enable you to
linefeed even if ROTATEd. This could perhaps be used if you wanted to
write some lines of text upside down! However you still cannot LLIST
when ROTATEd. Address 31220 holds the CSIZE value. Into this you can
POKE any number up to 255, and you will get characters of that size.
The maximum practical value for full-size characters is CSIZE 54,
or CSIZE 36 in ROTATE 1 or 3.

[to be continued]

IAN TRAYNOR signals that he tried SAFECRACKER (April 1983) on his
EASICALC and EASIFILE programs, and found that the hidden secrets
were not uncovered: on the contrary, SAFECRACKER cracked up!

*The EDITOR is distressed to have to inform Mr. TRAYNOR that he must
have made an error in entering SAFECRACKER, since it has been used
successfully to lay bare his ingenious method of protection.*

RAYMOND MINTOFF writes from Malta to ask where he can obtain the TANDY
products mentioned in the March issue, and where he can get a SHARP
service manual.

*For TANDY, write to   TANDY COMPUTER CENTRE,
                         20-21 ST.GILES HIGH STREET
                         CENTREPOINT, LONDON W.C.2
The manager is IAN SANDILANDS, and he will be happy to post to Malta.*

*For the service manual, write to   A.MORAN
  [there are 2 manuals, 1 for]   SHARP ELECTRONICS (UK) LTD
  [the CE 150, the other for]   THORP ROAD, MANCHESTER M10 9BE
  [the printer!!!   ]*

JOHN MACK, and others, ask how they can avoid the Linefeed, etc, which
occurs when the PC1500 is turned on.

*If when you turn the machine off, instead of using the OFF key, you
CALL 58175 , when you turn on again you will avoid all the whirring.
You will get the BUSY signal , which can be killed by touching any
key except DEF and SHIFT. (Information from IAN TRAYNOR).
WARNING: This method may temporarily disable the "Auto Power Off"*

ROY WRIGGLESWORTH asks for a simple program for a mailing list .

*If MAILINGLIST(April 1983) is too complex for your purpose, the
following may be adequate:*

```
10: L$=" "
20: READ A$
30: IF A$=L$ LF 5: GOTO 20
40: LPRINT A$: GOTO 20
50: DATA "M.THATCHER","10 DOWNING STREET","WHITEHALL",L$
60: DATA "MICHAEL FOOT","HAMPSTEAD HEATH",L$
70: DATA "ROY JENKINS","CLOUD CUCKOO LAND", L$
```

R.J.COURT, in New Zealand, has difficulty with the CE153 software board.
When running the CONSTELLATION program, (page 39 of the manual),
only one constellation will print, and whatever he does, he gets
"ERROR 9 IN 70"

*Can any kind reader help? Is there an error in SHARP's program?
It is not unlikely!*

The features are as follows:

WRITE.

enables text to be entered in lines limited to 36 characters. The only forbidden character is the DOUBLE QUOTE (for obvious reasons). Line length is automatically checked as written for length and overlength lines are truncated to the nearest whole word, and the new line displayed to enable text entry to continue at the right place. Text entry is terminated by typing zz (or ZZ).

READBACK

at a reasonable speed (which may be later adjusted as experience improves reading). Lines are numbered to facilitate editing.

EDIT

restricted to WHOLE LINE INSERTION, DELETION or REPLACEMENT. Note that the appropriate key must be HELD DOWN until the option chosen appears. New lines are also subject to truncation if necessary.

PRINTOUT

complete text, in CSIZE 1 with double line spacing for clarity. A BEEP signals the end of each line.

SAVE

to tape. Note that it will attempt to save the whole of the Dimensioned area, but, with experience the tape recorder can be stopped when all text has been saved (by pressing the BREAK KEY). Unlike the majority of micros, it seems that the 1500 does not use TRAILERS for the tape signal. Data may be labelled.

READ

from tape. Reading may also be terminated when all data have been read (listen to the signal!).

RESERVE KEYS

Should be formatted as follows:

```
F1 GOTO 380
F2 GOTO 90R
F3 GOTO 120R
F4 GOTO 110R
F5 GOTO 260R
F6 GOTO 270R
```

then labelled:

```
WR  RE  ED  PR  WT  RT
```

```
1:REM PiCoWRITER
   Mike O Regan
   1983
10:" "CLEAR :DIM
   M$(200)*37:K=1
20:WAIT :PRINT "F
   or MENU press
   RCL
30:WAIT 50:PRINT
   "Line ";K:
   INPUT M$(K)
40:IF M$(K)="ZZ"
   OR M$(K)="zz"
   THEN 20
50:IF LEN M$(K)<3
   7THEN 70
60:N=K:GOSUB 210.
70:IF K=198BEEP 5
80:K=K+1:GOTO 30
90:WAIT 5:FOR L=1
   TO K-1:PRINT L
   ;": ";
100:FOR I=1TO LEN
    M$(L):PRINT
    MID$ (M$(L),I,
    1);:NEXT I:
    PRINT "":NEXT
    L:GOTO 20
110:CSIZE 1:FOR L=
    1TO K-1:LPRINT
    M$(L):LF 1:
    BEEP 1:NEXT L:
    GOTO 20
120:WAIT 80:PRINT
    "I to INSERT.
    D to DELETE":
    PRINT "ANY OTH
    ER KEY to CHAN
    GE
130:D$=INKEY$ :IF
    D$=""THEN 120
140:IF D$="D"THEN
    330
150:IF D$="I"THEN
    280
160:INPUT "Line No
    ? ";N
170:WAIT 10:FOR I=
    1TO LEN M$(N):
    PRINT MID$ (M$
    (N),I,1);:NEXT
    I:PRINT ""
180:INPUT "New Lin
    e? ",M$(N)
190:IF LEN M$(N)<3
    7THEN 20
200:GOSUB 210:GOTO
    20
210:FOR I=37TO 25
    STEP -1:IF
    MID$ (M$(N),I,
    1)=" "THEN 230
```

```
220:NEXT I
230:WAIT 5:M$(N)=
    LEFT$ (M$(N),I
    -1)
240:BEEP 2:PAUSE "
    Line too LONG!
    ":PAUSE "short
    ened to . ."
250:FOR I=1TO LEN
    M$(N):PRINT
    MID$ (M$(N),I,
    1);:NEXT I:
    WAIT 50:PRINT
    "**":RETURN
260:INPUT "TITLE "
    ,T$:PRINT "TAP
    E TO RECORD-EN
    TER":PRINT #T$
    ;K,M$(*):GOTO
    20
270:WAIT 0:PRINT "
    TAPE TO PLAY-E
    NTER":INPUT #T
    $;K,M$(*):GOTO
    20
280:INPUT "Which L
    ine Number? ";
    N
290:K=K+1:FOR I=K
    TO N+1STEP -1:
    M$(I)=M$(I-1):
    NEXT I
300:INPUT "Line to
     INSERT ",M$(N
    )
310:IF LEN M$(N)<3
    6THEN 20
320:GOSUB 210:GOTO
    20
330:INPUT "Line to
     DELETE ";N
340:K=K-1:FOR I=N
    TO K:M$(I)=M$(
    I+1):NEXT I:M$
    (K+1)=""
350:GOTO 20
```

NOTE.. Lines 10 and 20 are scaled for machine with 8K RAM EXPANSION and should be scaled down for those with less memory.

If a herring and a half cost a shilling and a half, how many beans
make five? Plus VAT? And 2½% discount for cash? And if the rate is
altered?

These are the sort of questions PROCOS is designed to answer. I spent 4
hours learning how to make full use of the program: if I have missed
anything I can only say that the average user is not prepared to spend
longer, and will miss it too. Available are up to 255 items,
each nameable; and to each you may ascribe a value, which can be
either a number, or an arithmetical expression. After interrupting
the program, UNLOCK, go into PRO mode, and write anything up
to 3½K of program which will calculate all the results you want.
Go back into RUN mode, and calculation can be automatic or not , as
preferred. And if you wish to see a variation in the result, you
can enter this as an arithmetical expression on the result line.

The instruction manual is in two halves. The second half contains
a detailed list of commands and their uses, the structure of the
program, and the contents of all variables used. The first half ,
however, I found sketchy and inadequate.There were also one or two
vital errors, though I understand that these will be corrected.
The manual includes the advice "Feel free to experiment!" which
I always take to be a polite way of saying "*I cannot be bothered
to explain it, so work it out for yourself!*" The biggest snag is
the fact that when a result is varied by an arithmetical expression
there is no record or trace of this: so it is impossible to tell
whether the result shown is the original, or the varied result.

I can see many uses for this program: tax at varied rates, profit
on varying quantities, or any hypothetical financial or statistical
use. But to get any substantial benefit from the program, an
ability to program is essential. And this raises yet again the query:
if you have the programming techniques, why do you need the ready-
made program? Of course, the print options and display facilities
of PROCOS are excellent, and it does provide a framework for your
calculations: the only programming you need to do is for the actual
calculations: and as most experienced programmers will agree, up
to 80% of any program is occupied by 'housekeeping' and display.
PROCOS will do all this for you.

PROCOS, by MiCROL, is distributed by MICROMAIL of Cambridge,£18.95

MINDBOGGLE CORNER
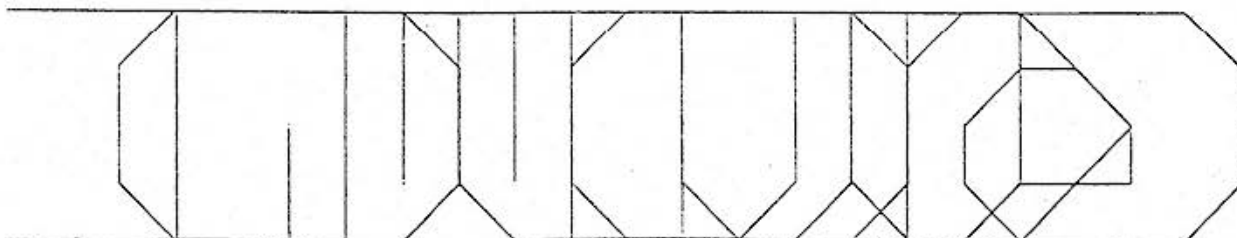
Is it the plan of a spaceship? Or a train? Or an illustration of the
mating habits of the Trobriand islanders? But is it the right way up?
Indeed, *is* there a right way up? Discover the secret, and you will have
no difficulty in finding the very concise program that creates this
apparently random pattern. There is a clue on page * of this issue,perhaps.
Super prize. Closing date July 15 - to give a chance for overseas subscribers
to enter. (We now have readers in 19 countries).

To those of you who, like myself, cannot even spell memonic memomnmic memnonic menmonmic mnemnomnic reminder, the concepts of machine code programming are bewildering and intimidating. We look upon machine code programmers as beings from another world, who probably even dream in binary, and talk in hex.

It is not really quite as bad as all that. Machine code is not easy; it requires absolute accuracy, patience and hard work. The information we are publishing will not turn you into a machine-code-programmer overnight. Much of it will not be comprehensible to beginners´- or to myself . But we hope to print during the coming months some useful subroutines in assembler: they may be used intact: but it is helpful to have at least a vague idea what one is doing.

Your computer works in binary, in ONs and OFFs, 1s and 0s.It does not really understand ordinary decimal numbers at all. Hex, in base 16, and 256ary, are a step in the direction of these binary operations. BASIC is an 'interpreted' language. This means that every time a program is run, every single statement is interpreted into binary each time. For instance if you had the line
    10: IF A=B GOTO 50
your computer must find the address of A, do the same with B, find their values, compare them, find the meaning of IF, find line 50, now find its correct address, and move the pointer to that address. Obviously much time may be saved if you can instruct the computer to fish out the contents of a specific address, and another, compare them, and then go directly to the address required by the condition. No quicker to write, but quicker for the CPU to do. You may think that a millisecond here or there does not matter very much: but it certainly does for repetitive operations. Who would not prefer to sort a few hundred numbers in 20 seconds instead of 10 minutes? Another advantage of machine code is that you are not bound by the limits of your BASIC vocabulary. An IF....ELSE instruction is for instance possible, and many others.

The earliest programmers were obliged to write in binary, a task few of us could cope with today. Strictly speaking, this is what pure 'machine code' really is. But we are able to write in hex: and these base 16 numbers are easily converted by the machine into binary. The CPU performs certain operations when fed with certain 'operation codes' in hex, when previously alerted to do so. It is alerted by the statement CALL : which means "go to the address mentioned, and treat the contents of that and the following addresses as direct operation code instructions: obey them, and return to the BASIC program when told to do so".

But even machine-code-programmers are human, and a series of op-code numbers is unreadable by humans: nor is it feasible to translate machine code programs written for one machine into the codes required by another. (Like translating from a foreign language you don´t know with the aid of a dictionary). But ASSEMBLY LANGUAGE (all those LDAs and LDIXs) is a sort of Esperanto. They remind you what you are doing, and can be read. They differ from chip to chip, but have a core of unanimity. On some machines you can actually program in these, and your machine will substitute the appropriate codes: but not on the PC 1500 as yet, if ever.

I hope that for beginners the above will give some idea what it is all about. I also hope that the experienced programmers who notice many errors in these ideas, will write and tell me, so that I can pass on their corrections to fellow readers. I started this magazine for the selfish motive that I wanted to KNOW about the PC 1500: so far my education is proceeding nicely; thank you all very much.

by IAN F. TRAYNOR

Some of the Registers

As well as an 8-bit Accumulator (A) the PC 1500 has 6 general-purpose 8-bit registers for use by the programmer: XH, XL, YH, YL, UH, UL. Apart from being used independently, they can also be paired to form 16-bit data pointers, using the 256-ary Hex coding to form an address. For example, if XH is set to 123(hex 7B) and XL is set to 176 (hex B0) the X-pair can be referred to in instruction codes as the ' X-register' , containing the address 31664 (123*256+176). Other registers,such as the the 16-bit program counter (P),the 5-bit status register (T) and the 16-bit stack pointer (S) may be dealt with in a later article.

Loading the Accumulator (A)

The Accumulator can be loaded with the contents of the 6 general-purose registers, the contents of the address held in X,Y, or U registers, or with the contents of a specified address, through the operation codes: -

Load A with contents of	Hex op-code	Mnemonic	Load A with contents of	Hex op-code	Mnemonic
XL	04	LDA XL	UH	A4	LDA UH
YL	14	LDA YL	address in X	05	LDA (X)
UL	24	LDA UL	address in Y	15	LDA (Y)
XH	84	LDA XH	address in U	25	LDA (U)
YH	94	LDA YH	address a,b	A5,a,b	LDA (ab)

Loading an immediate value (i) into registers

Load i into	Hex op-code	Mnemonic
A	B5,i	LDI A,i
XL	4A,i	LDI XL,i
YL	5A,i	LDI YL,i
UL	6A,i	LDI UL,i
XH	48,i	LDI XH,i
YH	58,i	LDI YH,i
UH	68,i	LDI UH,i

i must be a single byte,
0-255 (hex 00-FF)

Increasing (INC) or decreasing (DEC) registers by 1

Register	Hex op-code INC	Hex op-code DEC
A	DD	DF
XL	40	42
YL	50	52
UL	60	62
XH	FD 40	FD 42
YH	FD 50	FD 52
UH	FD 60	FD 62
X	44	46
Y	54	56
U	64	66

(Note that XH, YH, & UH need 2-byte code)

Combined loading and increasing (SIN) or decreasing (SDE) a register

(single-byte op-codes enable you to transfer the content of the *Accumulator* into an address held in the X, Y, or U register; and then to increase (SIN) or decrease (SDE) that register by 1 .

Address held in register	Hex op-code SIN	SDE
X	41	43
Y	51	53
U	61	63

Branches, Jumps and Returns

One type of conditional branch will be dealt with here: BNZ, i
which has the Hex Op-code 99.
Branch if Not Zero tests whether the immediately previous operation
resulted in zero. If it did NOT, the program branches back the specified
number of bytes i , counting from, *and including* the byte that holds i .

JSR,i,j has a Hex Op-code of BE , and does a subroutine call to an address
(in Hex or 256-ary code) held in the following 2 bytes i and j .

RTS - with the Hex Op-code 9A - *must* be at the end of every Assembly
Language subroutine that is CALLed from BASIC. Otherwise - Crash!!!

DEMONSTRATION PROGRAM

With the op-codes described in this article, you can try this Demonstration
Program, which displays the whole character-set on the screen. Scroll
left and right with the Cursor Keys.

After entering the program, key DEF S to execute lines 10-30. These
POKE the machine-code routine into addresses 30912 to 30930. These addresses
are normally occupied by fixed variable A$ and B$, so do not use these
variables! The advantage of using this area is that you do not use up
extra program memory: nor are you affected by what memory size you have.
After executing DEF S , key DEF SPACE to execute lines 100-130.Then scroll
with left and right Cursor Keys. You will need to BREAK to interupt the
program. This subroutine could be the basis for a wordprocessor program,
where several K of text could be scrolled continuously across the screen.

(In normal use , lines 10-30 could be deleted after execution: but leave
them in for experiments. Try altering the 8th value &1A to &F0 for 16
characters only).

```
10: "S"RESTORE
20: DATA &48, &7B, &
    4A, &B0, &6A, &20
    , &68, &1A, &24, &
    41, &60, &FD, &62
    , &99, &07, &BE, &
    E8, &CA, &9A
30: FOR Z=1TO 19:
    READ A:POKE 30
    911+Z, A:NEXT Z
    :END

100: " "CALL 30912:
     IF INKEY$ =""
     GOTO 100
110: IF ASC INKEY$
     =12IF PEEK 309
     17<229POKE 309
     17, PEEK 30917+
     1:GOTO 100
120: IF ASC INKEY$
     =8IF PEEK 3091
     7>32POKE 30917
     , PEEK 30917-1
130: GOTO 100
```

*This DEMONSTRATION PROGRAM is explained in line-by-line detail next
month; together with some more information about the operation codes.
Latest news is that in a few weeks or months TANDY's"TRS NEWS" will be
imported from USA, and will be sold at TANDY shops. It is said to contain
full details of all the machine code for PC2, serialised over several
months.*

SOFTWARE SURVEY

This represents all the software available for PC 1500 and PC2 at the present date, as far as I know, within UK. While great efforts have been made to make this as accurate and complete as possible, the accuracy and completeness cannot be guaranteed; owing to the constantly changing situation, and the reluctance of certain organisations to supply definite information.

M: mixed subjects   S: screen   P: printer required   **:good value

Programs or details in ITALICS : in preparation or under revision, and readers are seriously advised not to send remittances for these items without previous enquiry.

TITLE	SUBJECT	Memory etc.	AUTHOR	PRICE	FROM
EC15A	Soil drainage, Game, 2way layout etc.	2K M/P/S	SHARP	£14.95	SHARP dealers; ELKAN
BUSINESS FINANCE		2K M/P/S	TANDY	£9.95	TANDY COMPUTER CENTRES
MATHS (1) & (2)		2K P/S	" each	£9.95	
PERSONAL FINANCE		2K M/P/S	"	£9.95	
MATHS PLOTTER for graphs		2K P/S	"	£7.95	
**GAMES		2K M/P/S	"	£9.95	
CHEMISTRY & MATHS		2K M/P/S	"	£9.95	
EXECUTIVE	database	8K P	MICRoL	£18.95	MICROMAIL
PROCOS	numbercruncher	8K P	"	£18.95	
**DBASE	database	8K P	Wadsworth	£24.95(?)	MICROS FOR MANAGERS
3-D VIEW	*perspective*	*8K P*	Calderwood	£24.95	
PERT	*critical path analysis*	*8K P*	-	£24.95	
APD	*pipe flow formulae*	*8K P*	-	£24.95	
DOCTOR,DOCTOR	joke	2K M/S	Atkinson	£5.95	
EASIFILE	database	8K P	Traynor	£19.95	ELKAN ELECTRONICS
TEXTHANDLER (needs CE 153)		8K P	Quadri	£19.95	
**EASICALC	spreadsheet	8K P	Traynor	£19.95	
**EASITREND statistical	forecasting	4K S/P	Traynor	£19.95	
SALES PRESENTATION (customised to requirements)		8K P	Lindley	about £125	

TANDY COMPUTER CENTRE, CENTREPOINT,LONDON W.C.2
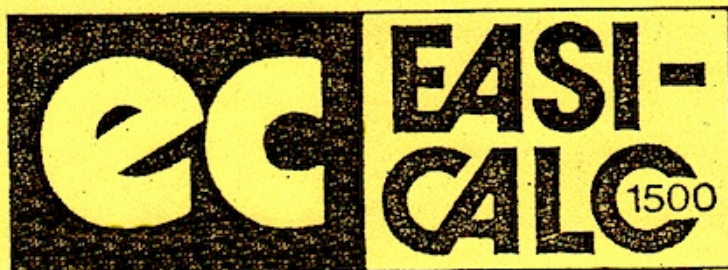*MICROMAIL, 38 BURLEIGH STREET, CAMBRIDGE*
MICROS FOR MANAGERS, 149 GLOUCESTER ROAD, LONDON S.W.7
ELKAN ELECTRONICS, 11 BURY NEW ROAD,PRESTWICH, MANCHESTER