nascom

programs + information

edited by

Merseyside Nascom Users Group

nm

Nascom Microcomputers

Merseyside Nascom User Group.

The Merseyside group was formed in January 1979 and since then the membership has grown from 5 to in excess of 150. In the 10 short months since formation, numerous lectures and demonstrations have taken place and the expertise that now exists in the group is reaching profesional standards.

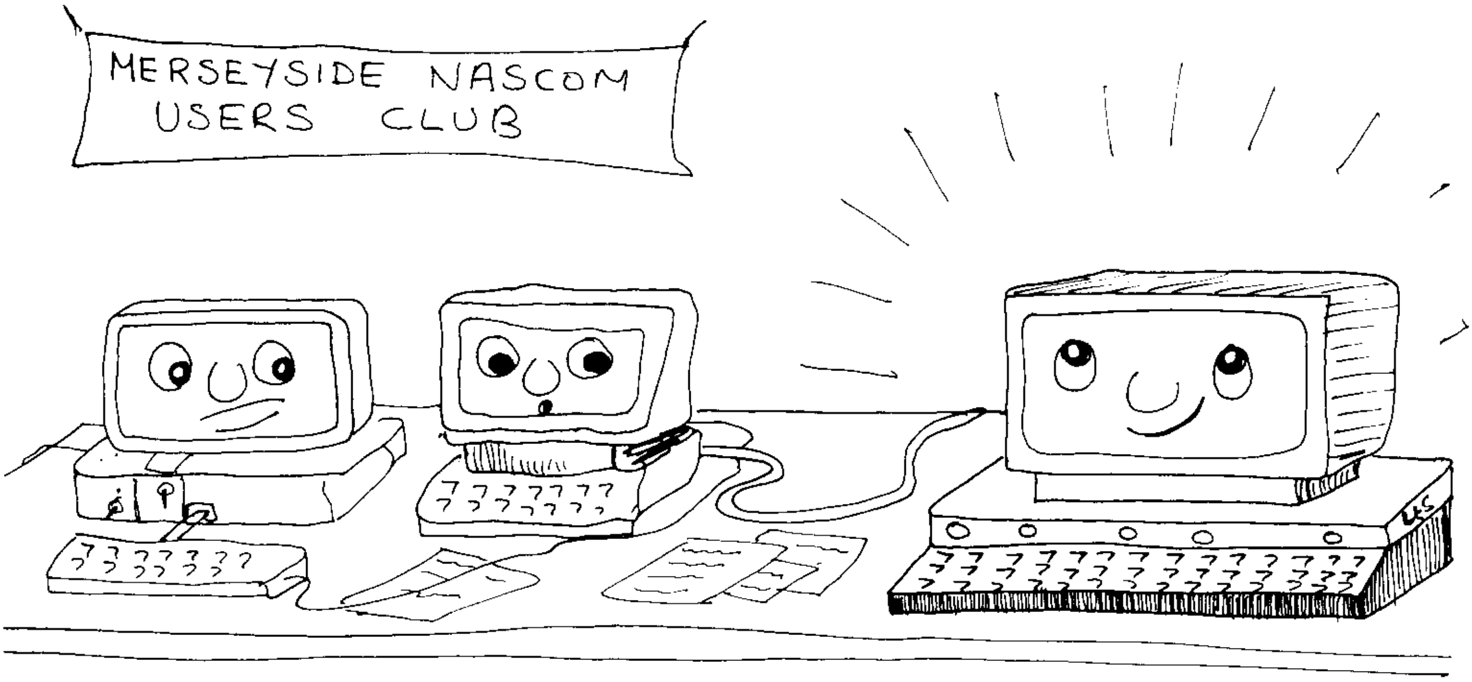
The group meets on the first Wednesday of each month at the Mona Hotel in James Street Liverpool from 7.30 pm onwards. All new members and visitors will be made welcome to the meetings, where you are assured of a very interesting and productive evening.

Anybody who requires more information may write to me at the address below, enclosing a stamped addressed envelope.

A reader enquiry service is being set up to deal with specific problems regarding programmes in the book. Details of the service can be found on page 64, but please remember the stamped addressed envelope. (overseas readers please enclose an International Postal Reply coupon).

Please don't write to Nascom.

Graham Myers
34 Hillcrest Drive,
Greasby,
Merseyside.



I remember when he came to these meetings in nothing more than a cardboard box!!

INMC

International Nascom Microcomputer Club

This book represents a major step forward in personal microcomputing in the U.K. I am sure you will be delighted with the standard of work herein. It represents a considerable amount of work by hobbyists who have bought a minimum kit of parts and built and expanded their system to give intelligent capability.

Over the past two years we have seen the explosion of microcomputing in the U.K. Obviously the immediately visible program libraries were for those boxed complete systems which many people decided to buy. However, it was fairly obvious that the majority of people were going to buy and build computer systems based around the boards that would give them not only a software knowledge but also some idea of the hardware. In most cases it has been these people who have founded the microcomputing clubs in the U.K. and, in fact, in Europe. The Merseyside Nascom Users' Group, which makes up a very large part of the Merseyside Computing Club, is no exception to this. This kind of output from a Computer Club is what I had hoped for during the last two years and that is why I feel this book is the first of a series that will be produced not only by the Merseyside Nascom Users' Group but by other groups throughout the country.

Kerr Borland
International Nascom Microcomputer Club

CONTENTS

Calender

B. McGennit

This program is self explanatory on execution. Remember to give the complete year, ie. 1979.

Executed at 0C50

```
0C50   DD 21 E5 OF EF 1E 2A 44    41 59 2D 46 49 4E 44 45
0C60   52 2A 1F 1F 1F 1F 57 68    69 63 68 20 59 65 61 72
0C70   3F 1F 00 CD FA OE DD 23    CD FA OE DD 23 EF 1F 57
0C80   68 69 63 68 20 4D 6F 6E    74 68 3F 1F 20 28 50 72
0C90   65 66 69 78 20 53 69 6E    67 6C 65 2D 64 69 67 69
OCAO   74 20 4D 6F 6E 74 68 73    20 57 69 74 68 20 27 30
OCBO   27 20 29 2E 1F 00 CD FA    OE B7 CC 23 OF FE 13 D4
OCCO   23 OF EF 1E 1F 00 06 03    DD 7E 00 4F E6 FO FE FO
OCDO   CC 23 OF 79 E6 OF FE OF    CC 23 OF DD 2B 10 E9 DD
OCEO   23 FD 21 EA OF 3E 05 32    ED OF AF 32 E9 OF 21 00
OCFO   00 16 00 1E OA 06 03 DD    7E 00 E6 FO B7 28 05 D6
ODOO   10 19 18 F8 DD 7E 00 E6    OF 85 FD 77 00 2E 00 DD
OD10   23 FD 23 10 E2 3A EA OF    ED 44 21 ED OF 86 77 CD
OD20   4D OF 3A EB OF F5 E6 03    4F 28 07 34 47 34 10 FD
OD30   18 06 3E 01 32 E9 OF AF    F1 B9 28 06 35 35 D6 04
OD40   18 F7 CD 4D OF DD 21 ED    OF FD 21 E9 OF 21 69 OF
OD50   16 00 3A EC OF 5F 19 7E    DD 86 00 DD 77 00 7B FE
OD60   03 38 09 FD 7E 00 DD 86    00 DD 77 00 1E OC 19 3A
OD70   EC OF FE 02 20 05 FD 7E    00 18 01 AF 86 32 E8 OF
OD80   21 ED OF CD 4D OF CD 5C    OF EF 20 20 00 21 82 OF
OD90   3A EC OF 5F 7E FE 00 28    03 23 18 F8 23 7E BB 20
ODAO   F3 23 7E FE 00 28 05 CD    3B 01 18 F5 EF 20 20 00
ODBO   3A E5 OF CD 44 02 3A E6    OF CD 44 02 EF 1F 00 CD
ODCO   5C OF EF 53 55 4E 20 4D    4F 4E 20 54 55 45 20 57
ODDO   45 44 20 54 48 55 20 46    52 49 20 53 41 54 1F 00
ODEO   CD 5C OF 3A ED OF B7 28    OB 3D F5 EF 20 20 20 20
```

```
0DF0   00 F1 18 F3 3A E8 0F 4F   16 01 3A ED 0F 5F 7A CD
0E00   44 02 EF 20 20 00 1C 7A   3C 27 57 7B FE 07 28 02
0E10   18 EC EF 1F 00 CD 5C 0F   1E 00 7A CD 44 02 EF 20
0E20   20 00 1C 7A B9 28 0C 7A   B7 3C 27 57 7B FE 07 28
0E30   E1 18 E7 EF 1F 1F 50 72   65 73 73 20 46 20 66 6F
0E40   72 20 66 75 74 75 72 65   20 4D 6F 6E 74 68 2C 20
0E50   61 6E 64 20 50 20 66 6F   72 20 70 61 73 74 20 4D
0E60   6F 6E 74 68 2E 00 EF 1F   1F 50 72 65 73 73 20 27
0E70   53 50 41 43 45 27 20 74   6F 20 74 72 79 20 61 67
0E80   61 69 6E 3B 20 61 6E 79   20 6F 74 68 65 72 20 74
0E90   6F 20 52 45 53 45 54 2E   1F 00 21 E7 0F CD 69 00
0EA0   30 FB FE 20 20 06 EF 1E   00 C3 50 0C FE 46 28 07
0EB0   FE 50 28 1F C3 00 00 7E   FE 12 20 12 3E 01 77 2B
0EC0   7E FE 99 20 09 AF 77 2B   7E FE 99 CC 23 0F B7 3C
0ED0   27 18 1B 7E FE 01 20 13   3E 12 77 2B 7E FE 00 20
0EE0   0A 3E 99 77 2B 7E FE 00   CC 23 0F B7 3D 27 77 DD
0EF0   21 E5 0F EF 1E 1F 00 C3   E1 0C 0E 00 CD 06 0F 07
0F00   07 07 07 E6 F0 4F CD 69   00 30 FB 57 CD 3B 01 7A
0F10   FE 3A 30 0B FE 30 38 07   E6 0F 81 DD 77 00 C9 F6
0F20   FF 18 F7 F1 EF 1E 1F 54   68 65 72 65 20 69 73 20
0F30   61 6E 20 65 72 72 6F 72   20 69 6E 20 79 6F 75 72
0F40   20 64 61 74 65 2E 2E 2E   1F 00 C3 66 0E 7E FE 80
0F50   30 04 D6 07 30 FC C6 07   30 FC 77 C9 EF 20 20 20
0F60   20 20 20 20 20 20 20 00   C9 00 00 03 03 06 01 04
0F70   06 02 05 00 03 05 31 28   31 30 31 30 31 31 30 31
0F80   30 31 00 01 4A 41 4E 55   41 52 59 00 02 46 45 42
0F90   52 55 41 52 59 00 03 4D   41 52 43 48 00 04 41 50
0FA0   52 49 4C 00 05 4D 41 59   00 06 4A 55 4E 45 00 07
0FB0   4A 55 4C 59 00 08 41 55   47 55 53 54 00 09 53 45
0FC0   50 54 45 4D 42 45 52 00   0A 4F 43 54 4F 42 45 52
```

Calendar......continued

```
0FD0   00 0B 4E 4F 56 45 4D 42   45 52 00 0C 44 45 43 45
0FE0   4D 42 45 52 00 19 79 10   31 00 13 4F 0A 01 06 0C
0FF0   06 0C 4E 00 44 01 7F 00   E7 0F 04 31 31 00 A0 0E.
```

****

Filemaster

Les. Chadwick

Filemaster has been written for a basic Nascom using 714 bytes. of memory from 0C50 to 0FDA. The program is designed to automatically display and modify text in file format. To gain the maximum use i should be used in conjunction with the automatic tape stop start circuit that is reproduced in this book

Acknowledgement is given to N. A. Purver for the use of his editor program which forms part of the Filemaster.

To create the initial file of text execute the program from 0D00.

Clear the screen using shift/backspace. The program is now ready to accept text from the keyboard Move the cursor on two places and print the number of your first file in between two prompt symbols e.g. (1)

Each file or identification number/letter must be printed in this way.

You are now ready to print anything that you wish. The maximum text for one file is a complete screen.

The cursor can be manipulated by various key pressings in addition to other commands.

Filemaster command characters

Shift/A: Moves the cursor to the left.

Shift/B: Moves the cursor to the right.

Shift/T: Moves the cursorup the screen.

Shift/L: Moves the cursor down the screen.

Shift/F: Deletes the character at the cursor position.

Shift/D: Deletes cursor line and scrolls up lines below.

Shift/H: Homes the cursor to the top of the screen.

Shift/G: If HEX code is typed then Character is printed.

Shift/I: Scrolls down all lines from cursor position.

Shift/J: Repeats the character at the cursor position.

Shift/R:	Will read in text from tape.
Shift/W:	Will output text onto tape.
Shift/X:	Will find the start of a requested file and halt ready for dump.
Shift/M:	Will put keyboard in lower case until pressed again.
Shift C:	Returns the cursor to the start of the current line.
Shift S:	Splits the line from the cursor position.
New Line:	Deletes all characters in front of the cursor position.

When files have been completed and dumped to tape. The program may then be executed from 0C50 and the screen will ask for the file number.

Press the required key and the program will further request 'Display or Modify?'. Switch on the tape from the beginning and the program will search for the required file and when found, will display it.

If the auto tape circuit is fitted then all tape functions are automatic.

If modify is requested when the program has displayed the requested file, it then goes into the editing mode ready for file modification.

When the file has been modified,rewind the tape a short distance to clear the start of the file. Press shift/X and switch on tape. The correct file will be found and the tape will halt after the file identification number. Then type shift/W and the modified file will be dumped onto tape.

If the auto tape circuit is not used, it is most important to stop the tape immediatly the load LED is extinguished.

If not, unwanted text may be left on tape prior to dumping.

Slight variations exist between the monitor T4 and B. Bug.

Listed below are the program changes necessary for running on the various monitor.

Adress	T4 monitor.	B. Bug.	T2 monitor
0ED7	5E	5D	5D
0EEF	5E	5D	5D
0EF6	5E	5D	5D
0EFD	5E	5D	5D
0F0A	5E	5D	5D

These are calls made to the output routine in monitor.

005E in T4 and 005D in B. Bug and T2.

Keyboard look up table

	T4/B Bug	T2
0E40	61	41
0E44	62	42
0E48	74	54
0E4C	6C	4C
0E50	68	48
0E54	6D	4D
0E58	64	44
0E5C	73	53
0E60	69	49
0E64	66	46
0E68	63	43
0E6C	67	47
0E70	77	5D
0E74	72	52
0E78	6A	4A
0E7C	78	5E

```
0C50   EF 1E 00 21 98 09 22 18   0C EF 57 68 69 63 68 20
0C60   46 69 6C 65 3F 20 3E 20   00 2A 18 0C 36 20 CD C1
0C70   0F 00 00 00 00 00 00 21   8E 0A 22 18 0C EF 44 69
0C80   73 70 6C 61 79 20 46 69   6C 65 20 28 44 29 20 6F
0C90   72 20 4D 6F 64 69 66 79   20 46 69 6C 65 20 28 4D
0CA0   29 20 00 2A 18 0C 36 20   CD 69 00 30 FB FE 44 28
0CB0   06 FE 4D 28 14 18 F1 CD   A8 0F CD D90C CD 3B 01
0CC0   CD A6 0D CD 0D 0F C3 93   0F CD A8 0F CD D9 0C CD
0CD0   A6 0D CD 0D 0F C3 00 0D   00 CD D0 0F 30 FB FE 3C
0CE0   28 02 18 F5 CD 3E 00 30   FB BA 28 02 18 EB C9 EF
0CF0   1E 00 CD D9 0C CD 3B 01   CD 87 0F 00 00 00 00 00
0D00   C3 CF 0D E5 29 29 7D D6   28 FE C0 30 05 7C D6 20
```

```
OD10   FE OF E1 C9 11 FF FF 18   03 11 CO FF 19 CD 03 OD
OD20   D8 F5 ED 52 F1 C9 11 01   00 18 03 11 40 00 CD 03
OD30   OD DO 19 18 CE CD 03 OD   DO E5 E5 D1 23 CD 03 OD
OD40   30 04 7E 12 18 F4 EB 36   20 E1 C9 E5 46 23 CD 02
OD50   OD 30 F6 7E 70 47 18 F5   E5 CD 03 OD 30 1A CD 14
OD60   OD 38 FB E5 E5 CD 2B OD   D1 30 04 7E 12 18 F5 EB
OD70   36 20 E1 CD 26 OD 38 EB   E1 C9 E5 CD 03 OD 30 17
OD80   CD 14 OD 38 FB E5 46 CD   2B OD 30 05 7E 70 47 18
OD90   F6 E1 CD 26 OD 38 EE E1   C9 F5 CD 4B OD CD 03 OD
ODAO   C1 DO 70 C3 26 OD 21 OA   08 C9 CD 14 OD DO C3 35
ODBO   OD CD A6 OD 06 OF C3 AB   OE 10 FB C9 36 20 CD 26
ODCO   OD 38 F9 CD C9 OD C3 2B   OD CD 14 OD 38 FB C9 CD
ODDO   A6 OD 4E 06 00 CD 69 00   38 1C CD 03 OD 30 F6 10
ODEO   F4 7E FE 5F 28 04 36 5F   18 E9 79 FE 5F 28 03 77
ODFO   18 E1 36 20 18 DD 71 E5   21 D2 OD E3 FE 1D CA AA
OEOO   OD FE 1E CA B1 OD FE 1F   CA BC OD FE 41 DA 00 OD
OE10   47 3A 09 OC CB 67 78 20   OE 3A 85 OF CB 47 78 CA
OE20   99 OD C6 20 C3 99 OD E5   21 40 OE 11 04 00 7E B7
OE30   28 06 B8 28 05 19 18 F6   E1 C9 23 E3 C9 41 00 00
OE40   61 C3 35 OF 62 C3 3B OF   74 C3 69 OF 6C C3 6F OF
OE50   68 C3 A6 OD 6D C3 81 OE   64 C3 88 OE 73 C3 8E OE
OE60   69 C3 A5 OE 66 C3 35 OD   63 C3 C9 OD 67 C3 B3 OE
OE70   77 C3 DO OE 72 C3 OD OF   6A C3 57 OF 78 C3 AF OF
OE80   00 E5 21 85 OF 34 E1 C9   CD C9 OD C3 58 OD E5 CD
OE90   7A OD CD BC OD E1 E5 CD   2B OD 30 07 06 30 CD AA
OEAO   OD 10 FB E1 C9 CD C9 OD   C3 8E OE E5 CD BC OD 10
OEBO   FB E1 C9 01 00 02 CD 69   00 30 FB FE 41 38 02 D6
OECO   07 D6 30 81 F5 87 87 87   87 4F F1 10 E9 C3 99 OD
OEDO   E5 CD A6 OD 3E 3E CD 5E   00 05 OF C5 01 01 30 7E
OEEO   23 FE 20 20 03 OC 18 10   OD 28 09 F5 3E 20 CD 5E
```

```
0EF0   00 F1 18 F4 0C CD 5E 00   10 E5 3E 1F CD 5E 00 11
0F00   10 00 19 C1 10 D5 3E 2A   E1 C3 5E 00 00 E5 CD 75
0F10   0F FE 3E 20 F9 CD 75 0F   FE 1F 28 0D FE 1D 28 0E
0F20   FE 2A 28 7E CD 99 0D 18   EC CD BC 0D 18 E7 CD AA
0F30   0D 18 E2 E1 C9 CD 41 0F   C3 14 0D CD 41 0F C3 26
0F40   0D AF 11 02 0C 06 07 12   13 10 FC 4E 36 5F 06 10
0F50   CD 35 00 10 FB 71 C9 CD   41 0F CD 4B 0D C3 26 0D
0F60   CD 41 0F CD 7A 0D C3 2B   0D CD 41 0F C3 19 0D CD
0F70   41 0F C3 2B 0D CD 69 00   38 08 DB 02 17 30 F6 DB
0F80   01 C9 E1 18 AE 89 00 3E   04 E5 21 00 0C AE D3 00
0F90   77 E1 C9 CD 69 00 30 FB   FE 53 28 03 18 F5 00 C3
0FA0   50 0C CD 51 00 C3 33 0F   EF 1E 00 CD 51 00 C9 CD
0FB0   51 00 CD D9 0C CD 51 00   C3 00 0D CD 51 00 C3 D0
0FC0   0E 11 3D 0E CD 69 00 30   FB CD 3B 01 12 C3 77 0C
0FD0   11 3D 0E 1A 57 CD 3E 00   C9 B5 D5 DD 92 B1 F5 FD
```

****

The Elliot Nascom 1 Clock

David Ell
(age

The program executes from 0c60, leaving space for addressing the alarm subroutine, and counting commen after typing the start time in HRS, MINS, and SECS (2 digits each). Time is displayed in 24 hour mode digital form tigether with simple graphics and by connection of an amplifier to the collector of the "DRIVE" i transistor TR2 a "tick-tock." sound is prcduced every second.

The accuracy of the clock can be adjusted by altering the delay loops at the location given.

Main routines are located as follows;-

0C69 — Display Time on screen

0C88 — Coarse Time adjustment

0CF0 — Starting routine

0DE0 — Clock face

0E24 — "tick"

0E3B — "tock"

0E48 — Pendulum

0E5E — Fine time adjustment

Clock can be regulated to less than 1 second/day error.

Users with T2 monitor insert 00 in place of 1C at the following addresses: 0D20, 0D67, 0DBC, 0DD7, 0E86

Alarm Subroutine if used, Execute from 0C50

Input alarm time in HRS and MINS.

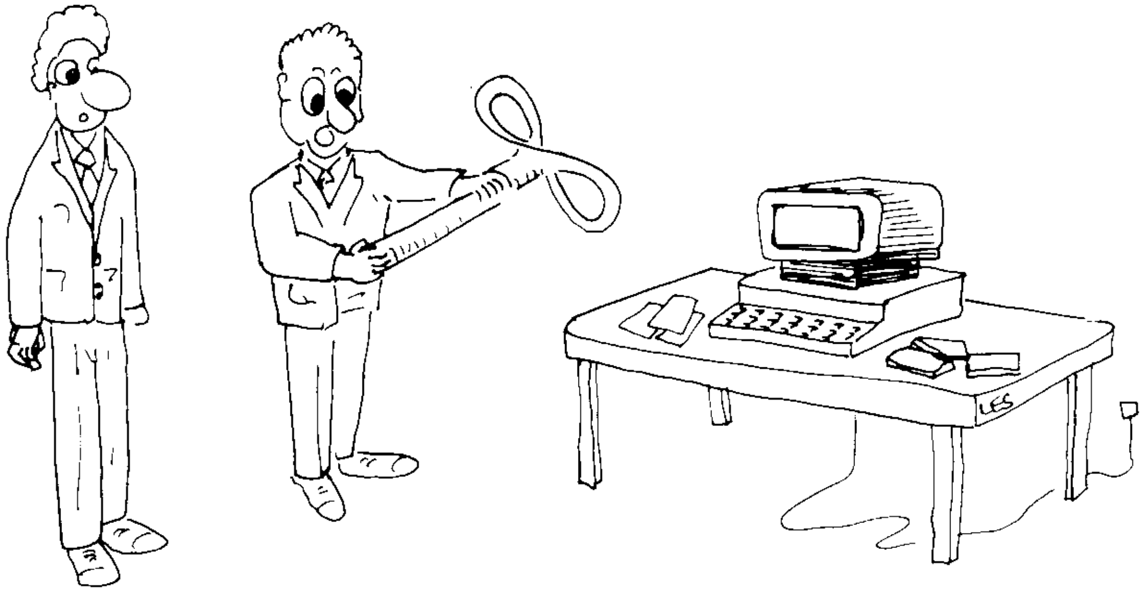
Address of alarm message 0E7E

```
0C60   3E 1E CD 3B 01 CD F5 0C    00 00 78 32 1B 0B 79 32
0C70   1C 0B 7A 32 27 0B 7B 32    28 0B 7C 32 33 0B 7D 32
0C80   34 0B 2C FE 3A CA 98 0C    C5 06 85 CD 35 00 10 FB
0C90   C1 CD E0 0D C3 69 0C 00    24 2E 30 7C FE 36 CA A5
0CA0   0C C3 69 0C 00 1C 7B FE    3A CA B2 0C 26 30 C3 69
0CB0   0C 00 26 30 1E 30 14 7A    FE 36 CA C0 0C C3 69 0C
0CC0   0C 21 30 30 11 30 30 79    FE 3A CA E0 0C 79 FE 34
0CD0   C2 69 0C 78 FE 32 C2 69    0C 01 30 30 C3 69 0C 00
0CE0   0E 30 04 C3 69 0C 00 00    00 00 00 00 00 00 00 00
0CF0   00 00 00 00 00 21 15 08    06 19 36 7E 23 10 FB 00
0D00   21 D5 0B 22 18 0C EF 54    48 45 20 45 4C 4C 49 4F
0D10   54 20 4E 41 53 43 4F 4D    20 31 20 43 4C 4F 43 4B
0D20   1C 00 00 21 CD 0A 06 2A    36 5F 23 10 FB 21 4D 0B
0D30   06 2A 36 7E 23 10 FB 21    0D 0b 22 18 0C EF 54 49
0D40   4D 45 3A 20 20 48 4F 55    52 53 2D 20 3F 20 20 4D
0D50   49 4E 55 54 45 53 2D 20    3F 20 20 53 45 43 4F 4E
0D60   44 53 2D 20 3F 20 20 1C    00 21 1B 0B CD 3E 00 F5
0D70   77 CD 3E 00 F5 23 77 21    27 0B CD 3E 00 F5 77 CD
0D80   3E 00 F5 23 77 21 33 0B    CD 3E 00 F5 77 CD 3E 00
0D90   F5 23 77 F1 6F F1 67 F1    5F F1 57 F1 4F F1 47 E5
0DA0   21 A2 08 36 0C 21 E1 08    36 7F 23 36 7F 23 36 7F
0DB0   21 20 09 22 18 0C EF 7F    7F 7F 7F 7F 1C 00 21 60
0DC0   09 36 7F 23 23 36 12 23    23 36 7F 21 A0 09 22 18
0DD0   0C EF 7F 7F 7F 7F 7F 1C    00 E1 C3 69 0C 00 00 00
0DE0   D9 79 FE 01 CA FB 0D FE    02 CA 05 0E FE 03 CA 0F
```

```
0DF0     0E 21 62 09 36 12 0E 01     C3 19 0E 21 62 09 36 13
0E00     0E 02 C3 30 0E 21 62 09     36 14 0E 03 C3 19 0E 21
0E10     62 09 36 11 0E 00 C3 30     0E 21 E2 09 3E 5C 11 40
0E20     00 F5 C5 06 06 CD 51 00     10 FB C1 F1 C3 48 0E 00
0E30     21 E2 09 3E 2F 11 3E 00     F5 C5 06 20 CD 51 00 10
0E40     FB C1 F1 C3 48 0E 00 00     E5 21 E1 09 06 E0 36 20
0E50     23 10 FB E1 06 02 77 23     19 10 FB 36 0E 06 55 F5
0E60     F1 F5 F1 10 FA D9 C9 00
```

Alarm sub routine.

```
0C50     CD 8A 0E 00 00 00 00 00     00 00 00 00 00 00 00 00
0E60     F1 F5 F1 10 F8 D9 78 FE     00 C0 79 FE 00 C0 7A FE
0E70     00 C0 7B FE 00 C0 7A FE     5F 08 22 18 0C EF 57 41
0E80     4B 45 20 55 50 21 1C 00     E1 C9 3E 1E CD 3B 01 21
0E90     90 09 22 18 0C EF 49 4E     50 55 54 20 41 4C 41 52
0EA0     4D 20 54 49 4D 45 20 20     3F 00 CD 3E 00 F5 F5 3E
0EB0     1D CD 3B 01 F1 CD 3B 01     F1 21 68 0E 77 CD 3E 00
0EC0     F5 CD 3B 01 F1 21 6C 0E     77 3E 3A CD 3B 01 CD 3E
0ED0     00 F5 CD 3B 01 21 70 0E     77 CD 3E 00 F5 CD 3B 01
0EE0     F1 21 74 0E 77 C3 53 0C....
```



'GOD KNOWS WHERE IT FITS IT CAME WITH THAT PROGRAM FOR A REAL TIME CLOCK

Just a taste of the TEXTIE MARK FIVE from S.P.C.
****

```
 XXXXXX    XXXXXX    XXXXXX   XXXXXXXX  XX    XX
XX    XX  XX    XX  XX    XX     XX     XX    XX
XX        XX    XX  XX    XX     XX     XX    XX
 XXXXXX   XX    XX  XX    XX     XX     XXXXXXXX
      XX  XX    XX  XX    XX     XX     XX    XX
XX    XX  XX    XX  XX    XX     XX     XX    XX
 XXXXXX    XXXXXX    XXXXXX      XX     XX    XX

 XXXXXX    XXXXXX   XX    XX  XXXXXXXX  XXXXXXX
XX    XX  XX    XX  XX    XX  XX        XX    XX
XX        XX    XX  XX    XX  XX        XX    XX
 XXXXXX   XXXXXXXX   XXXXXXX  XXXXXX    XXXXXXX
      XX  XX    XX        XX  XX        XXX
XX    XX  XX    XX  XX    XX  XX        XX XX
 XXXXXX   XX    XX   XXXXXX   XXXXXXXX  XX   XXX
```

When it comes to a Word Processor for the NASCOM 1 MARK FIVE will take some beating

For instance by using a combination of the 22 commands and controls, TEXTIE MARK FIVE will latch into upper case, change its line length on both VDU and Printer, display itself on the VDU (the menu and mode situation), Read and Write to Tape very fast indeed, tabulate (there are ten, selectable by the user) and provide very interesting edit facilities. **Without re-typing** it is possible to re-print the same piece of text in many different **guises like this:**

For instance by using a combination of the 22 commands and controls, TEXTIE MARK FIVE will latch into upper case, change its line length on both VDU and Printer, display itself on the VDU (the **menu** and **mode** situation), Read and Write to Tape very fast indeed, tabulate (**there are ten,** selectable by the user) and provide very interesting edit facilities. Without re-typing it is possible to re-print the same piece of text in many different guises like this:

For instance by using a combination of the 22 commands and controls, TEXTIE MARK FIVE will latch into upper case, change its line length on both VDU and Printer, display itself on the VDU (the menu and mode situation), Read and Write to Tape very fast indeed, tabulate (there are ten, selectable by the user) and provide very interesting edit facilities. Without re-typing it is possible to re-print the same piece of text in many different guises like this:

And here are all but one of the pre-set tabs:

```
TAB 0     TAB 1     TAB 2     TAB 3     TAB 4     TAB 5     TAB 6     TAB 7     TAB 8
TAB 0               TAB 2               TAB 4               TAB 6               TAB 8
```

TEXTIE MARK FIVE COSTS THIRTEEN POUNDS FIFTY ON CASSETTE INCLUDING DOCUMENTATION
THE SOFTWARE PUBLISHING COMPANY 8A CHURCH SIDE MANSFIELD NOTTS Tel: (0623) 29237

'Othello'

Mine of Informatic

Our thanks to Mine of Information and Mr. J. Gamson for allowing us to reproduce this game.

The game is played on an 8x8 board and the meaning of the indicators is as follows;

.(full stop) this represents an empty square.

,(comma) this represents a possible move.

To move you specify the row and column number of the square in which you wish to place a piece. The piec do not move once they are placed but they will change colour if they are captured by the opponent.

The winner is the one with the most pieces on the board at the end of the game.

The program executes at 0D00

To restart an interupted game execute at 0D12.

```
0CA0    4D 6F 49 2C 20 31 20 46    72 61 6E 63 69 73 20 41
0CB0    76 65 2C 20 53 74 20 41    6C 62 61 6E 73 07 30 37
0CC0    32 37 2D 35 32 32 30 31    09 09 09 09 4D 6F 76 65
0CD0    20 4E 6F 20 30 30 30 30    20 63 68 6F 69 63 65 73
0CE0    20 00 20 74 6F 20 6D 6F    76 65 75 6E 75 73 65 64
0CF0    01 09 0A 0B F5 F6 F7 FF    00 4A 08 3F 00 00 78 00
0D00    CD C8 0D 21 D4 0C CD 80    0D DA 86 02 21 E1 0C 7E
0D10    2F 77 CD 45 0E 01 0D 0A    11 68 0C CD B0 0D 21 D6
0D20    0C 06 02 CD 94 0D 20 1C    CD DB 01 21 FF 0C 34 7E
0D30    3D CA 03 0D EF 20 47 61    6D 65 20 4F 76 65 72 21
0D40    00 C3 86 02 21 FF 0C AF    77 EF 3F 72 6F 77 20 63
0D50    6F 6C 20 00 CD DB 01 21    54 0B CD 70 0D DA 15 0D
0D60    21 68 0C 06 00 09 CD 00    0E CA 15 0D C3 03 0D 00
0D70    CD A0 0D D8 CD 8C 0D CD    A0 0D D8 81 4F C9 00 00
0D80    3E 39 23 34 BE D0 36 30    2B 34 BE C9 17 4F 17 17
0D90    81 4F C9 00 AF B7 C4 8C    0D 86 23 C6 D0 10 F6 C9
0DA0    7E 23 FE 20 28 FA D6 31    D8 FE 08 3F 3C C9 00 00
0DB0    EF 1E 00 2A F9 0C C5 1A    13 77 23 36 20 23 10 F7
0DC0    0E 2C 09 C1 0D 20 EF C9    21 68 0C 01 2F 64 71 23
0DD0    10 FC CD 89 0E 01 2E 08    3E 08 71 23 3D 20 FB 23
```

```
0DE0     23 10 F5 21 30 30 22 D4    0C 22 D6 0C AF 32 FF 0C
0DF0     67 00 32 E1 0C 2F 6F 22    94 0C 6C 67 22 9E 0C C9
0E00     CD 80 0E C8 11 F0 0C 1A    13 B7 20 05 3A FC 0C B7
0E10     C9 4F 17 9F 47 E5 CD 20    0E CC 30 0E E1 18 E8 00
0E20     09 3A E1 0C 2F BE C0 09    BE 28 FC 2F BE C9 00 00
0E30     22 FB 0C 3A FD 0C B7 C0    3A E1 0C 2F B7 ED 42 BE
0E40     2F 77 28 F7 C9 AF 2F 32    FD 0C 21 30 30 22 D6 0C
0E50     21 73 0C 0E 08 06 08 C5    01 2C 2E 7E B8 28 04 B9
0E60     20 10 70 CD 00 0E 28 0A    36 2C E5 21 D6 0C CD 80
0E70     0D E1 23 C1 10 E1 23 23    0D 20 DA AF 32 FD 0C C9
0E80     AF 32 FC 0C BE C8 2F BE    C9 3E 39 21 70 0C 11 0A
0E90     00 06 08 3D 77 2B 10 FB    06 08 19 77 3C 10 FB 2E
0EA0     73 C9.......
```

Othello. copyright Mine of Information,
1 Francis Avenue,
St. Albans.

****

Right Type

Brian E. Kelly

This short program was one of the first programs that we came across on Merseyside. It provided us with a teaching aid in that it teaches you how to type and also by disecting it can teach you the fundamentals of programing.

It is for these reasons that we have included this program and we thank Brian for allowing us to re-print it.

The program executes at 0C75

```
P.C.     machine code        label     mn

0C50     54 79 70 69                   (data)
         6E 67 20 61
         62 69 6C 69
         74 79 20 69
         6D 70 72 6F
         76 65 72 2E
```

```
P.C.      machine code        label      mn.

          69 6E 63 6F
          72 72 65 63
          74 20 20 30
          30
0C75      21 50 0C            START.     LD HL 0C50
0C78      11 D5 0B                       LD DE 0BD5
0C7B      01 18 00                       LD BC 0018
0C7E      ED B0                          LDIR
0C80      11 E5 08                       LD DE 08E5
0C83      01 0D 00                       LD BC 000D
0C86      ED B0                          LDIR
0C88      21 6A 0C                       LD HL 0C6A
0C8B      11 CE 08                       LD DE 08CE
0C8E      01 0B 00                       LD BC 000B
0C91      ED B0                          LDIR
0C93      21 A0 09            RANDM      LD HL 09A0
0C96      ED 5F                          LD A,R
0C98      47                             LD B,A
0C99      0E 2C               LOOP1      LD C 2C
0C9B      3E 5A                          LD A 5A
0C9D      B9                  LOOP2      CP A,C
0C9E      28 F9                          JR,Z  (LOOP1)
0CA0      0C                             INC C
0CA1      10 FA                          DJNZ  (LOOP2)
0CA3      71                             LD (HL),C
0CA4      CD 3E 00            INKEY      CALL CHIN
0CA7      B9                             CP A,C
0CA8      20 05                          JR NZ  (BOOB)
0CAA      21 D8 08                       LD HL 08D8
```

```
0CAD     18 03                    JR (INC)
0CAF     21 F1 08        BOOB     LD HL 08F1
0CB2     3E 3A           INC      LD A,3A
0CB4     34                       INC (HL)
0CB5     BE                       CP A,(HL)
0CB6     20 08                    JR NZ  (CHECK)
0CB8     36 30                    LD (HL),30
0CBA     2B                       DEC HL
0CBB     34                       INC (HL)
0CBC     BE                       CP A.(HL)
0CBD     CA 86 02                 JP Z (PARSE)
0CC0     3E F0           CHECK    LD A,F0
0CC2     BD                       CP A,L
0CC3     28 DF                    JR,Z (INKEY)
0CC5     3C                       INC A
0CC6     BD                       CP A,L
0CC7     28 DB                    JR Z,(INKEY)
0CC9     18 C8                    JR (RANDM)
```

****

Noughts and Crosses Les Chadwick

This is a simple program designed for children. All the squares must be filled to complete the game.

The program executes at 0C50.

```
0C50    EF 1E 00 21 24 08 CD 6E   0C 21 30 08 CD 6E 0C 21
0C60    DA 08 CD 84 0C 21 5A 0A   CD 84 0C C3 99 0C 0E 10
0C70    22 18 0C EF 2E 00 CD 93   0C 06 3F 23 00 10 FC 0D
0C80    20 EE C9 06 10 22 18 0C   EF 2E 00 CD 93 0C 23 10
0C90    F4 C9 2A 18 0C 36 20 C9   21 5E 08 3E 31 77 21 6A
0CA0    08 3E 32 77 21 76 08 3E   33 77 21 9E 09 3E 34 77
0CB0    21 AA 09 3E 35 77 21 B6   09 3E 36 77 21 1E 0B 3E
```

```
0CC0        37 77 21 2A 0B 3E 38 77   21 36 0B 3E 39 77 21 CA
0CD0        08 22 18 0C EF 50 72 65   73 73 20 4B 65 79 00 CD
0CE0        93 0C 21 0A 09 22 18 0C   EF 4E 6F 2E 20 31 2E 00
0CF0        CD 93 0C 21 4A 09 22 18   0C EF 54 6F 20 22 54 41
0D00        4B 45 22 00 CD 93 0C 21   8A 09 22 18 0C EF 53 71
0D10        2E 20 4E 6F 2E 31 2E 00   CD 93 0C 21 CA 09 22 18
0D20        0C EF 45 74 63 2E 20 45   74 63 2E 00 CD 93 0C 21
0D30        8A 0A 22 18 0C EF 50 72   65 73 73 20 22 5A 22 2E
0D40        00 CD 93 0C 21 CA 0A 22   18 0C EF 74 6F 20 70 6C
0D50        61 79 20 61 67 61 69 6E   00 CD 93 0C CD 69 00 30
0D60        FB FE 31 28 24 FE 32 28   27 FE 33 28 2C FE 34 28
0D70        31 FE 35 28 36 FE 36 28   3B FE 37 28 40 FE 38 28
0D80        45 FE 39 28 4A C3 5D 0D   21 5E 08 3E 4F 77 C3 DA
0D90        0D 21 6A 08 3E 4F 77 C3   DA 0D 21 76 08 3E 4F 77
0DA0        C3 DA 0D 21 9E 09 3E 4F   77 C3 DA 0D 21 AA 09 3E
0DB0        4F 77 C3 DA 0D 21 B6 09   3E 4F 77 C3 DA 0D 21 1E
0DC0        0B 3E 4F 77 C3 DA 0D 21   2A 0B 3E 4F 77 C3 DA 0D
0DD0        21 36 0B 3E 4F 77 C3 DA   0D 00 00 00 00 00 00 00
0DE0        00 00 00 00 00 3E 4F 21   D5 0D BE 28 15 06 09 0E
0DF0        00 71 16 09 23 15 20 FC   10 F5 C3 5D 0D 00 00 00
0E00        06 09 21 8D 0D 0E 7F 71   16 09 23 15 20 FC 10 F5
0E10        C3 5D 0D  ........
```

****

***STOP PRESS *** STOP PRESS *** STOP PRESS *** STOP PRESS ***

Anybody having experience interfacing a Nascom to a printer is requested to write to Mr. Frank Butler of Mansfield giving details and documentation where possible. He is compiling a fact sheet on the subject for any one who is interested.

The address to write is ; Mr. F. Butler.
8A Church Side, Mansfield,
NOTTS. Tel. 0623 29237

Reverser - Resrever

(For B-Bug only)

A number of variations of this game are in use but this one seems to be the best. Our thanks to Frank Butler of Mansfield for this contribution.

The object of the game is to place 9 digits in numerical order from left to right.

You do this by reversing the order of the digits on each move.

For example;

If you have on the screen,

............................147653289

by reversing 3 digits you will get

............................741653289

if you now reverse 7 you will get

............................235614789

and so on until you get

............................123456789

All this is against the clock and the winner is the one who does it in the shortest time.

The program executes at 0E50

```
0C50    54 48 45 20 4E 55 4D 42   45 52 53 20 47 41 4D 45
0C60    20 4F 42 4A 45 43 54 3A   20 54 6F 20 70 6C 61 63
0C70    65 20 74 68 65 20 6E 75   6D 62 65 72 73 20 69 6E
0C80    20 6F 72 64 65 72 20 4D   45 54 48 4F 44 3A 20 45
0C90    6E 74 65 72 20 31 20 74   6F 20 39 20 2D 20 20 20
0CA0    20 20 20 20 20 20 20 20   20 20 20 20 4C 69 6E 65
0CB0    20 77 69 6C 6C 20 62 65   20 72 65 76 65 72 73 65
0CC0    64 20 66 72 6F 6D 20 74   68 65 20 6C 65 66 74 20
0CD0    66 6F 72 20 74 68 65 20   6E 75 6D 62 65 72 20 6F
0CE0    66 20 64 69 67 69 74 73   20 73 65 6C 65 63 74 65
0CF0    64 2E 54 68 65 20 6C 6F   77 65 73 74 20 73 63 6F
0D00    72 65 20 77 69 6E 73 20   31 20 32 20 33 20 34 20
0D10    35 20 36 20 37 20 38 20   39 20 53 43 4F 52 45 3A
```

```
0D20        47 41 4D 45 20 4F 56 45    52 50 6C 65 61 73 65 20
0D30        72 65 70 65 61 74 20 77    69 74 68 20 61 20 6E 75
0D40        6D 62 65 72 20 62 65 74    77 65 65 6E 20 31 20 61
0D50        6E 64 20 39 50 52 45 53    53 20 22 52 22 20 54 4F
0D60        20 52 45 2D 53 54 41 52    54 54 20 20 20 20 20 20
0D70        01 09 00 21 A0 0F 11 D0    0A ED A0 13 EA 79 0D C9
0D80        00 00 00 20 20 83 20 21    A0 0F 06 09 3E 20 77 23
0D90        10 FC 3E 00 77 21 A0 0F    E5 21 85 0D 3E 09 CD 7A
0DA0        04 01 09 00 21 A0 0F ED    B1 20 02 18 EC E1 77 23
0DB0        3E 00 BE 20 E3 00 00 00    00 00 00 06 09 21 A0 0F
0DC0        3E 30 ED 67 77 23 10 F8    C9 00 00 00 00 00 00 00
0DD0        06 0A 00 21 A0 0F 11 B8    0F ED A0 1B 1B 10 FA C9
0DE0        20 20 20 20 20 20 20 20    20 20 03 CD 69 00 FE 52
0DF0        CA 50 0E 00 00 00 00 00    00 00 FE 31 FA 64 0E FE
0E00        3A F2 64 0E 32 EA 0D 00    00 00 00 00 00 00 3A EA
0E10        0D 21 EA 0D 1E 00 73 ED    6F 00 00 00 00 00 00 00
0E20        00 00 00 00 00 01 00 00    11 00 00 46 5E 00 00 00
0E30        21 BA 0F ED 52 11 A0 0F    7E 12 13 23 00 10 F9 CD
0E40        70 0D CD D0 0D C3 91 0E    00 00 00 00 00 00 00 00
0E50        CD FB 0E CD 87 0D CD 70    0D CD D0 0D D9 1E 30 16
0E60        30 0E 30 D9 D9 ED 53 F4    09 79 32 F6 09 06 FF CD
0E70        69 00 38 14 10 F9 0C 3E    3A B9 20 E9 0E 30 14 BA
0E80        20 E3 16 30 1C BB 20 DD    D9 C3 EE 0D 00 00 00 00
0E90        00 21 90 0A 11 D0 0A 01    11 00 1A 13 ED A1 EA A4
0EA0        0E C3 A9 0E 28 F4 C3 64    0E CD 59 0F CD 82 0F CD
0EB0        69 00 FE 52 CA 50 0E 18    F6 CD 87 0D CD 70 0D CD
0EC0        D0 0D D9 1E 30 16 30 0E    30 ED 53 F4 09 00 79 32
0ED0        F6 09 D9 C9 00 00 00 20    20 20 20 20 20 20 20 20
0EE0        CD FB 0E CD 87 0D CD 70    0D CD D0 0D C3 EB 0D 20
0EF0        20 20 20 20 20 20 20 20    20 20 20 3E 1E CD 3B 01
```

```
0F00        01 10 00 21 50 0C 11 CB    0B ED B0 01 25 00 21 61
0F10        0C 11 4B 08 ED B0 01 25    00 21 87 0C 11 8B 08 ED
0F20        B0 01 23 00 21 AC 0C 11    0B 09 ED B0 01 22 00 21
0F30        D0 0C 11 4B 09 ED B0 01    15 00 21 F2 0C 11 CB 09
0F40        ED B0 01 11 00 21 08 0D    11 90 0A ED B0 01 06 00
0F50        21 1A 0D 11 EC 09 ED B0    C9 00 00 01 04 00 21 20
0F60        0D 11 AC 0A ED B0 01 04    00 21 25 0D 11 EC 0A ED
0F70        B0 C9 00 00 01 2B 00 21    29 0D 11 8C 0B ED B0 C9
0F80        00 00 01 15 00 21 54 0D    11 8C 0B ED B0 C9 00 00
0F90        00 00 00 00 4E 18 13 21    01 0C CB 66 28 12 FE 5A
```

****

????Hangman????

Graham Houghton.

This is the Nascom version of Hangman which has been adapted by Graham Houghton. You enter your list from 0EB0. No word should be longer than 9 letters and each word is spaced with 00. Enter the number of words plus 1 (hex) into 0D91

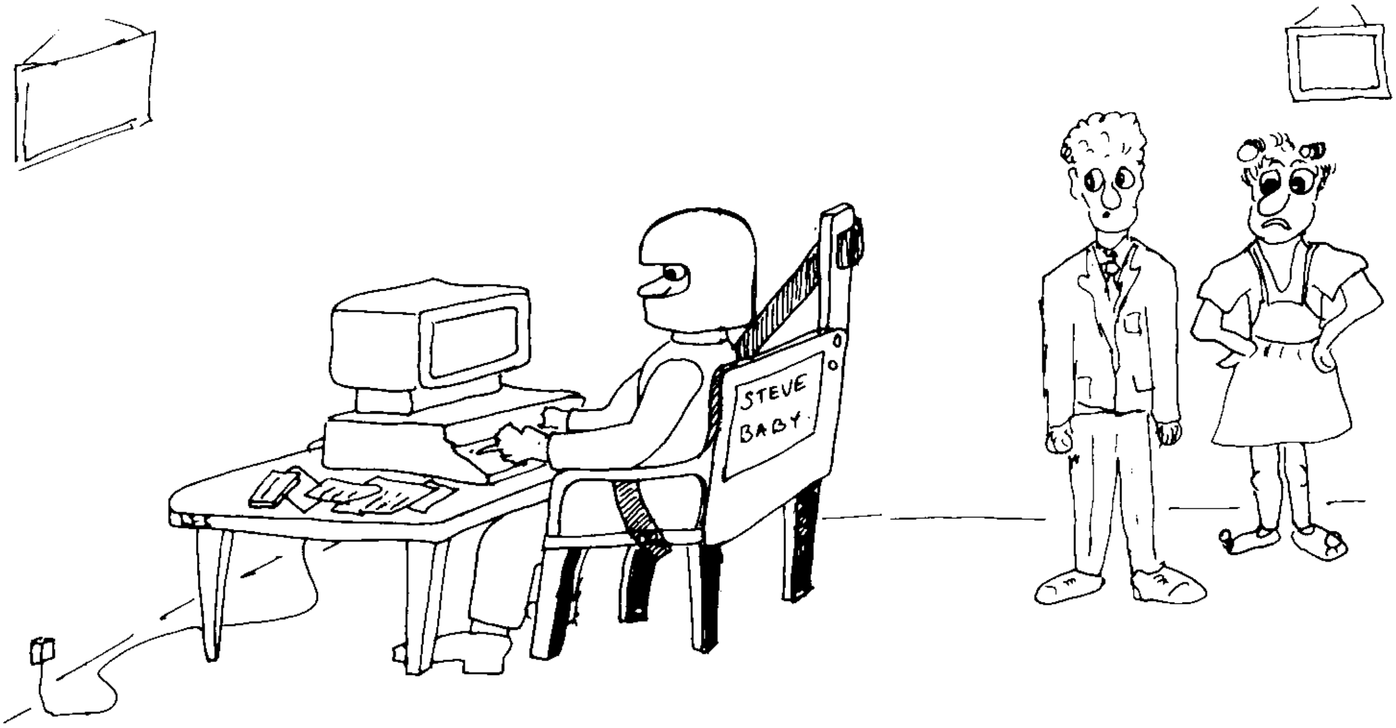
The program executes at 0CD0.

```
0CD0        EF 1E 00 21 D9 0B 22 18    0C EF 2A 2A 2A 20 48 41
0CE0        4E 47 4D 41 4E 20 2A 2A    2A 00 2A 18 0C 36 20 21
0CF0        8A 0B 22 18 0C C3 1E 0D    7E 23 CD 3B 01 10 F9 C9
0D00        CD 3B 01 C9 CD 69 00 30    FB C9 CD 40 02 C3 86 02
0D10        00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 21
0D20        A0 0E 11 B0 0E 06 10 2B    7E 1B 12 05 C2 27 0D 06
0D30        22 21 2C 0E CD F8 0C CD    A0 0D 06 0A CD F8 0C 06
0D40        0D 21 58 0E CD F8 0C CD    C3 0D CD 04 0D E6 7F 4F
0D50        CD 00 0D 57 CD C3 0D 7A    CD E6 0D 50 A1 C2 64 0D
0D60        2E A3 34 56 43 2E A7 CD    F8 0C 42 04 21 88 0E CD
0D70        F8 0C 7A FE 07 CA 08 0E    7B 2E A0 BE C2 3F 0D 06
0D80        14 21 65 0E CD F8 0C 3E    20 CD 00 0D CD 04 0D E6
0D90        7F CD 00 0D FE 4E CA 0A    0D C3 D0 0C 00 00 00 00
0DA0        E5 21 A1 0E CD D0 0D 96    D2 A4 0D 2E AF 97 5F 23
```

```
ODBO        22 A5 OE 23 1C BE C2 B3   OD 05 C2 AD OD 7B C6 30
ODCO        E1 77 C9 06 06 3E 20 CD   00 OD 05 C2 C7 OD C9 00
ODDO        3A 2B OE B7 C2 D8 OD 3C   47 E6 8E 78 EA EO OD 37
ODEO        17 32 2B OE 47 C9 48 43   D5 21 AO OE E5 2A A5 OE
ODFO        11 A7 OE BE C2 FC OD 12   4F E3 34 E3 05 CA 05 OE
OE00        1C 2C C3 F3 OD E1 D1 C9   06 OF 21 79 OE CD F8 OC
OE10        43 2A A5 OE CD F8 OC 06   08 21 71 OE CD F8 OC C3
OE20        87 OD 00 00 00 00 00 00   00 00 00 04 49 20 41 4D
OE30        20 54 48 49 4E 4B 49 4E   47 20 4F 46 20 41 20 57
OE40        4F 52 44 20 54 48 41 54   1F 20 48 41 53 20 37 20
OE50        4C 45 54 54 45 52 53 2E   1F 20 59 4F 55 52 20 47
OE60        55 45 53 53 3F 1F 20 57   45 4C 4C 20 44 4F 4E 45
OE70        21 1F 20 41 47 41 49 4E   3F 1F 20 54 48 45 20 57
OE80        4F 52 44 20 57 41 53 20   20 48 41 4E 47 4D 41 4E
OE90        00 2A 00 00 00 00 00 2D   2D 2D 2D 2D 2D 2D 2D 2D
OEAO        00 2A 00 00 00 C5 OE 2D   2D 2D 2D 2D 2D 2D 2D 2D.

OEBO.......  Start of word list.
```

****



"THE PROGRAM HE'S WRITING KEEPS CRASHING!"

Printer output routine.

Frank Butler.

This routine is based on the Printer output program presented in the Nascom newsletter for the Centronics Printer. That program used a bit in the PIO Port B for 'handshake' thus effectively using the whole port for the sake of recognising the state of one input line. The program below uses one of the spare user bits in the keyboard port either of the two can be used but only one or two 'Rotate' instructions will be required depending upon the user's choice) and this leaves the port B totally free for other uses.

The program was written to work in conjunction with a text processing system and a routine which enables Nascom 1 memory contents to be printed in HEX.

```
0C50    E5          PCHAR   PUSH HL
0C51    D5                  PUSH DE
0C52    C5                  PUSH BC
0C53    F5                  PUSH AF
0C54    3E 0F               LD A, 0F     INITIATE PORT
0C56    3E 0F               OUT (ACTRL), A
0C58    DB 00       PRDY    IN A, (BDATA)
0C5A    17                  RLA
0C5B    17                  RLA
0C5C    38 FA               JR C, PRDY
0C5E    F1                  POP AF
0C5F    CB FF               SET 7, A
0C61    D3 04               OUT  (ADATA), A
0C63    CB BF               RES 7, A
0C65    D3 04               OUT (ADATA), A
0C67    CB FF               SET 7, A
0C69    D3 04               OUT  (ADATA), A
0C6B    CD 35 00            CALL 0035   7.5 m.sec delay.
0C6E    C1                  POP BC
0C6F    D1                  POP DE
0C70    E1                  POP HL
0C71    C9                  RET
```

The following routine enables machine code to be output to a line printer.

It has been designed to run in conjunction with the 'print' routine above at 0C50 and is designed to work with an INTEGRAL DATA SYSTEMS IP 125 line printer.

The program outputs the Hex address followed by the contents of the next 16 locations. Then follows a line feed and the process repeats. The start and finish addresses (add 1 to the required finish address) are entered using the 'M' command at 0C80 and 0C82 entering the high order byte first. Execute the program at 0C88. The Hex conversion routine in this program is based on the Nasbug monitor Tabulate command.

```
0C88    2A 80 0C                LD HL, (ARGA): POINT HL START ADDRESS
0C8B    ED 5B 82 0C             LD DE, (ARGB): POINT DE END ADDRESS
0C8F    3E 0A        LINEST     LD A, 10d.
0C91    CD 50 0C                CALL 0C50
0C94    7C                      LD A, H
0C95    CD B8 0C                CALL DUMPY
0C98    7D                      LD A, L
0C99    CD B8 0C                CALL DUMPY
0C9C    3E 20                   LD A, 20
0C9E    CD 50 0C                CALL 0C50
0CA1    06 10                   LD B, 16d    ;LENGTH IN DECIMAL
0CA3    7E           CODEOT     LD A, (HL)
0CA4    23                      INC HL
0CA5    CD B8 0C                CALL DUMPY
0CA8    E5                      PUSH HL      :SAVE WHILST CHECK END
0CA9    ED 52                   SBC HL, DE
0CAB    E1                      POP HL
0CAC    CA D0 0C                JP Z, ENDIT :IF END JUMP OUT
0CAF    3E 20                   LD A, 20     ;PRINT A SPACE
0CB1    CD 50 0C                CALL 0C50
0CB4    10 ED                   DJNZ CODEOT
0CB6    18 D7                   JR  LINST
0CB8    F5           DUMPY      PUSH AF
0CB9    1F                      RRA
```

```
OCBA    1F                  RRA
OCBB    1F                  RRA
OCBC    1F                  RRA
OCBD    CD C1 OC            CALL  DUMPA
OCCO    F1                  POP  AF
OCC1    E6 OF     DUMPA     AND  OF
OCC3    C6 30               ADD  A, 30
OCC5    FE 3A               CP 3A
OCC7    DA CC OC            JP C, PRINTY
OCCA    C6 07               ADD A, 07
OCCC    CD 50 OC  PRINTY    CALL OC50
OCCF    C9                  RET
OCDO    3E OA     ENDIT     LD A, 10     ;LINE FEED
OCD2    CD 50 OC            CALL OC50
OCD5    C3 00 00            JP  0000     ;TO MONITOR.
```

****

Exam Mark Handler

This was the first program that I wrote for the Nascom, so no claim is made that it is in anyway a sophisticated program. It is a utilitarian program in every sense but apart from being very useful to teachers I hope others will find that they can adapt some of the routines used.

The program is executed from 0C50. The command ENTER DATA is printed and marks, which can range from 0 to 99 inclusive, are now entered. the end of each form is signified by a letter E and at the end of the entire list a full stop followed by newline is entered. the T2 MODIFY routine is used to enter the marks so the same rules apply to entering them as to modifying memory. ie. Backspace can be used to correct the present line, you must not type off the right of the screen (use new line before you reach it), leading zeros need not be entered, a space must be left between each entry etc.

The total number of marks and "E"s must not exceed 224 and there can be up to 17 form (sets). This is ample for a whole year group.

As soon as "newline" is keyed after entry of the data the marks are printed back 14 (a page) at a time in the following format.

PUPIL NUMBER (ie. First pupil entered in the year group is 1, next is 2 etc.)

MARK (Original raw score)

GRADE (This is the mark adjusted as explained below OR a letter grade)

POSITION IN YEAR

POSITION IN FORM

The average raw score for the year is also shown. Equal positions are indicated by an = sign and other positions adjusted accordingly. During entry of marks an absence can be indicated by AB. This allows all pupils to be given a number in advance and makes the task of entering results on form lists less prone to error. The absent pupil's number is printed out, but of course no results.

Leading zeros are not printed. In all calculations half marks are rounded up, below half rounded down The GRADE or adjusted mark is calculated from the average. Examination Boards use the average mark to determine the pass grades because this does not show fluctuation from year to year if the exams are of the same standard; and if the exams are not of the same standard grades based on the average are still valid. We have found the following adjustment to work well over the years and give a reliable indication of achievement

Marks above average are adjusted so that the new mark is = $\frac{50}{100}A$ old marks + 2 (50-A)

Marks below average and equal to average are adjusted so that the new mark = $\frac{50}{A}$ x old mark

This ensures that the average mark becomes 50 and no child recieves more than 99. Adjustments are on a sliding scale and the rank order is not affected. The positions calculated by the program are based on the original raw score so if with rounding two pupils achieve the same "grade" their positions depend on the marks on their scripts. If you prefer this section of the program can be replaced by a routine that gives a letter grade instead based on the raw score. (The second DUMP gives letter grades as follows: 65 and above = A, 55-64 = B,45-54 = C,35-44 = D,below35 = E).

After each page the display waits and prints "contd." if there are more pupils in that form to follow, or "N fo next form" if that form is complete. The space bar gets the next page of a form and N gets the next form At the end of the last form in the year group "N for next form" is not printed. At the end of any form, keying "A' causes a return to the start of the display of the first form again. Keying ! causes a return to the start of the program for new data to be entered. The existing data is lost.

After processing in this way marks from different subjects can be compared with each other as a 50 in one subject is now equivalent to a 50 in another.

This program can save you hours of repetitive work. It will produce for you a position in form and position in year group. The position in year group makes setting very simple and it is hoped to produce another program to provide a listing of pupils in rank order.

As well as this the year average is calculated so that you can compare the standard with prevìous exams. An adjusted mark is also produced for each pupil that is based on lthe average mark. This adjusted mark gives the average child 50. This means that the child can compare his results in different subjects provided each subject's marks are adjusted in this way. This is because a particular grade in one subject is now equivallent to the same grade in another regardless of how easy or difficult the examinations were.

The program does assuem that a reasonable spread of abilities are present in the year group. If they are not in your subject then you may prefer letter grades based on the original raw scores and these are easily obtained by replacing the relevant part of the program with a routine to do this.

```
0C50   31 33 0C EF 1E 00 3E 00   21 02 0F 06 E0 77 23 10
0C60   FC 01 0C 00 21 C8 0E 11   0A 0A ED B0 21 02 0F CD
0C70   B0 01 2B 22 E5 0F 22 ED   0F 3E 0A 32 01 0F 5D DD
0C80   21 DB 0F FD 21 E5 0F 4E   7E FE AB 28 4A 3E F0 A1
0C90   CB 0F 47 CB 0F CB 0F 80   47 3E 0F A1 FE 0A 38 35
0CA0   DD 22 E3 0F D5 E5 EB 2A   ED 0F AF ED 52 7D FE 00
0CB0   20 07 3A E5 0F BB 28 12   7D DD 77 14 DD 23 2A E5
0CC0   0F ED 52 44 4D 62 6B 23   ED B0 FD 35 00 E1 D1 2B
0CD0   22 ED 0F 18 03 80 77 2B   1D 20 AC 2A E5 0F 5D 55
0CE0   06 00 DD 21 00 00 7E FE   AB 20 05 15 36 FF 18 03
0CF0   4E DD 09 2B 1D 20 EF DD   E5 E1 3E 00 BA CA 50 0C
0D00   7A CD 9F 0E 7D 32 00 0F   CD 19 0E EF 1E 00 01 02
0D10   00 21 0E 0E 11 F2 0B ED   B0 01 21 00 21 DB 0E 11
0D20   D1 0B ED B0 21 01 0F 22   E7 0F 2A E7 0F CD 43 0D
0D30   EF 20 63 6F 6E 74 64 2E   1F 00 CD 3E 00 FE 20 20
0D40   F9 18 E7 06 0E C5 E5 CD   40 02 26 00 CD 08 0E E1
0D50   E5 6E 26 00 CD 08 0E E1   E5 66 3A 00 0F BC F5 38
0D60   06 CD B9 0E F1 18 13 6C   D6 32 ED 44 87 5F 16 00
0D70   19 65 CD B9 0E F1 D6 64   ED 44 CD 9F 0E CD 08 0E
0D80   E1 E5 ED 4B E5 0F DD 2A   E5 0F CD C7 0D CD 08 0E
0D90   E1 E5 CD 63 0E CD 08 0E   E1 C1 3A E5 0F BD CA EE
```

```
ODAO   OD 3A FD OE BD 20 19 23   22 E7 OF EF 20 4E 20 66
ODBO   6F 72 20 6E 65 78 74 20   66 6F 72 6D 00 C3 EE OD
ODCO   23 10 82 22 E7 OF C9 11   00 00 7E DD BE 00 FA D6
ODDO   OD 18 04 14 18 03 13 28   FA DD 2B OD 20 ED 15 1C
ODEO   3E 00 BA 28 05 3E 3D 32   10 OE EB 26 00 C9 CD 40
ODFO   02 10 FB E1 CD 3E 00 FE   4E CA 2A OD FE 41 CA 24
OE00   OD FE 21 CA 50 OC 20 EC   CD 19 OE EF 20 00 00 00
OE10   20 7C 00 3E 20 32 10 OE   C9 7D FE FF 20 OF F1 F1
OE20   E1 E5 CD 63 OE 3E 20 32   10 OE C3 98 OD 01 OD OE
OE30   FD 21 D4 OE AF FD 56 00   FD 5E 01 B7 ED 52 38 03
OE40   3C 18 F8 19 C6 30 02 03   FD 23 FD 23 7B FE 01 20
OE50   E3 1E 02 21 OD OE 7E FE   30 20 07 3E 20 77 23 1D
OE60   20 F4 C9 7D FE 01 20 1A   AF FD 2A E3 OF FD 86 14
OE70   32 FD OE FD 4E 14 ED 43   E9 OF FD 22 EB OF 3C 32
OE80   FF OE 3A FF OE BD 28 OC   DD 2A FD OE 3A E9 OF 4F
OE90   CD C7 OD C9 FD 2A EB OF   FD 2B 3A FD OE 18 CE ED
OEAO   44 4F 06 FF 1E 00 09 30   03 1C 18 FA ED 42 ED 44
OEBO   4F 7D 87 B9 38 01 1C 6B   C9 2E 00 11 32 00 3E 08
OECO   29 30 01 19 3D 20 F9 C9   45 4E 54 45 52 20 44 41
OEDO   54 41 20 OB 00 64 00 OA   00 01 00 4D 41 52 4B 20
OEEO   47 52 41 44 45 20 20 59   45 41 52 20 20 46 4F 52
OEFO   4D 20 20 20 59 72 2E 41   76 2E 3D 20 20 00 OF....
```

****

Graham Houghton. **"SCREENWRITER"**

This program is used to 'pre-load' your cassettes with an explanation of the program that follows. For example you may wish to display the name of the program and the instructions on how to execute the program.

To use the program you execute at 0C50. then type in a string of text such as 'Screenwriter'. This string is stored in RAM. and the only limit to the length of the string is the amount of user ram available. Always remember after typing 'new line' to then type a space otherwise on input the monitor will execute any command characters that appear at the beginning of the line.

This facility may be put to good use by finishing the string with 'new line', L, 'new line'. This will then put the monitor into the load function.

An indication of the amount of ram available is given by a display at the top centre of the screen.

Terminate the text string with shift 4 (dollar sign) which is used by the program as an end of text symbol.

Put your tape recorder onto record and type shift, comma. The text string is then loaded onto tape. You may then record the program to which the text string applies.

To write into RAM.   EOC50

To transfer text to tapeE0C66

To modify the speed of the output, change the contents of location 0C74

The maximum speed appears to be 03H.

```
0C50        11 B8 0C CD 69 00 30 FB
0C58        FE 3C 28 0A 12 13 CD 3B
0C60        01 CD A1 0C 18 ED EF 1E
0C68        1F 00 11 B8 0C CD 51 00
0C70        CD 99 0C 0E 03 CD 35 00
0C78        0D 20 FA 1A 06 24 B8 28
0C80        0F CD 3B 01 CD 5D 00 13
0C88        CD A1 0C 18 E6 CD 99 0C
0C90        CD 51 00 EF 1E 00 C3 86
0C98        02 06 00 CD 35 00 10 FB
0CA0        C9 ED 4B 18 0C 21 DE 08
0CA8        22 18 0C 7A CD 44 02 7B
 0CB0       CD 44 02 ED 43 18 0C C9
```

****

Crash Landing

Graham Houghton

Versions of Crash have been written for every type of processer based machine from the small pocket calculator to the main frame. Many versions for the Nascom are 'floating' around and this is just one of them.

The program executes at 0C80.

You enter the amount of fuel you wish to burn per second followed by new line. The maximum rate of burn is 399.

```
0C50    00 65 01 00 32 00 00 00   04 00 00 00 28 51 00 00
0C60    80 96 47 00 00 00 00 00   01 00 00 00 1A 0A 28 51
0C70    00 00 00 00 00 20 3E 20   00 2A 18 0C 36 20 CD 69
0C80    3E 1E CD 3B 01 CD B1 0F   21 D7 0B 22 18 0C EF 07
0C90    20 07 20 43 52 41 53 48   20 4C 41 4E 44 49 4E 47
0CA0    20 07 20 07 00 2A 18 0C   36 20 21 CB 08 22 18 0C
0CB0    EF 54 69 6D 65 20 28 73   29 20 2A 20 48 65 69 67
0CC0    68 74 20 28 66 29 20 2A   20 53 70 65 65 64 20 28
0CD0    66 2F 73 29 20 2A 20 46   75 65 6C 20 28 6C 62 29
0CE0    00 21 50 0C 06 25 36 00   23 10 FB 21 51 0C 36 65
0CF0    23 36 01 21 54 0C 36 32   21 5D 0C 36 50 21 62 0C
0D00    36 50 21 68 0C 36 01 21   0A 09 36 20 21 6F 0C 36
0D10    50 0E 00 CD 18 0F 21 19   0A 36 3E 23 36 5F 22 18
0D20    0C 21 8A 0B 36 20 CD 3E   00 FE 1F 28 05 CD 3B 01
0D30    18 F4 2A 18 0C 22 6C 0C   2A 6C 0C 36 20 11 1A 0A
0D40    CD 5A 02 3A 12 0C 2A 13   0C FE 04 30 23 7C FE 04
0D50    30 1E 22 64 0C C5 01 10   00 21 1A 0A 11 DA 09 ED
0D60    B0 06 10 36 20 2B 10 FB   36 5F 22 6C 0C C1 18 05
0D70    2A 6C 0C 36 3F 11 50 0C   21 64 0C CB E9 CD FE 0E
0D80    CB 71 28 17 CD 89 0F CB   E1 21 64 0C 36 00 23 36
0D90    00 21 50 0C 06 04 36 00   23 10 FB 11 5C 0C 21 54
0DA0    0C CB A9 CD FE 0E 11 5C   0C 21 64 0C CB E9 CD FE
0DB0    0E CB 71 28 17 06 04 21   6E 0C 36 00 23 10 FB 11
0DC0    6E 0C 21 5C 0C CD FE 0E   CB D1 18 0F C5 01 04 00
0DD0    21 5C 0C 11 6E 0C CD B0   C1 CB 91 11 60 0C 21 5C
0DE0    0C CD FE 0E CB 71 C2 3B   0E 11 58 0C 21 68 0C CB
0DF0    A9 CD FE 0E CD 18 0F 2A   6C 0C 22 18 0C CB 99 1E
0E00    04 06 CC CD 69 00 CB 59   20 19 30 17 FE 1F 20 04
0E10    CB D9 18 0F FE 1D 28 08   FE 3A 30 07 FE 30 38 03
0E20    CD 3B 01 10 DE 1D 20 D9   2A 18 0C 22 6C 0C CB 61
```

Crash landing......continued

```
0E30        C2 75 0D CB 59 C2 38 0D    C3 75 0D 3E 1E CD 3B 01
0E40        21 0D 08 22 18 0C EF 59    4F 55 27 52 45 20 44 4F
0E50        57 4E 21 20 59 4F 55 52    20 4C 41 4E 44 49 4E 47
0E60        20 54 4F 4F 4B 00 2A 18    0C 36 20 00 21 4F 09 22
0E70        18 0C EF 53 50 45 45 44    20 41 54 20 49 4D 50 41
0E80        43 54 20 57 41 53 00 2A    18 0C 36 20 11 2F 08 21
0E90        5A 0C CD 53 0F EF 73 65    63 2E 00 2A 18 0C 36 20
0EA0        11 68 09 21 70 0C CD 53    0F EF 66 2F 73 00 2A 18
0EB0        0C 36 20 76 04 0E 11 58    0C 21 68 0C CB A9 CD FE
0EC0        0E CD 18 76 2A 6C 0C 22    18 0C CB 99 1E 04 06 00
0ED0        CD 69 00 CB 59 20 15 30    13 FE 1F 20 04 CB D9 18
0EE0        0B FE 3A 30 76 FE 30 38    03 CD 3B 01 10 E2 1D 20
0EF0        DD 2A 18 0C 22 6C 0C CB    61 C2 5E 0D CB 59 CB B1
0F00        06 04 AF 1A CB 69 28 03    9E 18 01 8E 27 12 23 13
0F10        10 F1 CB 7F C8 CB F1 C9    21 0A 09 06 30 36 20 23
0F20        10 FB 11 0D 09 21 5A 0C    CD 53 0F 11 18 09 21 62
0F30        0C CD 53 0F 11 26 09 21    70 0C CD 53 0F 11 32 09
0F40        21 52 0C CD 53 0F 21 24    09 CB 51 20 03 36 0B C9
0F50        36 5E C9 ED 53 18 0C 06    03 CB F9 7E CB 79 28 13
0F60        F5 E6 F0 20 0B F1 E6 0F    20 0E 05 20 12 04 18 08
0F70        F1 CB B9 CD 44 02 18 08    CD 4D 02 CB B9 18 01 04
0F80        2B 10 D8 2A 18 0C 36 20    C9 21 8E 0A 22 18 0C EF
0F90        4F 55 54 20 4F 46 20 46    55 45 4C 20 41 54 20 20
0FA0        20 20 20 20 73 21 00 11    9E 0A 21 5A 0C CD 53 0F
0FB0        C9 21 0B 0A 22 18 0C EF    42 75 72 6E 20 72 61 74
0FC0        65 20 6C 62 2F 73 00 2A    18 0C 36 20 C9 .........
```

****

I BUY MY LIGHT READING AT MICRODIGITAL

Fundamental Algorithms. Knuth £9 50
Fundamental alogrithms is the first volume of D. E. Knuth's classic series on computer programming. Prof. Knuth writes with styled and wit (among many memorable quotes is one from McCall's Cookbook!). The book begins with a through discussion of the mathematics used in computer programming followed by a treatment of information structures,stacks, arrays, linked lists, dynamic storage allocation and trees. This classic work belongs on the reference shelf of everyone seriously interested in computer science.

Software Tools. Kernighan and Plaugher. £7.20.
Good programming is not learned from generalities, but by seeing how significant programs can be made clean, easy to read, easy to maintain and modify, human-engineered, efficient and reliable, by the application of commonsense and good programming practices. Careful study and imitation of good programs leads to better writing. [illegible] introduction. With the same style and clarity that characterizes their highly acclaimed book The Elements of Programming Style the authors have written Software Tools to teach how to write good programs that make good tools. The programs contained in the book are not artificial, but are actual programs [illegible] tools such as a text editor, macro-processor, text formatter etc.

Program Design. B. W. Liffick £4.50.
This is the first title in BYTE magazines much heralded series on programming techniques. The text introduces the subject of program design. The most critical part of developing a program is the design phase. Here most fatal errors are introduced and program specifications forgotten. It is also during this phase that errors are least costly to fix (both in terms of money and time); specifications are easiest to change, and program integrity is simplest to insure. Structured programming techniques, decision tables and hashing techniques mean that not only does the program stand a better chance of running first time but programming becomes more fun.

A Microprogrammed APL implementation. R. Zaks £14.75.
This book presents the complete theory and design of an interpreter for APL. One of its essential characteristics is to be implemented within 2K of ROM. This was made possible by an in-depth analysis of the intrinsic syntactic characteristics of the language. This analysis resulted in such a terse interpreter. Additional RAM memory is needed to store least frequently used APL operators which are implemented as APL functions. The original implementation was for a D.SC Meta 4; complete listings of the interpreter with control words for the META 4, a Snobol version and ALGOL [illegible] of the Parser are included. In addition the author claims the program can be transcoded into any 'standard' instruction set such as an 8080, 6800 or other with reasonable ease. In addition, the reader should gain a thorough understanding of the conceptual and practical problems associated with the parsing, interpretation and execution of a powerful lambda language such as APL.

Z-80 Programming for Logic Design. A. Osborne et al. £5 95
Here's the book we've all been waiting for! It's third in the series of Osborne's programming for logic design books. Written for both programmers and logic designers, it explains how an assembly language program can replace non-programmable logic devices [illegible] with direct reference to the Z-80.

Programming proverbs. H. F. Ledgard. £5.56.
Programmers can and should write programs that work first time. This statement may sound idealistic to those accustomed to long hours of debugging. Yet it's the theme of this book — a unique collection of 'proverbs' or rules and guidelines for writing more accurate, error free programs. And like all proverbs they are compact gems of truth.

Microprocessor Systems Design. E. Klingman. £14.00.
Outstanding for its information on real microprocessors, this text is both an introduction and a detailed reference source treating over a dozen processors, including new third generation devices.

Microcomputer Primer. M. Waite. £6.35.
Here's a microprocessor tutorial for readers have some electronics background. Microcomputer primer concentrates on the hardware of microcomputers (although there are sections dealing with software.

How to profit form your personal computer. T. Lewis. £6.36
If you have ever felt you need some help planning programming techniques for common business applications, this book is for you. Enjoyable and readable, it contains suggestions for accounting, payroll handling, inventory management and sorting mailing lists. Many terms and notations are explained. Sample programes in BASIC, the use of blueprints to design program structure, and a full glossary of terms are a few of this book's special features.

Basic with business applications. R. W. Lott £8 40
This book focuses on the BASIC language and its application to specific business problems. Topics include interest rate calculation, break even analysis, loan rates and depreciation.

How to program microcomputers. W. Barden £7 00
Here is a guide to assembly language programming of the Intel 8080, Motorola MC6800, and MOS 6502 microprocessors. It is written especially for beginning programmers with hobby's microcomputers based on one of these three chips.

Game playing with BASIC. D. Spencer £5.56
You'll enjoy the challenge of competing with your own computer. Games described include [illegible] Tac Toe, Nim, Roulette, Slot machines, Magic squares, Keno, etc. The style is non technical and each section gives complete rules for the game, how it works, illustrative flowcharts and example output for each program.

Computer lib/dream machines. T. Nelson £5.95
Written for the layman — the person who is intelligent [illegible] computers. It's written and self published by a philosopher who is also [illegible] and an excellent teacher of basic concepts. (For those who have [illegible] are constructed [illegible] real and substantial white bricks). Computer Lib/Dream Machines [illegible] the beginner and is also a refreshing self examination for the old hands [illegible].

Superwumpus. J. Emmerichs £4.80.
Superwumpus is an exciting computer game [illegible] in BASIC and 6800 assembly language. Not only addictively fun, Superwumpus is also [illegible] unusual data structures (the tunnel and cave systems of Superwumpus [illegible]).

Small Computer Systems Handbook. S. Libes £5 [illegible]
One of the best introductory texts we have seen. [illegible] systems handbook provides the necessary background in digital logic [illegible] and computer hardware and software basics. Only a minimal [illegible] required. Further the book offers an introduction to programming on the machine level [illegible] languages such as BASIC.

A Dictionary of Microcomputing. P. Burton £[illegible]
This is one of the best designed and [illegible] related terms yet seen on the market.

Computer resource book — algebra. T. A. Dwyer £4 [illegible]
A new way to learn about algebra with [illegible] the basic language and includes listings of over 60 programs for [illegible] equations, random number generation.

Home computers: a beginners glossary and guide. M. [illegible] and [illegible] Sippl £4 95
A general introduction to the personal [illegible] glossary.

Computer Science — A first course. Forsythe et al. £14 [illegible]
This is one of the most complete computer [illegible] book for the serious experimenter who wants a definitive treatment [illegible] development.

Computer Science — Projects and study problems. Forsythe et al. £[illegible]
A companion text to the above [illegible] equations, the eight queens problem etc.

Consumers guide to personal computing. S. Freiberger and P. Chew £4 [illegible]
Computer power and purchasing power [illegible] the new world of the personal computer. Here are [illegible] principles of microcomputers and a review of 64 [illegible].

Finite State fantasies £2.00
A collection of cartoons illustrating [illegible].

The new hobby computers £3.95
This book is a new introduction to [illegible] help the neophyte as well as the [illegible].

Hobby computers are here. £3 95
With over five thousand copies [illegible].

How you can learn to live with computers. H. Kleinberg £[illegible]
The perfect book for the layman [illegible].

Mondeb. D. Peters. £4.00
Another in the series of Byte Books [illegible] illustrates all the general features of Motorola's MIKBUG [illegible]. While extremely versatile, ease of use was a prime design [illegible] was minimum memory requirements while retaining maximum [illegible] is less than 3K in size. Command capabilities include [illegible] interrupt setting for debugging, memory testing, search [illegible].

Scelbi "8080" Gourmet Guide and cookbook £7 9[illegible]
A complete guide to programming the 8080 [illegible] general purpose routines for floating point arithmetic with 23 [illegible] random number generation, input/output processing [illegible]. A mine of hard to find information.

Basic Computer Games. Ahl et al. £5 50
Complete listings of 102 computer games [illegible] listings, description and sample runs.

Cheap Video cookbook. D. Lancaster £5 10
Here is a brand new, super low cost way [illegible] computer and [illegible] an ordinary set with minimum [illegible] inverse fonts and both construction details with [illegible].

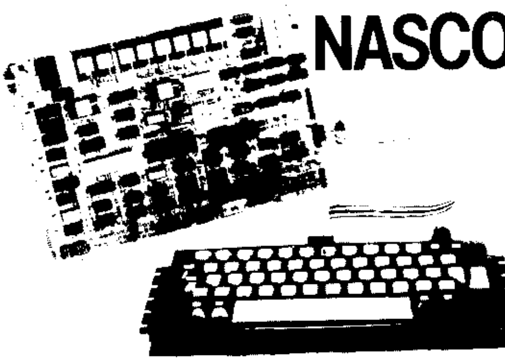
Microprocessor interfacing techniques. R. Zaks £7 95
Interfacing is no longer an art [illegible]. This book will teach you how to interconnect a complete system and interface [illegible] covers hardware and software skills and techniques including the use [illegible] buses such as the IEEE 488 or S100.

Programming a microcomputer: 6502. C. H. Foster £7 96
This informal, well written book [illegible] computers. Foster uses the 6502 microprocessor used in the [illegible] KIM, PET etc. as a basis for discussing the techniques of writing programs [illegible] computers.

TTL Data Book £5 00
The classic reference work published by Texas [illegible] TTL. The method of finding out about the [illegible] with [illegible] numbers.

To provide a better service we have re-organised ourselves.
The following remain the same:—
Our shop is at 25 Brunswick Street, Liverpool.
Telephone order number 051-236 0707
The following are changed:—
Mail order, software, engineering, accounts, etc., are now at :—
14 Castle Street, Liverpool.
Telephone (except telephone orders) 051-227 2535/6/7.

Just a little bit more... NASCOM-2



Compare its features:

- Z-80A 4MHZ. CPU: The most powerful 8-bit processor on the market
- 8K Basic: resident on board, MICROSOFT Basic, the industry standard, with extensions for on-screen editing, graphics, machine code interfacing. Optimised for speed (see benchmarks below).
- Full 57 Key Licon solid state keyboard: switch mechanisms are contactless, high reliability professional units for long trouble free life. Keyboard is mounted separately to avoid straining main P.C.B.
- Total of 20K on-board memory: 2K monitor (Nas-Sys 1), 1K Video RAM, 1K Work space RAM. 8K Microsoft Basic, 8K user RAM.
- Kansas City cassette interface: for reliable storage of programs and data at 300 or 1200 baud, with full checksum error detection.
- Nas-sys monitor: A powerful 2K machine code monitor provides an ideal environment for learning about and developing machine code programs. Nas-sys uses a blinking non destructive cursor, with 22 commands. ASCII terminals are fully supported via the serial interface; users can add their own I/O drivers via the system I/O vector table to support other devices.

Nas-sys commands are:

A—Hex arithmetic	N—return to normal
B—set breakpoint	O—Output to P I O
C—Copy	Q—Query input port
E—Execute	R—Read tape
G—Generate	S—Single step
H—Operate as half duplex. terminal	T—Tabulate memory
I—intelligent copy	U—activate user I O drivers
J—Execute at FFA	V—Verify tape
K—set keyboard options	W—Write tape
L—load from tape	X -set external device
M—Memory modify	Z—execute at FFD

- On board P.I.O. — An uncommited P.I.O. (MK 3881) giving 16 programmable I O lines with handshake.
- On board RS-232-Will interface directly into any standard teletype — allowing use of BASIC or Nas-sys from the teletype.
- Full on-screen editing: a complete screen editor with cursor movement (UP. DOWN. LEFT. RIGHT). Insert and delete, backspace etc
- Screen display of 16 lines x 48 characters: Stable, clear display to British television standards. Full 128 ASCII character set: option for further 128 graphics characters
- Fully buffered NASBUS compatible: Well defined bus structure with a range of expansion cards; including (shortly) a floppy disc system with CP m — the industry standard operating system.

	Nett	Vat	Total
Nascom-2	295.00	44.25	339 25
Power supply	24 50	3.68	28 18
10 C15 cassettes	4.44	0.66	5.10
Z-80 Programming manual (Mostek)			4 50
Z-80 Microcomputer handbook			6 95
Practical microcomputer programming the Z-80			20 00
Sargon-8K Z-80 Chess program (book)			9 50

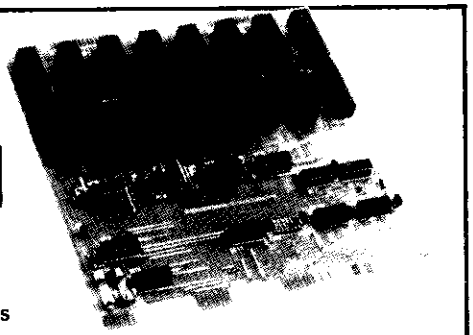
PERSONAL COMPUTER WORLD BENCHMARK TESTS

	APPLE II	NASCOM 2	RM. 380Z	PET
BM 1	1 5	1 1	1.4	1.7
BM 2	3.2	5 4	6.5	9.9
BM 3	7.3	11 1	13.2	18.4
BM 4	7.2	11.6	13.9	20.4
BM 5	8.9	12.6	15.0	21.7
BM 6	18.6	19.3	22.3	32.5
BM 7	28.2	27.6	31.6	50.9
BM 8		5.2	6.2	12.3

MICRODIGITAL

25 Brunswick Street, Liverpool L2 0BJ
Tel: 051-236 0707 (Mail Order) 051-227 2535 (All other Depts)

New for Nascom 1 from Microdigital



Put your Nascom to work with the new Microdigital Relay Board.

- 16 Reed Relays, totally isolated 200 mA. 50 V.D.C. 5 W max each. Operate and release time 1 ms (including bounce).
- Single sided, glass fibre board, with gold plated edge connectors and silk screened component layout.
- Plugs directly into Nasbus, does not interfere with normal Nascom operation, all interupt and D.M.A. Daisy Chain Links carried on. Draws only 250 mA from each of the +5 and -12V Rails.
- All components supplied, all IC's socketed, easy to build, and easy to program in Basic or Machine Code.
- Occupies 2 consecutive ports, link selectable — several boards can be used on one Nascom System.
- Output is via front edge connector on 0.1" centres. Uses standard edge connectors for connection to controlled devices.
- Complete manual with sample software.

- Light displays
- Industrial process control.
- Model Railway Control.
- Pre programmed music generation.
- Robots, Central Heating Systems.
- Stepping Motors.
- . . . . . ?

PRICE £49.95 plus V.A.T. (Total Cost £57.44)
Access, Barclaycard, Mail Order.

MICRODIGITAL

25 Brunswick Street, Liverpool L2 0BJ
Tel: 051-236 0707 (24 hour Mail Order)
051-227 2535 (All other Depts.)
Mail orders to: MICRODIGITAL LIMITED,
FREEPOST (No Stamp Required)
Liverpool L2 2AB.

Pico Pilot

Mark Stevenson

Pico-pilot is a high level language interpreter/editor written for the Nascom in Z80. code to run in 256 bytes. The language is a small subset of the PILOT language which is used for educational purposes.

The program to be interpreted is stored as source characters and the interpreter moves through the text and obeys the instructions found there. The source text may be visually edited on the VDU and then the program executed.

Editing, entering, displaying and runing programs

1. Entering source text.

When the interpreter is entered the origin prompt is printed(@) and it then waits for input. The prompt indicates that the text pointer is now pointing to the first text buffer location. Any characters may now be entered into memory by just typing them in.

Note. The backspace key can be used to correct errors on input.

When the program has been entered an end of file (EOF) marker '&' (shift 6) should be placed in store to indicate the last text position used, followed by new line. Shift backspace may now be used to return the text pointer to the start of the program and display *

Care should be taken not to type random characters now as they will overwrite the start of the program.

Several keys have functions that will not corrupt the entered text.

2. Running programs.

To execute a program a $ (shift 4) is entered which clears the screen resets the pointer to the start of the program and starts to interpret the instructions stored. If the program gets locked into a loop which does not contain an ACCEPT statement (see later) the only way to exit is by re-set and the interpreter must be executed again. ie. E0C50.

If the loop does have an ACCEPT statement then exit can be made by the use of shift/backspace. The origin prompt (@) will be displayed and editing can take place.

3. Command keys and editing.

Typing/will display the next, or first line of text up to a NL. leaving the cursor at the end of the line. Single character errors may be corrected by backspacing to the character in error. All the characters are saved in memory but are lost on the screen. Backspace over the incorrect character then type in the new one. Pushing / will then retrieve the lost text back from memory.

The / is displayed on the screen but is not stored in memory.

Note. The text is stored in consecutive memory locations so characters may not be added in the middle of the current line. If a new line is to be added it must be shorter or the same length as the one it is replacing.

To change a line, list the program using / to the line before the one you wish to replace. Then type NL. This will place the cursor at the beginning of the line you wish to replace and the new text can now be typed in. Terminate the text with a '%' (shift 5) instead of NL.

The interpreter will display the text not overwritten, then delete it and consolidate the following lines thus saving memory.

If any text is to be added at the end of the program then this is done by overwriting the EOF marker and placing a new EOF at the end of the new text.

4. Error detection.

Although this is a very small interpreter, certain errors are detected and flagged. If a line starts with a non valid character an error indicator will be printed along with the line at fault.

5. Notes to watch.

When the origin prompt is displayed be careful what you type.

Always remember the EOF marker.

Backspacing past the origin prompt will result in program crashes.

Instruction set

The instructions are of the form;

((program marker)) ((modifier)) (key letter):(parameter)

Objects in double brackets are optional.

Key letters and parameters.

T: (Text) The string of text is output to the VDU. It is terminated with NL.

Example T:This text is displayed on the VDU.

A:(Accept) The accept statement will take a single character from the keyboard after displaying a prompt. Program execution will then continue.

Example T:Please type in a letter. NL
A:NL

This will display on the VDU. Please type in a letter. ?__

E:(Exit) E halts execution of the program and returns the computer to command input mode.

R:(Remark) Any text may follow this key letter and is ignored by the interpreter. Used for program comments.

S:(sub-routine) S followed by a single character which is passed by the accumulator to a machine code held in memory. The sub routine may alter all the registers except DE. A return instruction C9 may be used to get back to Pico-Pilot.

M:(Match) M followed by a single character which is compared with the last character input by an accept statement.

The test flag is set to 'Y' if the characters are identical or 'N' if different. The flag can be used for conditional statements or jumps.

Example T:Please input the letter B NL

A: NL

M:B NL

This would set the match flag to 'Y' if a B was pressed or 'N' if a different character was pressed.

The character input by an accept is not lost so several matches can be made.

J:(Jump) J followed by a single digit will alter the execution of the program. The program jumps to a marker (*)

The marker is specified by the digit. Maximum of +9.

The instructions stored are executed sequentially unless a jump statement is encountered. The jump may tak

one of three forms;

J:d The text pointer is moved forward 'd' program markers and execution continues from there.

J:-d The text pointer is moved back 'd' program markers.

J:Ø The text pointer moves back to the last accept statement and repeats the prompt. ? This can be use

if an error is detected on input.

Example A: NL

T:An endless loop NL

J:Ø

This will continuously ask for input and print on the VDU

An endless loop

?-

The star '*' character may be used as a label or marker within a Pico Pilot program to indicate a start point t be used by a jump instruction. It may preceed any command character and there may be any number of the within a program. The jump statement however is limited to +9 program markers. This can be extended by jumping to another jump command.

In addition to a program marker, an instruction may also be preceeded by a modifier. (character N or Y)

This is the key to conditional programing. After a match is encountered the match flag is set and remains set until another match statement.

If the match was correct ie. the input character was the same, then any command preceeded by a Y will be executed and any preceeded by an N will not. If the match failed the only statements preceeded by N will be executed. Any statements not preceeded by a modifier will be executed in both cases.

Program Listing

```
0C50      31 00 10 EF 1E 00 2A 52    0D EF 40 00 CD 95 0C FE
0C60      1E 28 F3 FE 24 28 34 FE    1D 20 03 2B 18 EE FE 2F
0C70      20 05 CD 3D 0D 18 E5 FE    25 20 16 D5 E5 54 5D CD
0C80      3D 0D 3E 26 ED A0 BE 20    FB ED A0 ED A0 E1 D1 18
0C90      CB 77 23 18 C7 CD 3E 00    C3 3B 01 2A F2 0D EF 1E
0CA0      00 2B 23 7E FE 28 FA A2    0C FE 59 28 04 FE 4E 20
0CB0      0B BA 28 EE 3E 1F 23 BE    20 FC 18 E6 FE 41 20 14
0CC0      22 50 0D EF 3F 00 CD 95    0C FE 1E 28 89 5F 23 EF
0CD0      1F 00 18 CE FE 52 28 DC    FE 4D 20 0C 23 23 7E 16
0CE0      59 BB 28 BE 16 4E 18 BA    FE 4A 20 26 23 23 AF 4F
0CF0      7E FE 2D 20 03 4F 23 7E    E6 0F 47 20 05 2A 50 0D
0D00      18 A1 79 B7 28 02 2B 2B    23 7E FE 2A 20 F4 10 F2
0D10      18 90 FE 45 CA 56 0C FE    53 20 10 23 23 7E E5 21
0D20      27 0D E5 2A 54 0D E9 E1    C3 A2 0C FE 54 20 07 23
0D30      23 CD 3D 0D 18 99 EF 45    21 09 00 18 F4 0E 40 0D
0D40      CA 3B 01 7E CD 3B 01 23    3E 1F BE 20 F2 C9 00 00
0D50      00 00 56 0D 50 0C .....
```

Points to note within the program

Address	Label	Comment
0D50	LST.	Initially set to 00 00 but during execution it is used to hold the address of the last accept statement. Useful to find how much user ram is left.

0D52 LOC. Initially loaded with 0D 56. this is the pointer to the start of text area. It may be modified to accomodate a sub-routine.

0D54 SUB. On SUB the program executes routine pointed to at this address ie set to 0C50 but this can be altered.

0C50 INIT Moves the stack pointer to 1000 but for those with extended machines this address contents may be changed.

Abbreviations

Direct commands

T: Tabulate A: Accept M: Match J: Jump

S: Sub routine R: Remark E: Exit

Modifiers

*__: Line marker Y__: True match N__: False match

Pico Pilot example program

The following example is designed to show the various ways in which a pilot program is constructed. The line numbers are only there for guidance and should be included in the program. Load program then E0C50.

Line no. (reference only)

```
10     *T:THIS IS A PROGRAM BASED ON PICO PILOT  NL
20     T: (sp) NL
30     T: (sp) NL
40     T:I AM GOING TO ASK YOU A QUESTION NL
50     T:DO YOU WISH TO CONTINUE Y OR N NL
60     A: NL
70     M:Y NL
80     NJ:-1 NL
90     *YT: (sp) NL
100    YT:WHAT IS TWO PLUS TWO NL
110    A: NL
120    M:4 NL
130    YJ:2 NL
```

```
140     NM:3 NL
150     YJ:1 NL
160     NM:5 NL
170     YJ:1 NL
180     NT:YOU SHOULD NOT HAVE LEFT SCHOOL  NL
190     NT: 1+1=2  1+2=3  So 2+2=        NL
200     NJ:-1 NL
210     *T:you are very close try again NL
220     J:-2 NL
230     *T: CORRECT . . . TOP OF THE CLASS NL
240     T: DO YOU WANT TO TRY AGAIN Y OR N NL
250     A: NL
260     M:Y NL
270     YJ:-4 NL
280     NE:
290     EOF. (shift 6)              (sp) = space          NL = New line.
```

Type in the program as it is written ignoring the line numbers.

You can either list the program line by line by keying / or you can execute the program by keying the dollar sign (shift 4)

On execution you should get on the VDU;

```
This is a program based on Pico pilot.
I am going to ask a question
Do you wish to continue Y or N
?_
```

At this point the program has reached line no. 60 and the keyboard is being scanned for input.When a key is pressed a match is made at line 70 and depending on the result either at line 80 the program jumps back to the first question or at line 100 another question is asked.

The first section of this routine shows how programs are built up. By executing the rest of the program you will get an idea of how Pico Pilot works. Try answering incorrectly and see what happens.

Although this program is very short, it shows that on a basic Nascom, an interesting and amusing result can be achieved.

Messagewriter

Steven Abrams

This program is used to display a message or advert on the screen. The length of the message is limited only by the amount of RAM available.

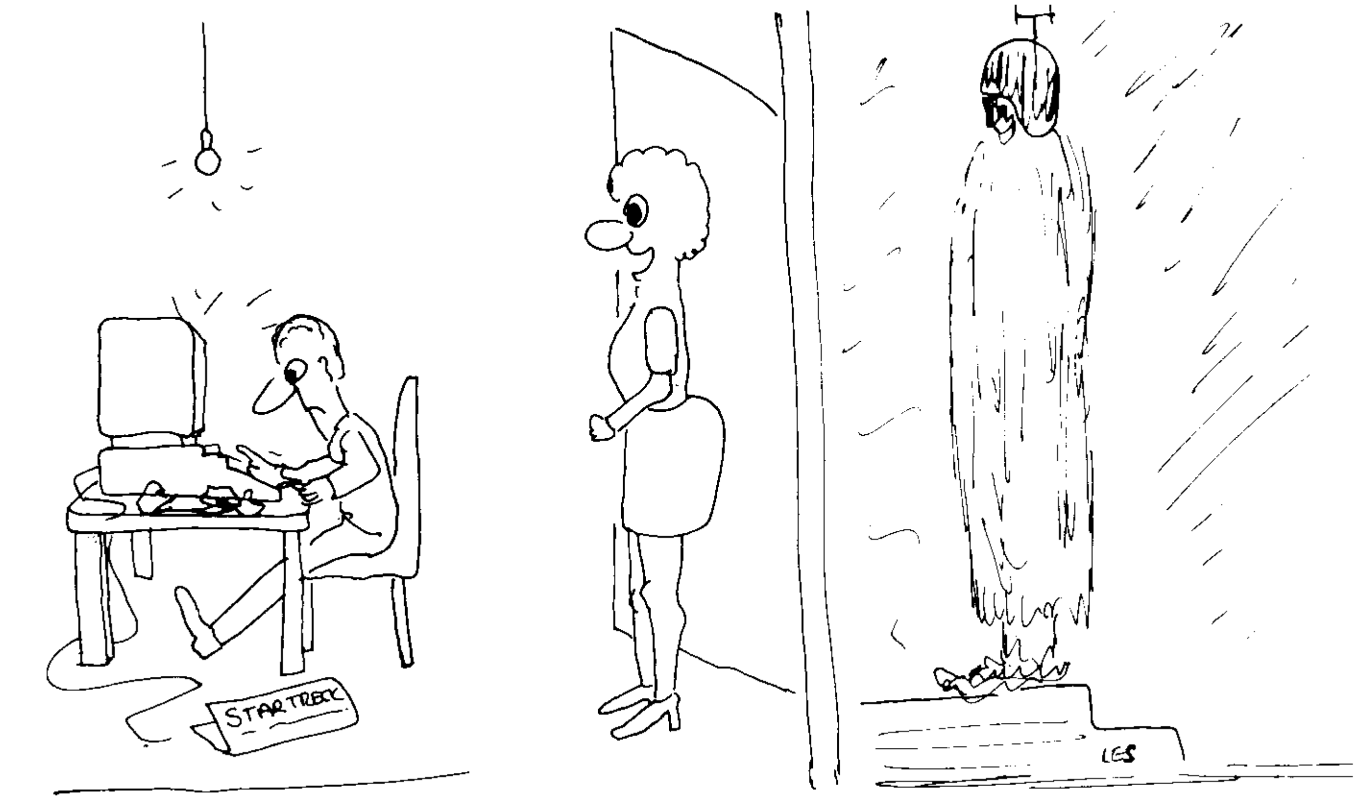
To enter the message into RAM, execute the program at 0C81, then type out the message. All keyboard keys and functions may be used.

At the end of the message type newline followed by @. The message will now start to scroll up on the screen a line at a time. The speed may be altered by changing the value of 0C6C for fine adjustment and 0C6A for coarse adjustment.

Once the message has scrolled through one complete run, it can be extended by pressing any key to interupt the program, then typing the additional message. Pressing newline then @ will restart the scrolling.

To write a new message, reset and execute from 0C81. To run the same message after a reset, execute at 0C52.

```
0C50    D1 E5 21 92 0C 7E FE 40   28 F6 CD 3B 01 FE 1F 28
0C60    08 CD 69 00 38 18 23 18   EC 0E 01 06 A0 CD 35 00
0C70    CD 69 00 38 09 10 F6 00   0D 20 F0 23 18 D7 E1 18
0C80    03 21 92 0C CD 3E 00 CD   3B 01 77 FE 40 28 C2 23
0C90    18 F2 ..........
```



"GRAHAM' THER'S ASTRANGE MAN AT THE DOOR WHO SAYS WILL YOU STOP MESSING ARROUND WITH HIS **KLINGONS**!.."

3D Noughts and crosses

Graham Houghton.

The combination of winning lines in this game is quite large and the skill of this game comes more from recognising when your opponent is about to win rather than winning yourself. The program is written to run in a basic Nascom and unfortunately there was not enough memory to write a routine to recognise when somebody had won so it is left to the players to decide.

The program executes at 0C50.

```
0C50    C3 CD 0E 21 8D 08 CD 6E    0C 21 CD 08 CD 6E 0C 21
0C60    0D 09 CD 6E 0C 21 4D 09    CD 6E 0C 18 1A 00 0E 04
0C70    06 04 3E 2E 77 23 23 10    FB 06 04 23 10 FD 3E 00
0C80    B9 C8 0D 18 EB 00 00 21    CB 0B 22 18 0C EF 20 20
0C90    20 33 44 20 4E 4F 55 47    48 54 53 20 41 4E 44 20
0CA0    43 52 4F 53 53 45 53 20    34 20 58 20 34 20 58 20
0CB0    34 20 4D 41 54 52 49 58    2E 00 2A 18 0C 36 20 21
0CC0    4D 08 22 18 0C EF 4C 45    56 45 4C 20 31 20 20 20
0CD0    20 20 4C 45 56 45 4C 20    32 20 20 20 20 20 4C 45
0CE0    56 45 4C 20 33 20 20 20    20 20 4C 45 56 45 4C 20
0CF0    34 00 2A 18 0C 36 20 21    0F 0A 22 18 0C EF 20 20
0D00    20 20 20 20 20 20 20 20    20 20 20 20 20 20 4C 45
0D10    56 45 4C 20 20 20 3F 00    2A 18 0C 36 20 11 37 00
0D20    19 22 18 0C EF 52 4F 57    20 20 20 20 20 3F 00 2A
0D30    18 0C 36 20 19 22 18 0C    EF 43 4F 4C 55 4D 4E 20
0D40    20 3F 00 2A 18 0C 36 20    11 40 00 21 27 0A 06 03
0D50    22 18 0C CD 69 00 30 FB    CD 3B 01 FE 31 CA 77 0D
0D60    FE 32 CA 77 0D FE 33 CA    77 0D FE 34 CA 77 0D FE
0D70    45 CA 57 0E C3 7B 0E 19    00 10 D5 11 27 0A 1A FE
0D80    31 CA 93 0D FE 32 CA 98    0D FE 33 CA 9D 0D FE 34
0D90    CA A2 0D 21 4C 08 18 0F    21 58 08 18 0A 21 64 08
0DA0    18 05 21 70 08 00 00 CD    F0 0E 1A D6 30 47 11 40
0DB0    00 19 10 FD 11 A7 0A 1A    D6 30 47 2B 23 23 10 FC
0DC0    7F FE 2E 28 03 C2 20 0E    C3 CB 0D CD E1 0D C3 BF
```

```
0DD0        0C 3E 00 77 21 CC 0D 3E    E1 77 21 6B 0A 3E 18 77

0DE0        C9 3E 18 77 21 CC 0D 3E    D1 77 21 6B 0A 3E 18 77

0DF0        C9 0D C6 0D C6 0D C6 0D    C6 0D C6 0D C6 0D C6 0D

0E00        C6 0D C6 0D C6 0D C6 0D    C6 0D C6 0D C6 0D C6 0D

0E10        C6 0D C6 0D C6 0D C6 0D    C7 0D C6 0D C6 0D C6 0D

0E20        21 13 0B 22 18 0C EF 20    53 4F 52 52 59 21 20 54

0E30        48 49 53 20 53 51 55 41    52 45 20 49 53 20 4F 43

0E40        43 55 50 49 45 44 20 2A    2A 2A 00 2A 18 0C 36 20

0E50        CD B2 0E C3 BF 0C FB 21    13 0B 22 18 0C EF 20 47

0E60        41 4D 45 20 54 45 52 4D    49 4E 41 54 45 44 21 00

0E70        2A 18 0C 36 20 CD B2 0E    C3 50 0C 21 13 0B 22 18

0E80        0C EF 20 53 4F 52 52 59    21 20 54 48 49 53 20 49

0E90        53 20 4E 4F 54 20 41 20    56 41 4C 49 44 20 4D 4F

0EA0        56 45 20 2A 2A 2A 00 2A    18 0C 36 20 CD B2 0E C3

0EB0        BF 0C 0E 02 3E 00 06 FF    CD 35 00 10 FB 0D B9 20

0EC0        F5 21 13 0B 3E 20 06 30    77 23 10 FC C9 EF 1E 00

0ED0        21 6B 0A 22 18 0C EF 20    20 54 4F 20 50 4C 41 59

0EE0        21 00 2A 18 0C 36 20 21    6B 0A 3E 00 77 C3 53 0C

0EF0        06 A0 CD 35 00 10 FB C9    .......................
```

****

Income Tax Steven Abrams

Before executing this program, modify 0C3D to 33, and 0C3E to 0C. If you should press reset, this must be repeated. Without this modification, the answer will not be accurate.

Execute program at 0C50

Enter the tax week as a two figure number. ie. week 5, enter 05

Next enter the taxable pay (maximum £99,999.99) using the full stop for the decimal place. When the second figure after the decimal place has been entered, the program will calculate the total tax due and print it out on the screen.

Should you get an error message, or to enter another figure for the same tax week, press 1.

To change the week number press 0. A new week number can now be entered.

For the purpose of this program, taxable pay is the total pay to date after deducting free pay.

The method of calculating free pay from a tax code with a suffix letter that is approved by the Inland Revenue is as follows.

Code 191H, Week 24

Change the suffix letter to a 9, then divide by 52. ie. code 191H becomes £1919. Divided by 52 becomes £36.903846. Round the answer up to the nearest 5 pence, then multiply by the week number ie. £36.903846 becomes £36.95 x 24 £886.80 free pay. This figure is then deducted from the total pay to date and the answer entered as taxable pay.

The rates of tax in this program are those current from 6th October 1979.

In the event of changes in the tax rates, a list of amendments is available. Please send S.A.E. and 50 pence to the address on the page 64.

The program uses 30 bytes for workspace starting at 0FDB. The modification mentioned above moves the stack so that it will not corrupt the program. Should your monitor set the stack at an address higher than 1000H, it will not be necessary to move the stack.

```
0C50   11 1A 09 21 DB 0E 01 08   00 ED B0 06 02 13 CD 3E
0C60   00 CD C3 0C 20 F8 12 10   F4 EB D6 30 06 0A 2B 86
0C70   D6 30 10 FB 32 DC 0F 08   06 03 21 DC 0F CD D3 0C
0C80   10 FB 11 5A 09 21 E3 0E   0E 0D ED B0 06 06 CD 3E
0C90   00 CD C3 0C 20 F8 12 13   FE 2E 28 04 10 F0 18 3D
0CA0   06 02 CD 3E 00 CD C3 0C   20 F8 12 13 FE 2E 28 2D
0CB0   10 F0 1B 1B 1B 1B 21 E2   0F EB 3E 23 BE 28 68 ED
0CC0   A8 18 F9 C5 01 39 0A B9   28 07 FE 2E 28 03 0D 10
0CD0   F6 C1 C9 C5 06 0A 23 36   30 10 FB C1 C9 11 9A 09
0CE0   21 F0 0E 01 0D 00 ED B0   C3 AB 0E 21 DD 0F 11 E7
0CF0   0F 06 06 1A BE D8 C0 23   13 10 F8 C9 D5 C5 06 08
0D00   1A 86 27 D6 30 CB 77 28   05 D6 10 2B 34 23 77 2B
0D10   1B 10 ED C1 D1 C9 D5 08   47 08 21 F0 0F CD FC 0C
0D20   10 F8 CD EB 0C D1 C9 11   0E 0F CD 16 0D 38 79 CD
```

```
0D30   C8 0E 11 16 0F CD 16 0D   38 76 CD C8 0E 11 1E 0F
0D40   CD 16 0D 38 73 CD C8 0E   11 26 0F CD 16 0D 38 70
0D50   CD C8 0E 11 2E 0F CD 16   0D 38 6D CD C8 0E 11 36
0D60   0F CD 16 0D 38 6A CD C8   0E 11 3E 0F CD 16 0D 38
0D70   67 CD C8 0E 11 46 0F CD   16 0D 38 64 CD C8 0E 11
0D80   4E 0F CD 16 0D 38 61 CD   C8 0E 11 56 0F CD 16 0D
0D90   38 5E CD C8 0E 11 5E 0F   CD 16 0D 38 5B CD C8 0E
0DA0   11 66 0F CD 16 0D 18 58   11 36 30 21 77 0F 18 56
0DB0   11 36 30 21 7F 0F 18 4E   11 35 35 21 87 0F 18 46
0DC0   11 35 30 21 8F 0F 18 3E   11 34 35 21 97 0F 18 36
0DD0   11 34 30 21 9F 0F 18 2E   11 33 30 21 A7 0F 18 26
0DE0   11 33 30 21 AF 0F 18 1E   11 32 35 21 B7 0F 18 16
0DF0   11 00 00 21 BF 0F 18 0E   11 00 00 21 C7 0F 18 06
0E00   11 00 00 21 CF 0F D5 E5   11 E6 0F 21 F0 0F 06 09
0E10   1A BE 28 0B 30 0D 96 27   D6 60 2B 34 23 18 07 3E
0E20   30 18 03 96 C6 30 12 2B   1B 10 E5 D1 08 47 08 D5
0E30   21 FA 0F CD FC 0C D1 10   F6 D1 21 E6 0F CD D3 0C
0E40   D9 06 07 D9 ED 53 EF 0F   01 E4 0F 0A 11 F0 0F 21
0E50   FA 0F FE 30 28 08 F5 CD   FC 0C F1 3D 18 EE D9 10
0E60   02 18 12 C5 06 0A 11 E7   0F 21 E8 0F ED A0 10 FC
0E70   C1 D9 0B 18 D6 21 F8 0F   7E FE 35 38 04 36 35 18
0E80   02 36 30 11 9A 09 21 FD   0E 01 09 00 ED B0 06 06
0E90   21 F2 0F 7E FE 30 20 08   23 10 F8 18 06 13 23 7E
0EA0   12 10 FA 3E 2E 12 13 0E   02 ED B0 CD 3E 00 FE 30
0EB0   20 09 21 1A 09 CD D3 0E   C3 50 0C FE 31 20 EC 21
0EC0   5A 09 CD D3 0E C3 78 0C   21 E6 0F 06 02 CD D3 0C
0ED0   10 FB C9 06 C0 36 20 23   10 FB C9 54 41 58 20 57
0EE0   45 45 4B 54 41 58 41 42   4C 45 20 50 41 59 20 23
0EF0   49 4E 56 41 4C 49 44 20   49 4E 50 55 54 54 41 58
0F00   20 44 55 45 20 23 2A 30   39 36 31 35 33 38 34 30
```

```
0F10   34 38 30 37 36 39 32 30    33 38 34 36 31 35 32 30
0F20   32 38 38 34 36 31 35 30    32 33 30 37 36 39 32 30
0F30   31 39 32 33 30 37 36 30    30 38 30 30 30 30 30 30
0F40   30 31 34 34 32 33 30 30    30 30 30 30 30 30 30 30
0F50   30 30 30 30 30 30 30 30    30 30 30 30 30 30 30 30
0F60   30 30 30 30 30 30 30 30    30 30 30 30 30 30 30 2A
0F70   30 34 38 37 37 34 30 34    30 31 39 39 32 37 38 39
0F80   30 31 34 36 33 39 34 33    30 30 39 38 33 31 37 34
0F90   30 30 37 32 33 35 35 39    30 30 35 36 39 37 31 32
0FA0   30 30 32 33 32 37 38 39    30 30 30 33 36 30 35 38
0FB0   30 30 30 .......  30 in every byte up to......
0FD0   30 30 30 30 30 30 30 30    30 30 30 30.........
```

****

J. M. Stevenson. **Tiny Basic. V.2**

1. Introduction.

Tiny Basic V.2. requires 2k. bytes of ram from 1000h. to 17FFh.

To start the interpreter the command E1000 (top store) is entered.

ie. E1000 2FFF for an 8k ram expanded Nascom.

The interpreter will reply by clearing the screen and printing;

```
TBI-V2

OK

:_
```

This indicates that the interpreter is now waiting for input.

(The ':' is the tiny basic prompt.)

If a new program is to be entered type 'NEW nl'. This will clear any old program or accidental text after power up.

2. commands.

Full details of the commands are not given but sample programs and details of how this basic differs from other tiny basics are given later.

commands and their minimum abbreviations.

Direct commands.

RUN	R.	DUMP	D.
LIST		NEW	N.
LOAD	LO.		

Commands and functions.

PRINT	P.	FOR	F.
NEXT	.	STEP	S.
TO	T.	RETURN	R.
GOSUB	GOS.	STOP	S.
REM	REM	INPUT	IN.
IF	IF	GOTO	G.
OUT	O.	POKE	PO.
CLEAR	C.	LET	(Implied)
RND (x)	R.(x)		
ABS(x)	A.(x)		
IN(x)	I.(x)		
PEEK(x)	P.(x)		
SIZE	S.		
CODE(x)	C.(x)		

3. Operation notes.

While a program is running or being listed the space bar may be used as a pause switch. One push and the operation is suspended until the bar is pushed again. As an alternative, after hitting pause any other key will abandon the operation and tiny basic will return to the command mode.

To abandon input (when adding a line to the program or inputting data to a running program) shift-backspace may be used.

4. Extensions to standard Tiny Basic.

LOAD- Loads a program from tape in B-Bug format.

DUMP- Dumps a program to tape in B-Bug format

POKE ADDR,DATA- Places DATA in store at ADDR(decimal)

OUT PTNUM,DATA Outputs DATA to port number.

CLEAR Clears all the variables A to Z to zero.

PEEK(ADDR) Opposite to POKE.

IN(PTNUM) opposite to OUT

CODE(ADDR) This function calls a machine code routine in decimal. The sub-routine may perform any operation provided the register pairs BC and DE are not changed on return. The return address is on the stack so a RET C9 may be used to get back to basic. Whatever is left in the HL register is returned as the value of the function.

RND(O) The RND function normally returns a random number but if zero is used the function returns the remainder from the last division which may be used for modulus calculations.

Line Trace. A line tracer may be called into operation by the command 'POKE 3700,128' From then on each line executed has its number printed preceeded by a ! mark. Tracing ceases when the command 'POKE 3700,0' is entered.

Memory Map.

0C50	0E73	Free space for machine code sub-routines.
0E74	0FFF	Stack and store used by Tiny Basic.
1000	17FF	Tiny Basic interpreter.
1800 top store.		Program store.
1800	1801	Pointer to top of prog. (set on execution)
0E7B	0EAE	Variable storage area. (2-bytes each) A to Z.
0EC0	0F2D	Buffer for input (110 characters)
0E79	0E7A	Pointer to top location to be used by TBI.
11C4		Call to tape read (normally CD 0C 07)
11D8		Call to tape write (normally CD 00 04)

These last two locations may be altered for use of load and dump routines in T2 format.

6. Useful notes.

This Basic is, bar minor differences, compatable with C C SOFT Level B and also Nascom B Basic. However the program text is stored as follows;

C C SOFT	NASCOM	TBI-V2
From 1600	From 1000	From 1800

So to convert between basics the programs must be relocated using the I command of B.Bug. Also the first two bytes (which point to the end of the program) must be altered to suit.

A Nascom program might start;

Addr.	Data.
1000	EB 11 xx xx

This must be changed to;

Addr.	Data.
1800	EB 19 xx xx

Other differences.

1. TBI-V2 uses <>as not equals rather than ≠
2. Double and single quotes may be used in print.
3. Consecutive commas are not allowed in print.
4. Control codes may be generated in print statements by using↑

   Thus↑↑clears the screen (same as $ in Nascom Basic)
5. TBI-V2 uses the more standard form for POKE and OUT commands unlike C C Soft.

If you are using a monitor that can provide lower case characters these may be used in text strings however capitals must be used for commands. The command seperator is semi-colon or new line.

Parameters within a command use a comma as a separator.

```
Sample Program.              SHUFFLES AND DEALS CARDS.
  10  FOR Y = 1TO 52
  20  @(Y)=Y;NEXT Y
  30  REM PACK FILLED
  50  FOR Y = 52 TO 1 STEP-1
  60  Z = RND (52)
  70  A = @(Y),@(Y) = @(Z), @(Z)=A
 100  NEXT Y
 105  INPUT "HOW MANY CARDS" X
 110  IF X>52  X = 52; REM NOTE NO 'THEN'
 115  IF X< 1 STOP; REM -VE NUM STOPS
```

```
120  PRINT 'HERE ARE YOUR CARDS'
125  FOR Y = 1 TO X
130  PRINT 'CARD NO.',Y,'IS',@(Y)
135  NEXT Y; GO TO 50
```

This produces a histogram on the VDU showing the randomness of the random number generator.

```
 5  PRINT ↑↑ ; REM CLEAR SCREEN
 7  FOR X = 0 TO 1000
20  Y = RND (14); REM FIND No. BETWEEN 1 AND 14
25  FOR Z = 0 TO 45
35  A = 2058 + (Y-1)*64 + Z; REM 2058 IS TOP LEFT
36  REM CALCULATE POSITION ON SCREEN
40  IF PEEK (A) = 32 POKE A,0; GOTO 90
41  REM BUILD HISTOGRAM ON SCREEN
45  IF PEEK (A + 1)= 33 POKE A, 33;GOTO 90
50  NEXT Z
60  POKE A,33
90  NEXT X
```

8. Editor option.

Rather than having to retype a line if an error is found the user may edit a line using the following routine (normally placed at 0C50)

```
0050   E5 D5 C5 12 3A BF 0E B7   20 16 11 C0 0E CD CD 15
0C60   CD 65 15 20 15 13 13 ED   53 75 0E 3E FF 32 BF 0E
0C70   2A 75 0E 7E 23 22 75 0E   18 02 3E 1F C1 D1 E1 C9
```

TBI-V2 should be modified as follows:

M17D8 CC 50 0C. nl (CALL Z, 0C50)

Now if a line number is entered and shift 6 (&) is hit one character from the old line will be entered onto the screen. More shift 6's will restore more characters and may be mixed with input from the keyboard.

BS. may be used to delete unwanted characters.

When the end of an old line is reached a 'nl' is automatically entered.

If no old line exists shift 6 gives a 'nl'. See if you can work out how to remember a line.

9 Stand alone routines.

1469	RNDS	Produces a random no. in HL between 1 and initial value in HL (not = 0 or -ve)
14B7	DIVIDE	Divides +ve HL by +ve DE. Result in BC, Remainder in HL. (DE not = 0)

```
1000    2A 0E 0C 22 79 0E 31 00    10 ED 5F 3C 32 77 0E AF
1010    32 74 0E EF 1E 54 42 49    2D 56 32 00 18 0B 21 06
1020    18 22 00 18 26 FF 22 02    18 31 00 10 21 33 10 22
1030    AF 0E 21 00 00 22 B5 0E    22 B1 0E EF 1F 4F 4B 1F
1040    00 3E 3A CD C5 17 D5 11    C0 0E CD CD 15 CD 1E 15
1050    7C B5 C1 28 49 1B 7C 12    1B 7D 12 C5 D5 79 93 F5
1060    CD 65 15 D5 20 10 D5 CD    7D 15 C1 2A 00 18 CD FF
1070    15 60 69 22 00 18 C1 2A    00 18 F1 E5 FE 03 28 A9
1080    85 5F 3E 00 8C 57 2A 79    0E EB CD EA 14 D2 56 15
1090    22 00 18 D1 CD 09 16 D1    E1 CD FF 15 18 A3 21 EE
10A0    16 CD 1E 15 D5 1A 13 FE    2E 28 12 23 BE 28 F6 3E
10B0    7F 1B BE 38 0E 23 BE 30    FC 23 D1 18 E4 3E 7F 23
10C0    BE 30 FC 7E 23 6E E6 7F    67 F1 E9 CD 25 15 C3 1E
10D0    10 CD 25 15 C3 29 10 CD    25 15 11 02 18 21 00 00
10E0    CD 6D 15 DA 29 10 ED 53    AF 0E 3A 74 0E 17 38 0A
10F0    13 13 CD B3 17 21 0A 17    18 A7 EF 21 00 C5 CD E0
1100    16 C1 18 EE CD 56 13 D5    CD 62 15 C2 F4 15 F1 18
1110    D5 CD CD 15 E5 21 FF FF    CD BA 15 2C 03 CD CD 15
1120    E3 CD 62 15 DA 29 10 E3    7C B5 CA 29 10 2B E3 CD
1130    E0 16 CD 5F 16 CD B3 17    CD 6D 15 18 E7 0E 08 CD
1140    BA 15 3B 05 EF 1F 00 18    A9 CD BA 15 1F 21 EF 1F
1150    00 18 8A CD BA 15 23 0D    CD 56 13 3E C0 A5 B4 C2
1160    F3 15 4D 18 05 CD 6D 16    18 10 CD BA 15 2C 05 CD
1170    0B 15 18 DF EF 1F 00 C3    06 15 CD 56 13 C5 CD A3
1180    16 C1 18 E6 CD 31 16 CD    56 13 D5 CD 65 15 C2 F4
1190    15 2A AF 0E E5 2A B1 0E    E5 21 00 00 22 B5 0E ED
11A0    73 B1 0E C3 E6 10 CD 25    15 2A B1 0E 7C B5 CA 2B
11B0    15 F9 E1 22 B1 0E E1 22    AF 0E D1 CD 16 16 C3 06
11C0    15 CD 25 15 CD 0C 07 18    12 CD 25 15 2A 00 18 22
11D0    0E 0C 21 00 18 22 0C 0C    CD 00 04 C3 29 10 21 7B
```

```
11E0    0E 06 34 AF 77 23 10 FC    C3 06 15 CD 31 16 CD F0
11F0    14 2B 22 B5 0E 21 82 17    C3 A1 10 CD 56 13 22 B9
1200    0E 21 88 17 C3 A1 10 CD    56 13 18 03 21 01 00 22
1210    B7 0E 2A AF 0E 22 BB 0E    ED 53 BD 0E 01 0A 00 ED
1220    5B B5 0E 60 68 39 18 01    09 7E 23 B6 28 18 7E 2B
1230    BA 20 F5 7E BB 20 F1 EB    21 00 00 39 44 4D 21 0A
1240    00 19 CD 09 16 F9 2A BD    0E EB C3 06 15 CD 87 15
1250    DA 2B 15 22 B3 0E D5 EB    2A B5 0E 7C B5 CA 2C 15
1260    CD EA 14 28 09 D1 CD 16    16 2A B3 0E 18 E8 5E 23
1270    56 2A B7 0E E5 7C AA 7A    19 FA 80 12 AC FA A3 12
1280    EB 2A B5 0E 73 23 72 2A    B9 0E F1 B7 F2 90 12 EB
1290    CD E4 14 D1 38 0F 2A BB    0E 22 AF 0E 2A BD 0E EB
12A0    C3 06 15 E1 D1 CD 16 16    C3 06 15 21 00 00 18 08
12B0    CD 56 13 7C B5 C2 F2 10    CD 7F 15 D2 E6 10 C3 29
12C0    10 ED 7B B3 0E E1 22 AF    0E D1 D1 D5 CD 6D 16 18
12D0    26 CD 87 15 38 16 CD 05    13 11 C0 0E CD 56 13 CD
12E0    25 15 D1 EB 73 23 72 E1    22 AF 0E D1 F1 CD BA 15
12F0    2C 02 18 D7 C3 06 15 D5    CD 87 15 DA 2B 15 43 D1
1300    CD 99 16 18 D1 C1 D5 EB    2A AF 0E E5 21 05 13 22
1310    AF 0E ED 73 B3 0E D5 3E    3F C5 C3 C5 17 CD B7 15
1320    2C 1C E5 CD 56 13 7D E1    61 4D ED 79 4C 18 21 CD
1330    B7 15 2C 0A E5 CD 56 13    7D E1 77 C3 06 15 C3 2B
1340    15 1A FE 1F 28 0A CD F0    14 CD BA 15 2C 02 18 F6
1350    C3 06 15 4A 4D 53 CD 9E    13 E5 21 90 17 C3 A1 10
1360    CD 89 13 D8 6F C9 CD 89    13 C8 6F C9 CD 89 13 C8
1370    D8 6F C9 CD 89 13 6F C8    D8 6C C9 CD 89 13 C0 6F
1380    C9 CD 89 13 D0 6F C9 E1    C9 79 E1 C1 E5 C5 4F CD
1390    9E 13 EB E3 CD E4 14 D1    21 00 00 3E 01 C9 CD BA
13A0    15 2D 05 21 00 00 18 27    CD BA 15 2B 00 CD D8 13
13B0    CD BA 15 2B 15 E5 CD D8    13 EB E3 7C AA 7A 19 D1
```

```
13C0     FA B0 13 AC F2 B0 13 C3     F3 15 CD BA 15 2D 8B E5
13D0     CD D8 13 CD CF 14 18 E1     CD 3A 14 CD BA 15 2A 29
13E0     E5 CD 3A 14 06 00 CD CC     14 E3 CD CC 14 EB E3 7C
13F0     B7 28 06 7A B2 EB C2 F4     15 7D 21 00 00 B7 28 2D
1400     19 DA F4 15 3D 20 F9 18     24 CD BA 15 2F 4C E5 CD
1410     3A 14 06 00 CD CC 14 E3     CD CC 14 EB E3 EB 7A B3
1420     CA F4 15 C5 CD B7 14 22     75 0E 60 69 C1 D1 7C B7
1430     FA F3 15 78 B7 FC CF 14     18 A1 21 60 17 C3 A1 10
1440     CD 87 15 38 05 7E 23 66     6F C9 CD CD 15 78 B7 C0
1450     CD BA 15 28 06 CD B7 15     29 01 C9 C3 2B 15 CD 50
1460     14 7C B7 FA F3 15 B5 28     1D D5 EB 2A 77 0E B7 CB
1470     74 28 01 3F CB 75 28 01     3F ED 6A 22 77 0E C5 CD
1480     B7 14 C1 D1 23 C9 2A 75     0E C9 CD 5C 14 1B CD CC
1490     14 13 C9 2A 00 18 D5 EB     2A 79 0E AF ED 52 D1 C9
14A0     CD 50 14 7E 6F 26 00 C9     CD 50 14 E9 CD 50 14 79
14B0     4D ED 68 26 00 4F C9 E5     6C 26 00 CD C2 14 41 7D
14C0     E1 67 0E FF 0C AF ED 52     30 FA 19 C9 7C B7 F0 7C
14D0     B5 C8 7C F5 2F 67 7D 2F     6F 23 F1 AC F2 F3 15 78
14E0     EE 80 47 C9 7C AA F2 EA     14 EB 7C BA C0 7D BB C9
14F0     CD 87 15 DA 2B 15 E5 CD     BA 15 3D 0D CD 56 13 44
1500     4D E1 71 23 70 C9 CD 0B     15 18 20 CD BA 15 3B 04
1510     F1 C3 F2 10 CD BA 15 1F     04 F1 C3 DD 10 C9 1A FE
1520     20 C0 13 18 F9 CD 1E 15     FE 1F C8 D5 EF 1F 57 48
1530     41 54 3F 1F 00 2A AF 0E     E5 7E 23 B6 D1 CA 29 10
1540     7E B7 FA C1 12 CD E0 16     C1 41 CD 99 16 EF 3F 00
1550     CD 5F 16 C3 29 10 D5 EF     1F 53 4F 52 52 59 1F 00
1560     18 D3 CD 25 15 7C B7 FA     F3 15 11 02 18 13 1A 1B
1570     87 D8 1A 95 47 13 1A 9C     38 04 1B B0 C9 13 13 1A
1580     FE 1F 20 FA 13 18 E6 CD     1E 15 D6 40 D8 20 18 13
1590     CD 50 14 29 38 5D D5 EB     CD 93 14 CD EA 14 DA 57
```

```
15A0     15 CD 59 16 19 D1 C9 FE    1B 3F D8 13 21 79 0E 07
15B0     85 6F 3E 00 8C 67 C9 CD    56 13 E3 CD 1E 15 BE 23
15C0     28 07 C5 4E 06 00 09 C1    1B 13 23 E3 C9 21 00 00
15D0     44 CD 1E 15 FE 30 D8 FE    3A D0 3E F0 A4 20 14 04
15E0     C5 44 4D 29 29 09 29 1A    13 E6 0F CD B0 15 C1 1A
15F0     F2 D4 15 D5 EF 1F 48 4F    57 3F 1F 00 C3 35 15 CD
1600     EA 14 C8 1A 02 13 03 18    F6 78 92 20 03 79 93 C8
1610     1B 2B 1A 77 18 F3 C1 E1    22 B5 0E 7C B5 28 10 E1
1620     22 B7 0E E1 22 B9 0E E1    22 BB 0E E1 22 BD 0E C5
1630     C9 21 5A 0F CD CF 14 C1    39 D2 56 15 2A B5 0E 7C
1640     B5 28 13 2A BD 0E E5 2A    BB 0E E5 2A B9 0E E5 2A
1650     B7 0E E5 2A B5 0E E5 C5    C9 2A 00 18 2B 2B C9 97
1660     47 1A 13 B8 C8 CD 4A 0C    FE 1F 20 F5 C9 CD BA 15
1670     22 0E 06 22 CD 61 16 FE    1F E1 CA DD 10 23 23 E9
1680     CD BA 15 27 04 06 27 18    EB CD BA 15 5E 0A 1A EE
1690     40 CD 4A 0C 1A 13 18 DF    C9 7B B8 C8 1A CD 4A 0C
16A0     13 18 F6 06 00 CD CC 14    F2 AE 16 06 2D 0D D5 11
16B0     0A 00 D5 0D C5 CD B7 14    78 B1 28 07 E3 2D E5 60
16C0     69 18 F2 C1 0D 79 B7 FA    CF 16 EF 20 00 18 F5 78
16D0     CD 4A 0C 5D 7B FE 0A D1    C8 C6 30 CD 4A 0C 18 F4
16E0     1A 6F 13 1A 67 13 0E 04    CD A3 16 EF 20 00 C9 4C
16F0     49 53 54 91 11 4E 45 57    90 CB 52 55 4E 90 D7 4C
1700     4F 41 44 91 C1 44 55 4D    50 91 C9 4E 45 58 54 92
1710     4D 46 4F 52 91 EB 49 46    92 B0 4C 45 54 93 46 47
1720     4F 54 4F 91 04 47 4F 53    55 42 91 84 52 45 54 55
1730     52 4E 91 A6 49 4E 50 55    54 92 CB 50 52 49 4E 54
1740     91 3D 52 45 4D 92 AB 53    54 4F 50 90 D1 50 4F 4B
1750     45 93 2F 4F 55 54 93 1D    43 4C 45 41 52 91 DE 93
1760     41 52 4E 44 94 5E 41 42    53 94 8A 53 49 5A 45 94
1770     93 50 45 45 4B 94 A0 43    4F 44 45 94 A8 49 4E 94
```

```
1780    AC 94 40 54 4F 91 FB 95    2B 53 54 45 50 92 07 92
1790    0C 3E 3D 93 60 3C 3E 93    66 3E 93 6C 3D 93 7B 3C
17A0    3D 93 73 3C 93 81 93 87    CD 4D 0C D0 FE 1E 37 C0
17B0    C3 29 10 CD 4D 0C D0 FE    20 C0 CD 4D 0C 30 FB FE
17C0    20 C8 C3 29 10 11 C0 0E    F5 AF 32 BF 0E F1 CD 4A
17D0    0C CD A8 17 30 FB FE 26    00 00 00 12 FE 1D 20 09
17E0    7B FE C0 28 EC 1A 1B 18    E5 FE 1F 28 09 7B FE 2D
17F0    28 DF 1A 13 18 D8 13 13    3E FF 12 1B EF 1F 00 C9.
```

For further information about this Tiny basic program or if you have any sugestions regarding it please contact;

Mr. J. M. Stevenson
33 Beacon Drive,
West Kirby,
Wirral.

****

Appendix 1.

Common faults encountered on the Nascom.

As time goes on it has become apparent that certain faults are occuring with regularity and it seems a sensible idea to list the more common problems that are being encountered.

Dealing with the main board first.

The most common trouble is centred around IC 18; 74LS123. If you find on switch on that certain characters are missing or strange characters are being displayed the chances are that IC 18 needs modification. Bend pin 5 underneath the IC away from the socket ie open circuit pin 5. Then take a shorting link and join pin 5 to pin 12 underneath the board.

If you have accidentally shorted the power rails or crossed them at some time the chances are that you get no response from the keyboard.

It seems that the weak link on the board is IC 41; 74LS378. It is worth keeping one of these chips spare.

The other main problem most people seem to encounter regards the tape input circuit. If Nascom transistors have been suplied such as NAS-01 I suggest that you change them. Any general purpose transistors will do. If you are in doubt as to what to use consult your local electronics shop which is bound to have something in stock which will suffice.

Faulty diodes have also been encountered in this part of the circuit.

Check them as per the Nascom manual. The input circuit requires quite a bit of 'Wumph' to operate so I advise using the external speaker socket of your tape recorder as opposed to the 'DIN' socket which generally gives a lower output.

Before you go delving into the Nascom, check your tape recorder preferably on another Nascom. There is bound to be somebody in your area who owns one.

Board faults have been encountered from time to time but not as often as people expect. generally the printed circiut board has been found to be faultless. Only rarely has it been found necessary to critisise the standard of soldering on home built kits, but its worth stressing for beginers to take extra care.

Memory faults occur quite often. There are so many symptoms such as double displays, random characters, unable to reset etc.

It is worth checking your memory chips on a working Nascom But remember if it is really warm don't risk it get a replacement. By the way it seems the nature of EPROMS to run warm so don't be concerned about this.

Always remember to follow the recomendations for handling CMOS devices and never remove anything when the power is on.

Eproms have been found to self destruct on a number of occasions so it is worth having a pre-programmed one for emergency use. The ceramic ones seem to be more prone to this. Funny things these eproms one day they will not work at all and the next they are perfect..sound a bit like me.

Faults on memory cards.

A large number of memory cards seem to suffer from memory 'plague'.

they work perfectly for a while and then crash.

A number of modifications are recommended to cure this fault.

1. Ensure IC's 1 and 2 on the card (81 LS97) are National semiconductor origin. They are marked NS or have a logo like two S's on their side.
2. Change IC 9 on the buffer board if it is a 74 LS 04. Replace it with a 74 S 04.
3. Grid the back of the memory card. ie. join the Ov to Ov and +5v to +5v horizontally accross the back.
4. Fit 4K7 resistors between pins 9 and 14 on the back of the dynamic memory chips. IC's.

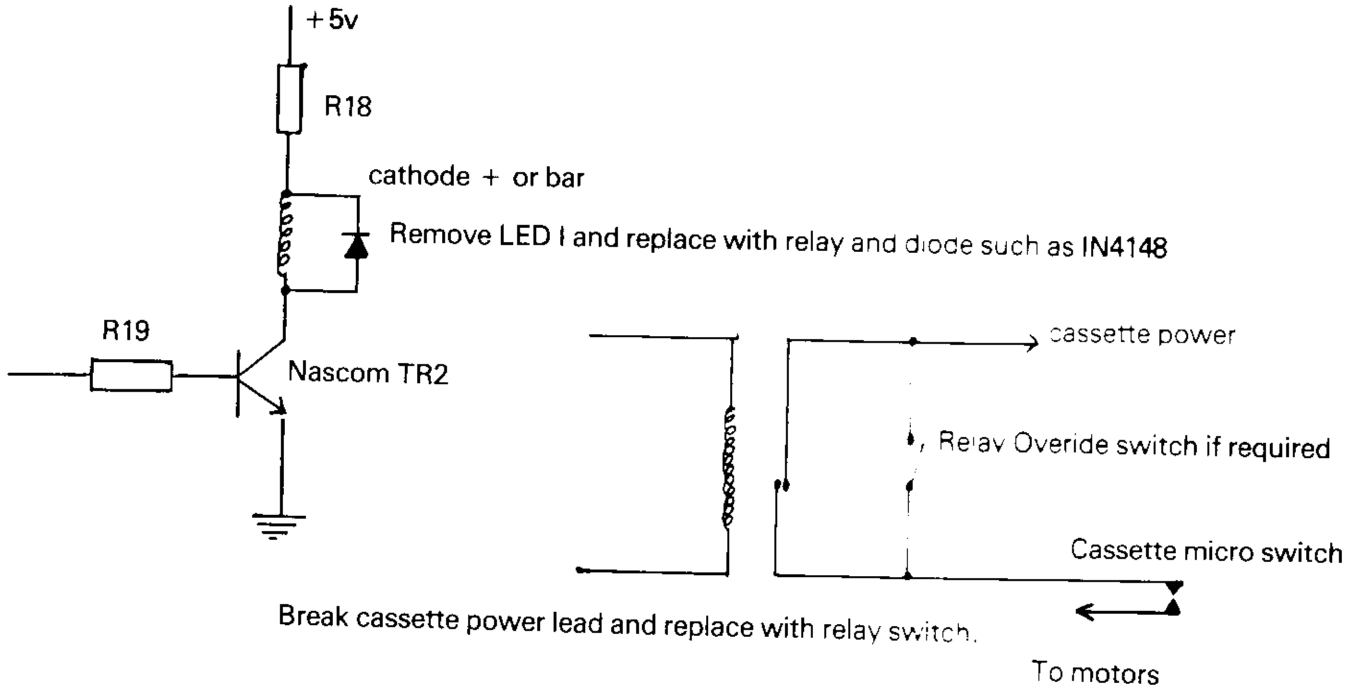
This last modification has usually been found to effect a cure.

Operating a tape recorder automatically.

This is simply done by replacing the load LED with a relay. Most relays available will do this job efficiently you don't need a big one that will suply power to London but only about 250 ma.

The relay should have a coil resistance of about 500 ohms and you should remember to solder a diode in parallel. with it to protect the driving transistor. Reversed biased of course.

A convenient place to attach the switch contacts is usually the microswitch found in most tape recorders. It is usually situated under the drive mechanism. If in doubt get your local TV. Audio engineer to do it for you.


+5v
R18
cathode + or bar
Remove LED I and replace with relay and diode such as IN4148
R19
Nascom TR2
cassette power
Relay Overide switch if required
Cassette micro switch
Break cassette power lead and replace with relay switch.
To motors

****

Appendix 2

Listing of Z80 op-codes in numerical order

The next 5 pages contain a list of Z80 op-codes in numerical order instead of the usual mnemonic order This list can be used to disassemble programmes, but care should be taken not to confuse data with the program

Please note that we have only printed one list of index register codes, as both the IX and IY registers have the same codes except for the first byte which is DD for the IX register and FD for the IY register.

øø	NOP
ø1	LD BC NN
ø2	LD (BC) A
ø3	Inc BC
ø4	Inc B
ø5	Dec B
ø6	LD B N
ø7	RLCA
ø8	Ex AF AF'
ø9	Add HL BC
øA	LD A (BC)
øB	Dec BC
øC	Inc C
øD	Dec C
øE	LD C N
øF	RRCA
1ø	DJNZ xx
11	LD DE NN
12	LD (DE) A
13	Inc DE
14	Inc D
15	Dec D
16	LD D N
17	RLA
18	JR xx
19	Add HL DE
1A	LD A (DE)
1B	Dec DE
1C	Inc E
1D	Dec E
1E	LD E N
1F	RRA
2ø	JR NZ xx
21	LD HL NN
22	LD (NN) HL
23	Inc HL
24	Inc H
25	Dec H
26	LD H N
27	DAA
28	JR Z xx
29	Add HL HL
2A	LD HL (NN)
2B	Dec HL
2C	Inc L
2D	Dec L
2E	LD L N
2F	CPL
3ø	JR NC xx
31	LD SP NN
32	LD (NN), A
33	Inc SP
34	Inc (HL)
35	Dec (HL)
36	LD (HL) N
37	SCF
38	JR C xx
39	ADD HL SP
3A	LD A (NN)
3B	Dec SP
3C	Inc A
3D	Dec A
3E	LD A N
3F	CCF
4ø	LB B B
41	LD B C
42	LD B D
43	LD B E
44	LD B H
45	LD B L
46	LD B (HL)
47	LD B A
48	LD C B
49	LD C C
4A	LD C D
4B	LD C E
4C	LD C H
4D	LD C L
4E	LD C (HL)
4F	LD C A
5ø	LD D B
51	LD D C
52	LD D D
53	LD D E
54	LD D H
55	LD D L
56	LD D (HL)
57	LD D A
58	LD E B
59	LD E C
5A	LD E D
5B	LD E E
5C	LD E H
5D	LD E L
5E	LD E (HL)
5F	LD E A
6ø	LD H B
61	LD H C
62	LD H D
63	LD H E
64	LD H H
65	LD H L
66	LD H (HL)
67	LD H A
68	LD L B
69	LD L C
6A	LD L D
6B	LD L E
6C	LD L H
6D	LD L L
6E	LD L (HL)
6F	LD L A
7ø	LD (HL) B
71	LD (HL) C
72	LD (HL) D
73	LD (HL) E
74	LD (HL) H
75	LD (HL) L
76	Halt
77	LD (HL) A
78	LD A B
79	LD A C
7A	LD A D
7B	LD A E
7C	LD A H
7D	LD A L
7E	LD A (HL)
7F	LD A A
8ø	Add A B
81	Add A C
82	Add A D
83	Add A E
84	Add A H
85	Add A L
86	Add A (HL)
87	Add A A
88	Adc A B
89	Adc A C
8A	Adc A D
8B	Adc A E
8C	Adc A H
8D	Adc A L
8E	Adc A (HL)
8F	Adc A A

9Ø	Sub B
91	Sub C
92	Sub D
93	Sub E
94	Sub H
95	Sub L
96	Sub (HL)
97	Sub A
98	Sbc A, B
99	Sbc A, C
9A	Sbc A, D
9B	Sbc A, E
9C	Sbc A, H
9D	Sbc A, L
9E	Sbc A, (HL)
9F	Sbc A, A
AØ	And B
A1	And C
A2	And D
A3	And E
A4	And H
A5	And L
A6	And (HL)
A7	And A
A8	Xor B
A9	Xor C
AA	Xor D
AB	Xor E
AC	Xor H
AD	Xor L
AE	Xor (HL)
AF	Xor A
BØ	Or B
B1	Or C
B2	Or D
B3	Or E
B4	Or H
B5	Or L
B6	Or (HL)
B7	Or A
B8	Cp B
B9	Cp C
BA	Cp D
BB	Cp E
BC	Cp H
BD	Cp L
BE	Cp (HL)
BF	Cp A
CØ	Ret NZ
C1	Pop BC
C2	Jp NZ, NN
C3	Jp NN
C4	Call NZ, NN
C5	Push BC
C6	Add A, N
C7	Rst Ø
C8	Ret Z
C9	Ret
CA	Jp Z, NN
CB	See separate list
CC	Call Z, NN
CD	Call NN
CE	Adc A, N
CF	Rst Ø8H
DØ	Ret NC
D1	Pop DE
D2	Jp NC , NN
D3	Out (N), A
D4	Call NC, NN
D5	Push DE
D6	Sub N
D7	Rst 1ØH
D8	Ret C
D9	Exx
DA	Jp C, N
DB	In A, (N)
DC	Call C, NN
DD	See IX register list
DE	Sbc A, N
DF	Rst 18H
EØ	Ret PO
E1	Pop HL
E2	Jp PO, NN
E3	Ex (SP), HL
E4	Call PO, NN
E5	Push HL
E6	And N
E7	Rst 2ØH
E8	Ret PE
E9	Jp (HL)
EA	Jp PE, NN
EB	Ex DE, HL
EC	Call PE, NN
ED	See separate list
EE	Xor N
EF	Rst 28H
FØ	Ret P
F1	Pop AF
F2	Jp P, NN
F3	DI
F4	Call P, NN
F5	Push AF
F6	Or N
F7	Rst 3ØH
F8	Ret M
F9	LD SP, HL
FA	Jp M, NN
FB	EI
FC	Call M, NN
FD	Index register IY see list for DD.
FE	Cp N
FF	Rst 38H

Note:
Index registers.
Apart from the first byte, both index registers have the same object codes.
Only one list has been compiled. This should be used for both registers.

DD = IX register
FD = IY register

ED 4ϕ	In B, C	ED 57	LD A, I	ED 7A	ADC HL, SP
ED 41	Out (C), B	ED 58	In E, (C)	ED 7B	LD SP, (NN)
ED 42	SBC HL, BC	ED 59	Out (C), E	ED Aϕ	LDI
ED 43	LD (NN), BC	ED 5A	ADC HL, DE	ED A1	CPI
ED 44	NEG	ED 5B	LD DE, (NN)	ED A2	INI
ED 45	RETN	ED 5E	IM, 2	ED A3	Out 1
ED 46	IM, ϕ	ED 5F	LD A, R	ED A8	LDD
ED 47	LD I, A	ED 6ϕ	In H, (C)	ED A9	CPD
ED 48	In C, (C)	ED 61	Out (C), H	ED AA	IND
ED 49	Out (C), C	ED 62	SBC HL, HL	ED AB	Out D
ED 4A	ADC HL, BC	ED 67	RRD	ED Bϕ	LDIR
ED 4B	LD BC, (NN)	ED 68	In L, (C)	ED B1	CPIR
ED 4D	RETI	ED 69	Out (C), L	ED B2	INIR
ED 4F	LD R, A	ED 6A	ADC HL, HL	ED B3	OTIR
ED 5ϕ	In D, (C)	ED 6F	RLD	ED B8	LDDR
ED 51	Out (C), D	ED 72	SBC HL, SP	ED B9	CPDR
ED 52	SBC HL, DE	ED 73	LD (NN), SP	ED BA	INDR
ED 53	LD (NN), DE	ED 78	In A, (C)	ED BB	OTDR
ED 56	IM, 1	ED 79	Out (C), A		

DD ϕ9	Add IX, BC	DD CB xx ϕE	Rrc (IX+Ind)
DD 19	Add IX, DE	DD CB xx 16	Rl (IX+Ind)
DD 21 xx xx	LD IX, NN	DD CB xx 1E	Rr (IX+Ind)
DD 22 xx xx	LD (NN), IX	DD CB xx 26	Sla (IX+Ind)
DD 23	Inc IX	DD CB xx 2E	Sra (IX+Ind)
DD 29	Add IX,IX	DD CB xx 3E	Srl (IX+Ind)
DD 2A xx xx	LD IX, (NN)	DD CB xx 46	Bit ϕ, (IX+Ind)
DD 2B xx	Dec IX	DD CB xx 4E	Bit 1, (IX+Ind)
DD 34 xx	Inc (IX+Ind)	DD CB xx 56	Bit 2, (IX+Ind)
DD 35 xx	Dec (IX+Ind)	DD CB xx 5E	Bit 3, (IX+Ind)
DD 36 xx xx	LD (IX+Ind), N	DD CB xx 66	Bit 4, (IX+Ind)
DD 39	Add IX, SP	DD CB xx 6E	Bit 5, (IX+Ind)
DD 46 xx	LD B, (IX+Ind)	DD CB xx 76	Bit 6, (IX+Ind)
DD 4E xx	LD C, (IX+Ind)	DD CB xx 7E	Bit 7, (IX+Ind)
DD 56 xx	LD D, (IX+Ind)	DD CB xx 86	Res ϕ, (IX+Ind)
DD 5E xx	LD E, (IX+Ind)	DD CB xx 8E	Res 1, (IX+Ind)
DD 66 xx	LD H, (IX+Ind)	DD CB xx 96	Res 2, (IX+Ind)
DD 6E xx	LD L, (IX+Ind)	DD CB xx 9E	Res 3, (IX+Ind)
DD 7ϕ xx	LD (IX+Ind), B	DD CB xx A6	Res 4, (IX+Ind)
DD 71 xx	LD (IX+Ind), C	DD CB xx AE	Res 5, (IX+Ind)
DD 72 xx	LD (IX+Ind), D	DD CB xx B6	Res 6, (IX+Ind)
DD 73 xx	LD (IX+Ind), E	DD CB xx BE	Res 7, (IX+Ind)
DD 74 xx	LD (IX+Ind), H	DD CB xx C6	Set ϕ, (IX+Ind)
DD 75 xx	LD (IX+Ind), L	DD CB xx CE	Set 1, (IX+Ind)
DD 77 xx	LD (IX+Ind), A	DD CB xx D6	Set 2, (IX+Ind)
DD 7E xx	LD A, (IX+Ind)	DD CB xx DE	Set 3, (IX+Ind)
DD 86 xx	Add A, (IX+Ind)	DD CB xx E6	Set 4, (IX+Ind)
DD 8E xx	Adc A, (IX+Ind)	DD CB xx EE	Set 5, (IX+Ind)
DD 96 xx	Sub (IX+Ind)	DD CB xx F6	Set 6, (IX+Ind)
DD 9E xx	Sbc A, (IX+Ind)	DD CB xx FE	Set 7, (IX+Ind)
DD A6 xx	And (IX+Ind)	DD E1	Pop IX
DD AE xx	Xor (IX+Ind)	DD E3	Ex (SP), IX
DD B6 xx	Or (IX+Ind)	DD E5	Push IX
DD BE xx	Cp (IX+Ind)	DD E9	Jp (IX)
DD CB xx ϕ6	Rlc (IX+Ind)	DD F9	LD SP, IX

The first byte of all codes on this page is **CB**

CB	00	RLC	B	CB	38	SRL	B	CB	68	Bit 5	B
	01	"	C		39	"	C		69	"	C
	02	"	D		3A	"	D		6A	"	D
	03	"	E		3B	"	E		6B	"	E
	04	"	H		3C	"	H		6C	"	H
	05	"	L		3D	"	L		6D	"	L
	06	"	(HL)		3E	"	(HL)		6E	"	(HL)
	07	"	A		3F	"	A		6F	"	A
	08	RRC	B		40	Bit 0	B		70	Bit 6	B
	09	"	C		41	"	C		71	"	C
	0A	"	D		42	"	D		72	"	D
	0B	"	E		43	"	E		73	"	E
	0C	"	H		44	"	H		74	"	H
	0D	"	L		45	"	L		75	"	L
	0E	"	(HL)		46	"	(HL)		76	"	(HL)
	0F	"	A		47	"	A		77	"	A
	10	RL	B		48	Bit 1	B		78	Bit 7	B
	11	"	C		49	"	C		79	"	C
	12	"	D		4A	"	D		7A	"	D
	13	"	E		4B	"	E		7B	"	E
	14	"	H		4C	"	H		7C	"	H
	15	"	L		4D	"	L		7D	"	L
	16	"	(HL)		4E	"	(HL)		7E	"	(HL)
	17	"	A		4F	"	A		7F	"	A
	18	RR	B		50	Bit 2	B		80	Res 0	B
	19	"	C		51	"	C		81	"	C
	1A	"	D		52	"	D		82	"	D
	1B	"	E		53	"	E		83	"	E
	1C	"	H		54	"	H		84	"	H
	1D	"	L		55	"	L		85	"	L
	1E	"	(HL)		56	"	(HL)		86	"	(HL)
	1F	"	A		57	"	A		87	"	A
	20	SLA	B		58	Bit 3	B		88	Res 1	B
	21	"	C		59	"	C		89	"	C
	22	"	D		5A	"	D		8A	"	D
	23	"	E		5B	"	E		8B	"	E
	24	"	H		5C	"	H		8C	"	H
	25	"	L		5D	"	L		8D	"	L
	26	"	(HL)		5E	"	(HL)		8E	"	(HL)
	27	"	A		5F	"	A		8F	"	A
	28	SRA	B		60	Bit 4	B		90	Res 2	B
	29	"	C		61	"	C		91	"	C
	2A	"	D		62	"	D		92	"	D
	2B	"	E		63	"	E		93	"	E
	2C	"	H		64	"	H		94	"	H
	2D	"	L		65	"	L		95	"	L
	2E	"	(HL)		66	"	(HL)		96	"	(HL)
	2F	"	A		67	"	A		97	"	A

The first byte of all codes on this page is **CB**

CB 98	Res 3	B
99	"	C
9A	"	D
9B	"	E
9C	"	H
9D	"	L
9E	"	(HL)
9F	"	A
A0	Res 4	B
A1	"	C
A2	"	D
A3	"	E
A4	"	H
A5	"	L
A6	"	(HL)
A7	"	A
A8	Res 5	B
A9	"	C
AA	"	D
AB	"	E
AC	"	H
AD	"	L
AE	"	(HL)
AF	"	A
B0	Res 6	B
B1	"	C
B2	"	D
B3	"	E
B4	"	H
B5	"	L
B6	"	(HL)
B7	"	A
B8	Res 7	B
B9	"	C
BA	"	D
BB	"	E
BC	"	H
BD	"	L
BE	"	(HL)
BF	"	A
C0	Set 0	B
C1	"	C
C2	"	D
C3	"	E
C4	"	H
C5	"	L
C6	"	(HL)
C7	"	A
CB C8	Set 1	B
C9	"	C
CA	"	D
CB	"	E
CC	"	H
CD	"	L
CE	"	(HL)
CF	"	A
D0	Set 2	B
D1	"	C
D2	"	D
D3	"	E
D4	"	H
D5	"	L
D6	"	(HL)
D7	"	A
D8	Set 3	B
D9	"	C
DA	"	D
DB	"	E
DC	"	H
DD	"	L
DE	"	(HL)
DF	"	A
E0	Set 4	B
E1	"	C
E2	"	D
E3	"	E
E4	"	H
E5	"	L
E6	"	(HL)
E7	"	A
E8	Set 5	B
E9	"	C
EA	"	D
EB	"	E
EC	"	H
ED	"	L
EE	"	(HL)
EF	"	A
F0	Set 6	B
F1	"	C
F2	"	D
F3	"	E
F4	"	H
F5	"	L
F6	"	(HL)
F7	"	A
CB F8	Set 7	B
F9	"	C
FA	"	D
FB	"	E
FC	"	H
FD	"	L
FE	"	(HL)
FF	"	A

NOTES

NOTES

Reader information service

In the event of any error being discovered, we would be grateful if you would please let us know in order that a list of corrections (if any) can be compiled.

It is our intention to send this list to all owners of this book who complete the form below and return it to us with a **stamped addressed envelope.** (Overseas readers please enclose an International Postal Reply coupon).

Your comments and criticisms will be welcomed.

Name..

Address..

....

Where did you purchase this book..

Would you be interested in a 2nd book?..

Do you have a program for book 2?...

Any comments please..............

Merseyside Nascom Users Group. C/o. T. Lyon and Co. Ltd., Samuel House, Taylor Street, Liverpool L5 5AD. England.

the new TEXTIE MARK FIVE
nascom 1 word processor

:powerful features include:

* WORD WRAP-ROUND ON UP TO 120 COLUMN LINE
* AUTO LATERAL OFFSET MOVES PAGE SIDEWAYS
* FULL PRINTER CONTROL margin-spacing-length
* POWERFUL EDITING, LATCHABLE SHIFT ETC.,ETC.
* RELOCATABLE ERROR CHECKED TAPE READ & WRITE

ON CASSETTE WITH DOCUMENTATION P/P ONLY 13.50
REQUIRES T4 AND A MINIMUM OF 8K RAM EXPANSION

---

for adults' amusement only

soothsayer
****

BASED ON ANCIENT WRITINGS "soothsayer" MAKES AN IDEAL PARTY GAME ANY TIME OF THE YEAR ANSWERING QUESTIONS ABOUT MARRIAGE, PERSONAL WEALTH, ETC. USES LARGE AMOUNT OF DATA IN FORMING ITS REPLIES BUT WILL RUN ON AN UNEXPANDED NASCOM 1. ASK YOUR FRIENDS TO JOIN YOU IN SOME OF THE MOST ENJOYABLE COMPUTER ENTERTAINMENT AVAILABLE. SELLS AT 6.50.

ORDER TODAY DIRECTLY FROM:

THE SOFTWARE PUBLISHING COMPANY
8A CHURCH SIDE, MANSFIELD, NOTTS
Telephone: (0623) 29237

TELEPHONE ORDERS BY ACCESS