MICRO'S GAZETTE

MSX AND SPECTRAVIDEO MAGAZINE

Registered by Australia Post
Publication No. TBH 1099

ISSUE: 011

NOVEMBER - DECEMBER 1990

MSX

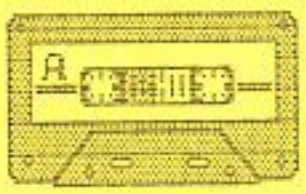
ASCII SOFTWARE

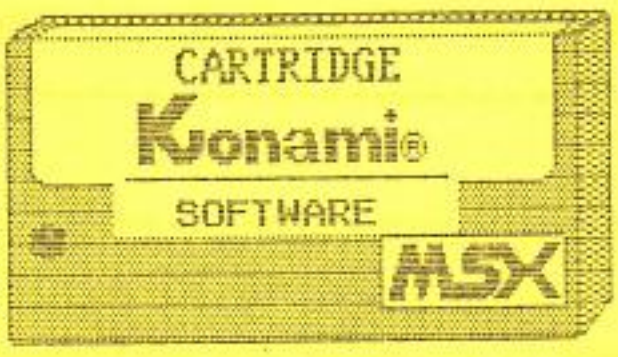
COMPILE

ROM / DISK 2DD

KONAMI!

ROM / DISK 2DD





T&ESOFT INC.

ROM / DISK 2DD

C.H.SOFTWARE

ROM / DISK 2DD

T&ESOFT INC.

MSX

CONTENTS

MICRO'S GAZETTE is produced and published with the help of the following people:

EDITOR - Mark Sands
ASSISTANT EDITOR & BASIC COLUMN - Craig Hursey
TECHNICAL ADVISER - Robin Young
RESIDENT HACKER - Jason Balogh
MUSIC COLUMN - Peter Nielsen
MSX NEWS - Ross Brisbane
GAMES INSTRUCTIONS - The Garrett Family

PRINTED BY MICRO'S GAZETTE

Send all Correspondence to -:

MICRO'S GAZETTE
P.O. BOX 663,
ST. LEONARDS,
TAS. 7250
PHONE (003) 392 775

EDITORIAL

The holiday season has been and, presumably for most of us, gone. I had a lovely week camping on the banks of a small river with my family and we enjoyed ourselves immensely. The children spent all week in the rather fresh water, while I was content to sit on the bank in the shade of the trees with a book and stubbie and let my mental and physical batteries recharge after the hectic weeks (both at work as well as at home) leading up to christmas. I hope you all had a good break as well.

The response to the shareware offer was well received by many, so, I have decided to continue with it into this new year. We have aquired a contact with 1200 MSX 1 programs, we have already purchased some programs, and we will buy more during the year.

ROSS has started a series on dBASE II this issue, and I am sure that many of you will find this usefull, the book I read over the christmas period was " AN INTRODUCTION TO dBASE II ", and it was an eye opener to the power of this program.

PETER has responded to another of my never ending requests, and has wandered off into the fields of sequential files this issue, hopefully he will follow it up next issue with an article on how to pull the info out of these files and send them to your trusty printers in the format that you want. How about it PETE?

CRAIG has been busy this time, not only doing his basic column, but also a piece on his new MSX TURBO R COMPUTER, we believe this is the first to be brought into AUSTRALIA.

We have a users column this issue compliments of GORDON BROWELL. GORDON has been an MSX user for some years and has now branched out into VIDEOGRAPHY. He will tell you how you can combine the two hobbies to your benefit.

This would seem to be the ideal place to remind you of the fact that any article sent in and published, whether it be a user column, listing, software reviews or even games instructions, is rewarded with a free issue added to your subscription. We do need help and will run out of these sort of things soon.

We have the following software to add to our shareware list this issue, RAMBO III, BLASTEROIDS, HUNT FOR RED OCTOBER, OBLITERATOR, DAWN PATROL, FLIGHT DECK II, THE WAY OF THE TIGER, ZAXXON (Sony HB75as only). At the moment these are disk only. Sorry. We have others but they have not been sorted yet, but our lists continue to grow slowly each issue.

Well thats all from me for another issue, on closing I would like to thank all of the members who take the time to let us know how appreciative they are of our efforts to keep this magazine going, I also am appreciative of the help given to me in producing this magazine by those people mentioned on the contents page, we all do this in the interest of our hobby, MSX computing.

KEEP MSXing!!!!

MARK SANDS

MSX NEWS
BY ROSS BRISBANE

MSX DATABASE MANAGEMENT PART 1 of 4 Parts (over four editions).
What we will be doing together will really surprise you. The possibilities of this software in everyday applications are almost endless.

Part 1 : Creating the Database Example File, Adding and Basic Interrogation of Database
Part 2 : Advanced Interrogation Techniques and Introduction to Dbases Programming Language
Part 3 : Creating a Program, Screen Layouts, Add, Edit, View
Part 4 : Other uses for Dbase, Advanced Programming Techniques and using the Report Writer in Dbase

You will the need the following minimum hardware & software.
Check List:
Sorry you cannot use Dbase II at all on only MSX cassette based at all.

1. A MSX I or II Computer with 64K memory
2. One 3.5" Disk Drive (built in or connected via Slot)
3. MSX-DOS Operating System on boot disk
4. Dbase II on same disk as 3.
5. A Printer for Part 4 Article

Micro's Gazette will gladly supply 3 & 4 of check list.
If you do not own a printer I can get them brand new with 12 months warranty for $279.00. They are EPSON compatible. A MSX printer cable will have to be made up. I think Micro's Gazette should be able to help with cables.

Let's start with a basic instruction on MSX-DOS operating environment.

MSX-DOS allows greater freedom to create different applications. The two advantages are:

1. More memory available for programs. ROM Basic is "paged out" temporarily leaving more memory in a "free" state. I understand that this memory is as high as 55K totally available. Basic is still there and is invoked by typing BASIC at the A> prompt. In BASIC mode type the following : PRINT FRE(0). Now consider how much memory you have!. You can knock out the ROM Basic again temporarily by typing CALL SYSTEM in BASIC ROM mode!. You are then dropped out to the freer MSX-DOS environment.

2. We will not be using BASIC ROM at all! There is too much dependence on this MODE and leads to false ideas on Computer operation particularly for the MSX System.

3. The MSX-DOS environment was designed to match MS-DOS syntax. (It will not however run IBM software). This MSX-DOS environment was designed to be BIOS compatible with C/PM in 95% of all cases. C/PM was the forerunner operating system to MS-DOS.

MS-DOS stands for Microsoft Disk Operating System

C/PM stands for Control Processing Micros

MSX-DOS stand for Microsoft e-X-tended Disk Operating System
The X in e-X-tended means that Machines that run this operating system share common hardware and software compatibility by signed agreement.

4. By freeing the environment on your MSX computer we are no longer BASIC dependent we can create many different types of files on disks. This "freedom" will baffle you at first but you will be very surprised at what we'll create together in Part 1.

Let's begin.

What is a Database ? Well a telephone book, encyclopedias, a Catalogue, library listing of books. What about a newspaper. Yes or No. After all a newspaper contains information.

Answer: A database is any information that is presented in such a manner that makes the finding and locating of that information E-A-S-Y ! !.

dBASE II is a software TOOL that can create an electronic lot of information and store it on a disk for incredible E-A-S-Y retrieval or look up.

Example : I have 30 videos all recorded with information. The videos are NUMBERED but what is on them ?. I can put this information in a book or I can put this on DISK !. Now suppose I had 500 videos. Dbase can handle this with no problem. I simply tell the computer to find ! a title I may be looking for !. dBASEwill do this in seconds for me.

A database has RECORDS.

The telephone book has Surname, Initials, Street, Suburb/Town. What is the KEY to finding information in the telephone. Surname of course which is in Alphabetical Order. Suppose I wanted to do a search in dBASE for a person whose surname was Smith. I would type in the following: FIND Smith. The computer will then Find those names that have the surname Smith !.

So a Database is an orderly listing of information.
A Database has Records of which one is the KEY.

This KEY is referred to as a FIELD. Many FIELDS make up a RECORD.
Many RECORDS make up a DATABASE.

Our first Exercise :

* How to plan them
* How to make them
* How to use them
* How to change them

Note: BOLD is what you type in !

1. Boot your computer with MSX-DOS & Dbase II disk in Drive A

2. Don't worry about date. Press enter to pass.

3. At A>DBASE (remember TYPE in ONLY the letters in bold)

4. Press Enter Button on keyboard

5. Dbase loads in a few moments

6. A date PROMPT will come up. Just press enter.

7. A 'Dot Prompt' will appear like this as follows:
.

8. At .QUIT (remember type in ONLY the letters in bold)

9. Press Enter and you will be returned to the MSX-DOS Prompt A>.

10. The above should all work well. If it does move on to create your first electronic database. If you have trouble start at Step 1 above to Step 10 until you are confident.

Creating our Database Example : Software Database of MSX Software.

1. A>DBASE and press Enter

2. Press Enter to bypass Date prompt in Dbase

3. At the Dot prompt like : (REMEMBER ONLY TYPE WORD IN BOLD)

4. Dbase asks you for file name like :

ENTER FILE NAME: MSXSWARE

5. Dbase asks you for :

ENTER RECORD STRUCTURE AS FOLLOWS:

FIELD	NAME,TYPE,WIDTH,DECIMAL PLACES
001	SOFTWARE,C,15 Press Enter

6. Dbase puts a 002 on the screen as follows:

002	DESCRIPT,C,7 Press Enter

7. Do the same for the rest.

003	MSXVER,C,3	Press Enter
004	MEMORY,C,4	Press Enter
005	COMMENT,C,12	Press Enter
006	DISKNO,N,3	Press Enter
007	Press Enter	

8. Dbase asks the following to which we answer Yes :

INPUT DATA NOW? Y

9. The following screen appears something like this :

RECORD # 00001

```
SOFTWARE      :                :
DESCRIPT      :       :
MSXVER        :   :
MEMORY        :    :
COMMENT       :            :
DISKNO        :   :
```

Dbase is waiting for you to key in your information !. The first piece of software you have below is an example filled in !.

```
RECORD # 00001

SOFTWARE      :TETRIS          :
DESCRIPT      :GAME    :
MSXVER        :I   :
MEMORY        :64k :
COMMENT       :SKILL       :
DISKNO        :  1:
```

You press enter to fill in another.

```
RECORD # 00002

SOFTWARE      :BOXING          :
DESCRIPT      :GAME    :
MSXVER        :I   :
MEMORY        :64k :
COMMENT       :GOOD GRAPHI:
DISKNO        :  1:
```

Notice above that Dbase numbers this as RECORD 2.

10. You can continue to fill in More Records, noting that dbase increments by one as you add more records. If you leave the Record you on blank, DBASE exits AUTOMATICALLY.

For our example I want you to enter at least 20 known MSX software titles. Use the shareware list if you want to or else if you have a lot of time on your hands enter all your MSX software collection you have matching what you enter to the disk NO. (the disk number field) that the particular software is on. If you don't have a number system on your disks we can do this later.

In Database Management don't worry about the order or type of software you enter. Just enter it into the database. This can be sorted later, printed in any order, cataloguing type of software into unique list on screen or whatever. Remember to end your final entry by leaving last record blank. Once we get software into database we can share lists around via DBASE !.

We now have a Database with so many records.

11. To look at files on disk we type :

dBASE THEN LISTS ALL DATABASE CREATED SO FAR. Notice the extension of DBF. This stands an a unique identifier for this D ata B ase F ile.

12. To look at structure of your database you should do two steps. Step 1 insures dBASE is looking at the correct database. Step 2 displays the structure (fields or slots) of you database. As follows :

dBASE displays the structure of your database to the screen. Learn to read whats on the screen. Dbase gives you a lot of information about your database file.

12. To set the list of what you have keyed we type in :

Have you been able to pick the difference between DATABASE STRUCTURE and DATABASE CONTENTS.

13. Try the next command. This one suppresses the record Number.

14. Suppose I wish to search for a particular game in the following Example I would type :

This powerful command searches the field SOFTWARE for the word TETRIS in your database. If it exists, it then finds the game and displays the game to the screen !.

The computer goes to the beginning of your database (record 1) looks through SOFTWARE and sights any character string equivalent of TETRIS. If nothing is found Dbase returns you to the Dot Prompt.

15. To get back to the database and 'open' it for more additions we simply type :

RECAP :

1. Disk in drive with MSX-DOS and DBASE on it.
2. Turn machine on. Don't worry about date. Press ENTER
3. At A> prompt type : DBASE
4. Bypass date in Dbase by pressing ENTER
5. To call up database type at DOT prompt : USE MSXSWARE
6. To add more records type : APPEND
7. To stop adding records just leave last entry blank press ENTER
8. To list to screen type : DISPLAY ALL
9. To list a particular piece of software : DISPLAY FOR "BOXING"$SOFTWARE
10. Remember to always EXIT dBASE by typing : QUIT

Note: MSX-DOS'S DIR command shows you have created your database on disk as well. Once you have EXITED dBASE you safely switch off your computer !.

In closing you will notice the English like commands for using DBASE. They do not change. They are a fixed narrow range of commands that you learn over a period of time in Dbase.

If you are confident you may like to try creating you own database. Any letter name of up to 8 letters will do. Do not worry about the DBF extension. Dbase adds this automatically. How do we create our own.

Remember : .CREATE Press ENTER.

Bye till next time: Ross J Brisbane
11 Thirlemere Road,
Alexandra Hills, QLD. 4161
Phone : (07) 824 5242

USERS COLUMN

MSX ON TELEVISION

Gidday. Remember me? I'm the guy who did his block trying to unravel random disk files and ended up writing a long screed on the subject which was published in issue V.3 N.4. I've always had it in mind to give you my support by contributing items of interest, but somehow domestic chores always seem to gobble up computer time.

Anyhow, here is something that may be of interest to other MSX'ers. The main idea of this story is really an attempt to link PC's to Videography. What's that again?

Videography is the latest craze of using Video Camera (Camcorders). The story tells it all. Hope you can use it.

GORDON BROWELL

Remember the early days of discovery when your first personal computer was brand new? And what about that first practical program you wrote and ran successfully? For me, those were the days when practice and theory came together to produce frustrating bugs for which the computer refused to take the blame. I was reminded of those days when I turned my back on my SONY HB75as and bought myself a video camera. Siggy Freud would have known why I avoided SONY and instead chose SHARP.

Shooting and editing movies has many parallels with computing. Every error is certainly yours. Getting it right keeps driving one back to the shop to buy extra peripherals. The saving grace is that, like computers, the tape can be used over and over again. But the real fun doesn't begin until it is necessary to edit all your good shots into some sort of logical sequence. And again, like computer programs, your time-consuming efforts are immediately available for assessment on screen whenever you enter RUN.

My first effort to emulate the professionals was anything but professional. On the floor in front of the family TV and VCR was a TANDY Multi-channel Mixer, two mono cassette players, a microphone and the camera, all tied together with yards and yards of trip wires, to say nothing of scripts, editing notes, music cassettes and VHS-C cassettes. Even my wife acknowledged the similarity with my early days of struggle with Dick Smith computer gadgets. She even repeated her threat (or promise) to run off with the milkman.

Then came the break-through. Imagination, lateral thinking and sheer genius came to the rescue. Back to the shop. The assistant had not retired on the profits of my purchases. This time I bought a Mini-Transmitter (a Video Sender), after making the assistant promise to take it back if it wasn't suitable. I need not have worried though. It worked superbly. You've got to strike lucky sometimes, eh? Anyhow, this little box of tricks actually transmits Audio and Video signals on UHF (CH-31). The quality from across my living room is excellent. The picture is not even disturbed by people walking about. The heap of garbage has gone from the carpet and all my editing gear is now neatly arranged on a small table in a discreet corner of the room. Editing Video movies is now a doddle. The milkman stands no chance against a genius for a husband.

While all this is going on, my SONY HB sat sullenly and quietly sulking. Then came another flash of sheer genius. I remembered the 6-pin DIN output socket on the back of the computer. Only three pins were needed to gain seperate Audio and Video outputs. Back to the shop. This time for a 6-pin DIN plug and an ARISTA video switch. Then in just one afternoon I knocked up six BASIC programs

using lots of DATA and VPOKE fancy alpha-numeric characters and decorative graphics for putting video movie titles on the screen and recorded by the VCR. For the first time I had found a good practical use for the MSX 3-channel sound channels. The titles are accompanied by MSX music. Haaah! You should see my movies now!

All that remains now is to let you know what equipment I use for editing:

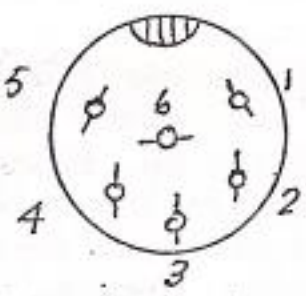
A SHARP C670 VHS-C Video Camera.
A TANDY Realistic 4-Channel Stereo Microphone Mixer (8 channels MONO)
An ARISTA VS300 Video Selector to switch between camera and computer.
A VIDEO SENDER UHF Ch-31 - UT-66 (both Audio and Video seperately tunable).
A couple of PANASONIC Cassette Recorders with press-button REMOTE switches. (MONO)

Incidentally, the screen display of my computer monitor from the Video Sender is better than the RF output. The pins of the 6-PIN DIN socket are given below.

HOOROO!!!!!

From HB-75AS
POWER BOARD diagram.

Pins 2, 3 & 4 only
used for Video Sender.
Twin shielded cable
(common RETURN).



PINS 1 & 5 =
+12V OUT

PINS 4 & 6 =
AUDIO OUT

PIN 2 = VIDEO

PIN 3 = GROUND

FOR SALE

1 Coleco Adaptor	$ 20.00
1 SVI 318/328 Expander	$ 10.00
13 or so Commercial SVI 318/328 Games Tape & cartridge	$ 25.00
1 Printer Card for 318/328 Computers	$ 20.00
1 Coleco Joystick in pieces	$ 3.00
1 Complete 328 Spectravideo Computer 64K Mark II (Built Modulator) with Power Pack PERFECT CONDITION	$ 50.00
Case for 5.25 Drive	$ 10.00
3 Lengths of 36 Way Cable for Drives/Printer with connectors (36 pin connector/centronics)	$ 12.00
1 Floppy Contoller for 318/328 Computers	$ 20.00
1 5.25 Bare Disk Drive Double-Sided	$ 25.00
1 80 Column Card	$ 45.00
One MSX Express 738 Computer in faultless condition:	$ 299.00
One only IBM XT motherboard (80286 processor) under warranty 1024K user RAM	$ 280.00

All prices are Negotiable except the MSX Express 738

Post and pack for small items	add $ 5.00
Post and pack larger Items	add $10.00
Post and pack COMPUTERS	add $15.00

Ross J Brisbane
11 Thirlemere Road,
Alexandra Hills Q 4121
Ph. (07) 824 5242

MSX MUSIC
BY PETER NIELSEN

SEQUENTIAL FILES

MSX MUSIC appears to have taken a back seat again. Mark has asked me to explain a bit about sequential files as there appears to be quite a few users who don't understand what they are about, and consequently my last months article on random files was just so much gibberish. I will now endeavour to rectify that situation, so all you experts who have been using sequential files for yonks, can skip this if you think you "know it all". Then again, one can always learn a new angle on something, as was demonstrated to me some weeks ago by a new owner who asked me "why can't you just do ----- ". I had never thought about it that way and discovered that you "could" just do --- !!!! We live and learn, so here goes.

A sequential file is a number of records (can be of varying length up to 255 characters) that has been written to a medium (disc, tape, catridge, etc) and can only be retrieved by reading from record 1 ,record 2, etc, till the record required (or the end) is reached. If you think about data written to a cassette tape, it becomes obvious that the data is sequential, one block of data followed by another.

To access a file it must first be OPENed for the type of operation that you wish to perform. e.g.

```
OPEN" file name" FOR    INPUT   ' allows you to read information
or  OPEN" file name" FOR    OUTPUT  ' allows you to write information
```

Using the command OPEN, gets the computer to allocate a small section of memory (typically 256 bytes), for use as a temporary storage area. This area is called a File Control Block (or buffer), and the computer either reads or writes to the medium in blocks of information. When writing, it puts together the sections of data into the buffer and when this is full, it will then write the block to the disc or tape. If you have already used files you will have noticed that the disc or tape operates in bursts. (Saving a program in ascii gives the same effect). As there may be the need to read information from one medium and write it to another medium, (copying a tape to a disc or vice versa), your MSX computer allows you to have seven file buffers allocated at any one time. Therefore you must tell your computer when you OPEN a file, which File Control Block to use, and so your OPEN command should look like :-

```
OPEN" file name" FOR     INPUT     AS  #1
                      or OUTPUT
```

The "#" sign can be translated as "number", and if you wished to have a second file operational at the same time, you could OPEN it as #2.

I know that some books say that you can have up to fourteen files open at one time, but most computers will only allow up to #6. File number zero is permanently allocated for LOAD, SAVE, MERGE, and RUN.

For the cassette user:
In the OPEN statement you need to specify a device name followed by a colon, inside the quotes around the filename. e.g.

```
OPEN "CAS: filename " FOR INPUT AS #1
```
and your file name can only be six letters long.

For disc users who are lucky enough to have more than one disc drive, your device name is the drive character. e.g. "A:filename" or "B:filename". If you don't specify which drive, the default drive of A: will be automatically selected. If you want the B: drive to be the default drive, then go to DOS, type B in response to the A> prompt, and then go back to Basic. Other devices are supported under MSX BASIC such as CRT: GRP: and LPT: but I don't intend to go into them at the moment. Please consult your BASIC Reference Manual for further explanation.

Disc users have another mode of operation available to them for use with sequential files. The file can be OPENed for APPEND AS #1. This mode allows you to write more information to the file and the computer will put it on the end of the file. BEWARE !! Some disc controllers will APPEND to a file without realizing that there is another program or file following. This results in the APPENDed information being written over the start of the following program. Mine does, so I always make sure that a file that I am going to APPEND to, is the last one on the disc.

At this point I would like to explain (and emphasize the importance of) the CLOSE command. Once you have finished with the file, "CLOSE" it !!!! This is particularly important on OUTPUT, as this command will empty any remaing information in the buffer and write an End Of File marker (EOF) to the file. When you are reading information from a file, the EOF marker will let you know that the end of the file has been reached, and you can avoid your program stopping because of an Input Past End error. CLOSE-ing a file can also avoid a few other "Disasters" (See last issue).

To make a file of data that we can use later in the article, we will have to first :-

```
OPEN "TEST" FOR OUTPUT AS #1
```

We can now send information to the buffer for eventual writing to the file. This is achieved by using the command "PRINT #1," and following it with the data we wish to store. (Dont forget the comma after the file number !)

```
PRINT #1, 12345
```

If you have more than one item of data to store, you simply seperate the numbers with some type of "end of record" mark. These can take various forms and can be interpreted differently, depending on how the records are read in. This may sound a bit confusing, but an explanation of the normal PRINT command (prints to the screen), may help to some extent.

1. The command PRINT 12345 : PRINT 678 will result in each number being printed on a seperate line because the PRINT command ends with an automatic carriage return and line feed. (CR ascii 13 and LF ascii 10) . Thus we get:-

```
12345
678
```

2. The command PRINT 12345;678 results in:-

```
12345  678
```

The semicolon cancels the CR and LF, and both numbers are printed on the same line. Please note that in BASIC all numbers printed with a leading and trailing, blank space (hence two spaces between the numbers)

3.

The command PRINT 12345" "678 where the space is enclosed in quotes will result in :-
12345 678
the same as when the semi-colon was used.

4.

The command PRINT 12345,678 will result in :-
12345 678
because the comma is interpreted as "move to the next column that is 14 or a multiple of 14."

5.

The command PRINT "12345";"678" will result in:-
12345678
because the numbers were in quotes and the computer then treated them as strings. (no leading or trailing blanks).

6.

The command PRINT 12345 678 will result in :-
12345678
because the blank space is ignored the same way as when a blank space is used between command words in MSX Basic Programming.

After all that, we now get to the point.

THE COMPUTER PRINTS TO A FILE IN EXACTLY THE SAME MANNER !!!!!

You can use a comma, a space in quotes, a comma in quotes, a semicolon, or put the information inside quotes and they will all work as end of record markers, but each one can be interpreted differently by INPUT #1, when you read the file, depending on whether you are storing the information in a string or a numeric variable.

The only really safe way is to use one PRINT command for each item, as was done in example 1. This may appear to be a waste of programing space, (and the time taken to type them in) but it can be achieved by the use of an array and a FOR/NEXT loop. For those who don't know, a single dimensional array can be made by using DIM name (quantity). DIM NA$(99) will create 100 string variables of the name NA$. - NA$(0)to NA$(99)- and DIM PH(99) will create 100 seperate numeric variables of the name PH. You could fill these with names and phone numbers and then file them using :-

```
500 OPEN "NAMEPHON" FOR OUTPUT AS #1
510 FOR X = 0 TO 99
520 PRINT #1, NA$ (X)
530 PRINT #1, PH  (X)
540 NEXT X
550 CLOSE
```

This way of printing 100 (or more) records to a file is quite compact, but more importantly it is "safe", as each item is followed by a CR-LF pair which will not be misinterpreted when reading the file back into the computer.

To read information from a file back into the computer we must first :-

```
OPEN "NAMEPHON" FOR INPUT AS #1
```

And then INPUT #1, variable name

With INPUT #1, the variable name can be a list of names seperated by a comma, and will not cause problems if the file was PRINTED properly.

To read back in the name and phone list that we filed previously we can use the same FOR/NEXT loop and array idea that we used to PRINT them. The only difference is in line 620 where we test for the EOF marker on the file.

```
600 OPEN "NAMEPHON" FOR INPUT AS #1
610 FOR X = 0 TO 99
620 IF EOF (1) GOTO 650
630 INPUT #1, NA$ (X) , PH (X)
640 NEXT X
650 CLOSE
```

If there will always be a known number of records in the file it is not necessary to look for the End Of File mark. In cases where the number of records in the file is unknown, then line 620 is essential. An attempt to INPUT from a file when there are no more records, will result in an Input Past End error and the program will halt. This can be overcome by the use of an error trap routine, but that is another article to write someday.

When data is read into Numeric variables INPUT#, it would appear that most End Of Record marks are recognized and no trouble. If however the data was PRINT# ed as in example 5, the semicolons used in the PRINT# statement do not give any mark, and so on reading the record into a string variable results in all the records up to an CR-LF,(or up to 255) - WILL BE READ INTO THE FIRST STRING VARIABLE !!

Numeric variables can only store numbers, (that is the characters 1234567890), and string variables can store both alpha characters and the numeric characters. Using the INPUT# command when getting data from files partly reverses this, and can have some interesting consequence. Even if the data was previously stored as a number, and written to the file as a number, it can still be INPUT # from the file into a string and is accepted as such. If you INPUT# string data into a numeric variable, the record is actually read from the buffer and an implicit VAL statement is carried out, thus 0 is stored in the variable and the program continues merrily on its way.

I hope that the preceeding explanation has been of some help to someone out there in MSX land, and I hope in the next issue to explain how to get the valuable information you have so carefully filed, translated into a readable paper-printed form.

Till the next issue - Happy Programming,

Peter.

Anyone wishing to write to Peter direct with queries regarding Music, etc. may do so by addressing their letters to P.O. BOX 232, MOONAH, TAS., 7009

```
2670 LPRINT A$(0,Y)TAB(25)": "A$(X,Y)
2680 NEXT Y
2690 LPRINT STRING$(80,"-")
2700 :
2710 LPRINT:NEXT X
2720 RETURN
2730 DEFSTR D,U,L,R
2740 DO=CHR$(10)
2750 UP=CHR$(30)
2760 RI=CHR$(28)
2770 LE=CHR$(29)
2780 RESTORE 2800
2790 FOR T = 3032TO3055:READA:VPOKET,A:
NEXTT:RETURN
2800 DATA 0,32,52,40,32,32,0,0,0,156,16
0,152,132,248,0,0,0,136,80,32,80,136,0,
0
3000 IF ERR=19 AND (ERL=150 OR ERL=160
OR ERL=170) THEN PRINT:PRINT"READ ERROR
  ":INPUT "Do you want to try again (Y/N
) ";A$:IF A$="Y" OR A$="y" THEN RESUME
150 ELSE RESUME 46
3010 IF ERR=19 AND (ERL=1660 OR ERL=167
0 OR ERL=1680) THEN PRINT:PRINT"WRITE E
RROR, please save again":PRINT"Rewind t
he tape and press any key":CLOSE #1:A$=
INPUT$(1):RESUME1660
3020 IF ERR=19 AND (ERL=1770 OR ERL=178
0 OR ERL=1790) THEN PRINT:PRINT"WRITE E
RROR, please save again":PRINT"Rewind t
he tape and press any key":CLOSE #1:A$=
INPUT$(1):RESUME1660
3030 IF ERR=9 THEN PRINT:PRINT"The numb
er you typed is too high":RESUME NEXT
3040 IF ERR=19 AND (ERL=2650 OR ERL=267
0 OR ERL=2690) THEN PRINT:PRINT"Still p
rinting...":PRINT:RESUME NEXT
3050 IF ERR=15 THEN PRINT"You can only
use 255 bytes !":PRINT:RESUME NEXT
3060 IF ERR=14 THEN PRINT:PRINT"THE MEM
ORY IS FULL !":PRINT"Please save this f
ile and start":PRINT"with a new file.":
PRINT:FOR T = 1 TO 2000:NEXTT:RESUME 16
40
3070 IF ERR=7 THEN PRINT:PRINT"Your com
puter hasn't enough MEMORY":END
3080 IF ERR=2 THEN PRINT:PRINT"Syntax e
rror in line"ERL
3090 ON ERROR GOTO 0
3100 END
4000 CODE$="          ":TIME=0
4010 BEEP:CODE$=CODE$+INKEY$:IF RIGHT$(
CODE$,6)="3x3=11" OR RIGHT$(CODE$,6)="3
X3=11" THEN ON ERROR GOTO 0:END
4020 IF TIME<1000 THEN 4010
4030 RETURN
9999 END
```

GAMES INSTRUCTIONS

VERA_CRUZ----------You have just been made DET_SERG DEdDet-Sag in the crime squad at Saint Etienne (Loire).
As an Officer of the Police and in your capacity as a Judicial Police Officer, you are capable of leading any criminal investigation.

No sooner have you taken up your new post when you are called upon to investigate what the Press are calling "The Vera-Cruz Affair".

On the 8th August, 1986 ,you are notified by the caretaker of the "Forest" apartment's of a discovery of a body. It has been provisionally identified as one of the resident's and , from the caretaker's description, it is believed to be a case of suicide with a shotgun as the weapon involved.

You and your team must go to the scene of the crime to ascertain the facts. Once this is complete you will have to carry out the actual investigation,using the methods at your disposal.

PRICIPAL OF THE GAME

Has the resident of the FOREST really committed suicide as the caretaker believes?
You must start your investigation by carring out routine inquires at the scene of crime. HOWEVER, a word of advice, AN INVESTIGATION CANNOT END WITHOUT EFFICIENT AND EXHAUSTIVE INQUIRES. These serve as a basis for research and often evidence. Remember this...

1.THE FOREST RESIDENCE:

Study the scene to find clues, When you find a clue you must take photo's of the clues. To move the camera you use the cursor keys. To take the photo, hit SPACE-BAR.
Make sure you you have made note of ALL the details and, when you are ready press ENTER.

2.THE STATE POLICE FORCE.

You can use traditional police methods or use THE DIAMOND COMPUTER NETWORK. Use this network to communicate with other police services, justice departments, prison's and with the National Police.

ACCESS PROCEDURES FOR THE DIAMOND NETWORK.

CODE M (MESSAGE).

A message addressed to any service must include the code for that service and it's base town. The message must include the type of information required in as much detail as possible. The reply from the service contacted will arrive in the form of a listing in a few minutes.

CODE P(PRINTER)

If you have a 80 column printer you will not need to take notes manually. All you have to do to keep track of your evidence is to press the I key. To deactivate press the same key.
N.B.:All messages must be confirmed by pressing ENTER.

SERVICE'S YOU CAN CONTACT ON THE DIAMOND NETWORK.

Another Police Squad.

Computer Code GIE + name of squad.
If you want to contact the St Galmier squad (42) write GIE St Galmier opposite 'ORIG', and then make up your message in the form-INFORMATION ON 'XXXX AFFAIR. If the squad you contact is concerned with that particular case it will reply giving you all the information in it's possession. There is one police squad per town.

District squad for judicial information.

COMPUTER CODE BDRJ.

There is one police squad per district (as opposed to town) which will tell you if a particular person is wanted by the police. You address your requests to BDRJ St Etienne in the following way,-IDENTIFICATION OF DUPONT LEON- and if this person is wanted you will be told by whom and why.

JUDICIAL RESEARCH AND COMPARISION CENTRE.

COMPUTER CODE CRRJ.

If you wish to know the record of someone known to the judiciary or to make a comparision on a particular object which could be a clue, you can contact this service using CRRJ LYON and you should compose your message as follows - INFORMATION ON 'OBJECT'-(as much information as possible) or ON 'A CERTAIN PERSON'(Christian name and Surname). If the person has a record you will get a photo and some information.

POLICE HEADQUARTERS.

COMPUTER CODE PREF.

To get the name of the owner of a vehicle from the registration number, you should address your messagess to PRF St ETIENNE (If it is a 42 vehicle, the number of a vehicle corresponding to a town) then input the message IDENTIFICATION No1111 AA 42, for example. If the registration is usefull to the inquiry you will get the information you require.

PRISON'S

COMPUTER CODE PRIS.

For information on a prisoner address you request to PRIS BEAUMETTES or PRIS ST PAUL, WITH A MESSAGE SUCH AS INFORMATION ON DUPONT LEON. If this is usefull to your inquiry you will get a reply.

POLICE STATION

COMPUTER CODE CIAT

You can call these police stations up by using e.g INFORMATION ON 'XXX' AFFAIR'.

TRADITIONAL METHODS.

OPTION S: STATEMENT
OPTION C: COMPARISON OF EVIDENCE.
OPTION E: VARIOUS EXAMINATIONS.(eg AUTOPSY VERA CRUZ)
OPTION A: ARREST. If you are sure of someones guilt, it only remains for you to arrest them.

FINAL ADVICE:
****

Watch out for wrong arrest.

IN CASE OF DIFFICULTY CHECK YOUR KEYBOARD IS IN CAPITAL LETTERS.

GOOD LUCK.

****

BLAGGER
---

20 SCREENS 4 LIVES

Follow the exploits of Roger the Dodger, master burglar, as he works his way through numerous buildings- searching for golden keys, robbing safe after safe but always on the alert to the many obstacles and alarm systems and those spooky night-watchmen.

When one screen is cleaned out you move in an endless pattern of banks, shops, and houses.

POINTS
---

100 Points per key
Time bonus for completion of each sreen.
Extra man every 10,000 points.

TO MOVE

Left = Z
Right = X
Jump = Shift

Or joystick in port 2

GOOD LUCK.

BEGINNERS BASIC

BY CRAIG HURSEY

VARIABLES

In this issue I will finish my section on Variables and return back to commands next issue. Also in this issue you will find a review on the new MSX turbo R.

Variables are names used to represent in RAM that BASIC uses automatically to store values used in a BASIC program. The value of a variable may be assigned by the programmer, or it may be assigned as the result of calculations in the program. Whenever a variable is first used, a storage area is created to hold it, initialized to zero. The size of this area depends on the type of the variable used.

Variable names may be any length. Up to 2 characters are significant, and the rest ignored (unless you accidentally duplicated a reserved keyword). Variable names can contain letters and numbers. However, the first character must be a letter.

A variable name may not be a reserved word and may not contain embedded reserved words. Reserved words include all MSX BASIC commands, statements, function names and operator names. If a variable begins with FN, it is assumed to be a call to a user-defined function.

All reserved words should be listed in your Computers manual.

Variables may contain either a numeric value or a string. String variable names are written with a dollar sign ($) as the last character For example, in A$="MSX Turbo R", the dollar sign is a variable type declaration character, that is, it declares that the variable will contain a string value. Strings are known internally because they have a "valtype" of 3, which indicates the fact that each string variable or array element, in addition to the name type, takes up three bytes to store the data (not counting the actual length of the character string the variable points to).

Numeric variable names may declare integer, single precision or double presision values. The type declaration characters for these variable names are as follows:

Symbol	Type represented	valtype
%	Integer variable	2
!	Single precision variable	4
#	Double precision variable	8

The default type for a numeric variable name is double precision. The valtype is what BASIC internally uses to determine the type, and is the number of bytes required for every variable or array element of that type to hold its data.

All variables require an overhead of three bytes each to hold the variable name and the valtype.

Examples of MSX BASIC variable names:

PI#	Declares a double precision value.
MINIMUM!	Declares a single precision value.
LIMIT%	Declares an integer value.
N$	Declares a string value.
ABC	Represents a double precision value.

There is a second method by which variable types may be declared.
The MSX BASIC contain statements which allow this. I will explain these in later issues.

Array variables

An array is a group or table of values referenced by the same array variable name. Each element (piece of data) in an array will hold data of the same type. When you turn on your computer, the machine will allow you to use arrays with upto 10 elements in them, numbered 0 to 9, without having to declare that you want to use an array. If more elements are needed then you will have to dimension an array using the BASIC DIM satement.
Thus DIM A(20) will dimension an array called A to have 21 elements, numbered 0 to 20.

When we first dimension an array, each element has the value 0 if it is a numeric array and empty string if it is a string array. Assigning values to an element of an array is quite easy.
Examples.

```
A(1)=1.234
A(2)=3
LET A(4)=23
```

These statements will all assign the appropriate values to the various elements of array A. An array can be one dimensional, like the example array A, or can have muti-dimensions. The best way to describe this type is to think of it like a Table or grid. It is like having rows & columns for the array. Shown below is a diagram of an array A(4,4) which has 4 rows and 4 columns.

```
-----------------------------------------
- A(0,0) - A(1,0) - A(2,0) - A(3,0) -
-----------------------------------------
- A(0,1) - A(1,1) - A(2,1) - A(3,1) -
-----------------------------------------
- A(0,2) - A(1,2) - A(2,2) - A(3,2) -
-----------------------------------------
- A(0,3) - A(1,3) - A(2,3) - A(3,3) -
-----------------------------------------
```

These arrays allow you to calculate things easily or keep tabs on results and keeps the program neat and tidy so it is easy to follow. I will be fully describing the DIM statement and DEFINT/DEFSTR/DEFDBL/DEFSNG as well.

OVERVIEW OF MSX GENERATIONS

A LOOK AT THE MSX GENERATIONS
As you are all aware by now, we are onto the fourth generation of MSX home computers since it all started late 1983 in the early development stage.
I will breifly look at the specifications or changes to all models.

First generation. MSX 1 Release 1984

GRAPHICS: On these was average for the time VDP 9919 which had the following features.
16 colours, 4 screen modes, 32 sprites (single coloured), 40 column display.
SOUND: Was 3 tone generators 8 octaves (AY-3-8910 PSG)
ROM: 32K BASIC MSX
RAM: MIN. 8K , Video RAM of 16K
CPU Z80A 8bit.

Second generation. MSX 2 Released 1986
The improvements on the first are as follows:

GRAPHICS: An improved VDP 9938 (YAMAHA & ASCII) which had 8 screen modes, 512 colours, multi color sprites and 80 column display.
SOUND: The same plus optional FM chip with 3 tone, 8 voices, 8 octaves, this was called MSX-AUDIO
ROM: 48K BASIC version 2
Minmum ram 32K, Video Ram 64K
SAME CPU: Z80A

Third generation MSX 2+ Released 1988 (October)

The following are improvements over the second generation
GRAPHICS: new VDP 9958 (Yamaha & ASCII) which has 12 screen modes, 19268 colours, high speed movement horizontal or vertical movement
SOUND: still original AY-3-8910 PSG plus MSX-MUSIC 9 tone, 8 octave, 64 voices or 5 tone, 6 rhythm, 8 octave, 64 voice FM synth.
ROM: 128K BASIC version 3 (BASIC 3.0,Disk basic,etc)
RAM: 64K, Video RAM 128K
SAME CPU. Built in Disk drive.

Fourth generation MSX Turbo R Released 1990 (December)

GRAPHICS: Same as 2+
SOUND: Same as 2+ plus PCM (Pulse Code Modulator) built in mic.
ROM: 928k BASIC version 4.0 plus JIS1&2, MSX JE, DOS2 etc.
RAM: 256K, Video RAM 128K, S-RAM 16K
CPU: Z80A, R800 16bit

As you can see the MSX standard has come a long way in 6 years.
And has kept its No.1 place in JAPAN.

FOR SALE

SVI-738 with EXTENSION DRIVE & CASE
SVI 767 CASSETTE RECORDER
2 JOYSTICKS
YAMAHA PN101 9 PIN DOT MATRIX MSX PRINTER
GREEN SCREEN MONITOR & 100 DISKS OF SOFTWARE
$750.00

SVI-738 SERVICE & TECHNICAL MANUAL
SVI-728 SERVICE & TECHNICAL MANUAL
YAMAHA PN101 SERVICE MANUAL
YAMAHA CX5 MII SERVIVE MANUAL
YAMAHA FDO5 DISK DRIVE SERVICE MANUAL
YAMAHA YK20 KEYBOARD SERVICE MANUAL
YAMAHA SFG01 MUSIC BIOS MANUAL
PRINT XPRESS MANUAL

10 ASSORTED ROM CHIPS FOR SVI-738 & SVI-728
40 EPROMS FOR YAMAHA CARTRIDGES
POWER SUPPLY BOARDS, STEPPING MOTORS
$150.00 THE LOT

FREIGHT OR POSTAGE TO BE NEGOTIATED.
ALL ENQUIRES TO BE DIRECTED TO: ROD GILLETT, STRINGZ MUSIC
153 MOLLISON STREET, KYNETON, VIC. 3444

MSX TURBO R REVIEW

The MSX turbo R has many new things about it that are mind boggling. (apart from Japanese Manuals) any one out there that can read Japanese?
First things first it, arrived on the 24th December. So I couldn't start hacking on it straight away. Before I forget the model is a PANASONIC A1ST serial Number 4370 so I have an early production model.

The computer is in a dark creamy grey colour. Keyboard is a nice lighter grey colour. The keyboard has a nice feel to it. Although the Sanyo was a bit softer. The layout is about the same with a numeric keypad on the right hand side, a shorter space bar but a couple of extra keys in that space. Although the use of these has eluded me, they may have some thing to do with the Japanese Word Pro and other functions built in.
The most impressive thing was, that when turned on, a menu appears (after the MSX logo) which gives you options to choose from, including Word Processor other filing functions, etc including PCM Tools (Calculator/calendar/time) BASIC , DOS and other funtions.
Some of these functions require the disks that come with the computer.
The menu however, can be turned off by using the switch on the front panel of the computer. This makes it behave like all other MSX computers. As the computer comes with two disks. One is a utility disk containing graphics program. PCM program (not bad), printer program for colour printers. The other MSX-DOS2 master disk (Japanese version).
These programs are designed for use by Japanese, but it does not take long to work out how to use most things. Graphics program being easier than the others. The PCM function is true to life in recording and playback mode through TV or other speakers. the program allows storing or direct play through to TV. The program allows you to speed up or slow down when playing back and saves to disk. In DOS you have 160K for use but in BASIC you have 64K for storage. The PCM function is an extension of MSX MUSIC. New commands to work out (only two of them).
Now a bit about the R800 as far as I can tell it is made by ASCII & YAMAHA it's speed is 28.63 MHz while a Z80A is 3.57 MHz which makes R800 8 times faster than the Z80A.
Since the computer has both Dos systems & Disk basic's to choose simple place in a DOS 1 or DOS 2 formated disk. As the computer checks the disk before Booting up if Dos 1 it boots up DISK BASIC version 1.0 and uses the Z80A CPU. But if a DOS2 disk is in or no disk it boots up to Disk Basic version 2.00 and the R800 CPU is used. Below is a program used for tests between Z80A & R800.

Program 1.

```
10 'Test time taken to perform 0-1000 loop in BASIC
20 '
30 TIME=0
40 FOR LOOP=0TO 1000:NEXT
50 PRINT"Time :";TIME
60 END
```

```
Time :17        (DOS 2  R800)
Time :98        (DOS 1  Z80A)
```

Most commands are performed faster except for Disk operations because disk speeds are set. All machine code programs run normally because they are Z80A programs & not R800. I won't be discussing DOS 2 as it was reviewed in ISSUE 009 of Micro's Gazette.
There really is not much I can say because all other features were done in the review of the MSX 2+.

But before I go. On taking the top off I found that the RAM can be upgraded to 512K/1M plus a extra ROM and a DiskDrive connection on the PCB not connected. What expansion's are planned for the NEW MSX TURBO R range from PANASONIC ???.

Also there are rumours that SONY will be releasing a model in April some time so stay tuned. Well I hope this review gives you some detail of the power of MSX.
Following are the specifications of this computer.

SPECIFICATIONS OF THE PANASONIC A1ST MSX TURBO R

CPU's Z80A (3.579545 MHz)
R800 (28.636360 MHz)

ROM 80KB (NORMAL BASIC)
64KB (MSX DISK BASIC)
16KB (MSX MUSIC)
256KB (16bit JIS 1 & JIS 2)
512KB (MSX JE)
608KB (MSX-JE/Word processor)

Main RAM as standard 256KB
Video Ram as standard 128KB
S-RAM 16KB

Video Display Processor V9958

Sound:
PSG: AY-3-8910A
FM Sound: YM2413
PCM: 8bit 1 cH

Disk Drive 2DD 720K Formated.

All specs of unit.

Keyboard has 93 Keys. (counting Numermic Pad etc.)

RESET Button ; RENSHA controller ; MENU switch

2 - joystick ports
2 - Cartridge expansion ports (one top & one Back)

RF-OUT (NTSC Japanese) (B/W) ; AUDIO OUT ; VIDEO OUT (Japanese NTSC) (B/W)
RGB OUT (Analogue 8 pin) ; S-Video out (picture quality between RF & RGB)
PRINTER Interface (MSX Type) ; (no cassette port)
Built in micro phone & extenal micro phone socket on back panel.

LISTINGS

```
5 ON ERROR GOTO 3000:ON STOP GOSUB 4000
10 SCREEN 0:NOGWAT=13000:WIDTH 40:COLOR
 15,4,4:KEYOFF
20 CLEAR NOGWAT:ON ERROR GOTO 3000:ON S
TOP GOSUB 4000:STOP ON:DIM A$(200,5):GO
SUB2730
30 LP$=CHR$(12)+"DATA INPUT :"
40 CLS:PRINTTAB(6)STRING$(28,"C")
50 PRINTTAB(13)"{¦} TURBOBASE"
51 PRINT"         (C)1987 Robtek Softwar
e""
60 PRINTTAB(6)STRING$(28,"@")
70 PRINTDO;DO;TAB(6)"MENU"
80 PRINTDO"      1. Create new file"
90 PRINTDO"      2. Load file"
100 PRINTDO"       Which ? "
110 ZZ$=INKEY$:IF ZZ$="" THEN 110
120 A=VAL(ZZ$):IF A<1 OR A>2 THEN 110
130 IF A=1 THEN 260
140 CLS:INPUT "Filename ";B$
150 OPEN B$ FOR INPUT AS #1:PRINTDO"Loa
ding file : "B$
160 INPUT #1,N
170 INPUT #1,M
180 FOR X=0TON
190 FOR Y=0TOM
200 INPUT #1,C$
210 A$(X,Y)=C$
220 IF EOF(1)=-1 THEN 240
230 NEXT Y,X
240 CLOSE #1
250 GOTO 450
260 X=0
270 CLS:PRINT"Create new file : Record-
layout":PRINT
280 PRINTDO"6 fields are available per
record":PRINT
290 PRINTDO"Type * to correct and 'stop
' when you":PRINT:PRINT"are ready":PRIN
T:PRINT
300 FOR Y=0 TO 5
310 PRINT"Field #"Y+1;:INPUT A$(X,Y):IF
 Y=0 THEN 330
320 IF A$(X,Y)="*" THEN Y=Y-1:PRINT UP;
UP;:GOTO 310
330 IF A$(X,Y)="stop"OR A$(X,Y)="STOP"
THEN Y=Y-1:GOTO 360
340 A$(X,Y)=STR$(Y+1)+" "+A$(X,Y)
350 NEXT Y:Y=Y-1
360 IF Y<0 THEN RUN
370 M=Y
380 CLS:PRINT"This is your selection":P
RINT:PRINT
390 FOR Y=0 TO M
400 PRINT A$(X,Y):NEXT Y
410 PRINT DO"Good (y/n) ?"
420 ZZ$=INKEY$:IF ZZ$="" THEN 420
430 IF ZZ$="N"ORZZ$="n" THEN PRINTDO;DO
"Then start again...":FORY=0TO2000:NEXT
Y:GOTO 260
440 IF ZZ$<>"Y"ANDZZ$<>"y" THEN 420
450 PRINTDO:PRINT"SELECTION  H   "LE;LE
;LE;LE;LE;LE;:INPUT A$
460 IFA$="1"THEN590
470 IFA$="2"THEN750
480 IFA$="4"THEN970
490 IFA$="5"THEN1200
500 IFA$="h"ORA$="H"THEN1330
510 IFA$="e"ORA$="E"THENCLS:NEW
520 IFA$="3"THEN1460
530 IFA$="7"THEN1530
540 IFA$="9"THEN1640
550 IFA$="8"THEN1850
560 IFA$="6"THEN2010
561 IFA$="BAARD"THEN 9999
570 PRINTDO"COMMAND NOT KNOWN"
580 GOTO 450
590 PRINTLP$:PRINT
600 IFA$(1,0)=""THEN620
610 PRINTDO;DO"Data already present!! "
:GOTO 450
620 FOR X=1TO200
630 PRINTDO;USING "Record nr.(###)";X;:
PRINTDO
640 FOR Y=0 TO M
650 NL=LEN(A$(0,Y))
660 PRINT A$(0,Y)TAB(15)": ";:INPUT A$(
X,Y)
670 IFA$(X,Y)="*"THENY=Y-1:PRINTUP;UP;:
GOTO 660
680 IFA$(X,Y)="stop"ORA$(X,Y)="STOP"THE
N 720
690 NEXTY
700 IF X/2=INT(X/2) THEN PRINT LP$:PRIN
TDO;DO
710 NEXT X
720 N=X-1
730 PRINTDO;N"RECORDS PRESENT"
740 GOTO 450
750 CLS:PRINT"SHOW FILE CONTENTS
760 PRINT DO;DO"Which record(s) do you
want to see"
770 PRINTDO;DO"1. All":PRINT"2. One":PR
INT"3. Range"
780 PRINTDO"Which ?"
790 ZZ$=INKEY$:IF ZZ$="" THEN 790
800 A=VAL(ZZ$):IFA=1ORA=2ORA=3THEN820
810 GOTO 790
820 ONAGOSUB930,940,960
830 CLS
840 FOR X=ATOB
850 PRINT USING"Rec. nr (###)";X;:PRINT
DO
860 FOR Y=0TOM
870 PRINT A$(0,Y)TAB(15)": "A$(X,Y)
880 NEXTY
890 PRINT STRING$(39,"-")
900 ZZ$=INKEY$:IFZZ$=""THEN900
910 NEXT X
920 GOTO 450
930 A=1:B=N:RETURN
940 PRINTDO:INPUT"Which Record nr. ";A
950 B=A:RETURN
960 PRINTDO:INPUT"Start rec.nr., End re
c.nr. ";A,B:RETURN
970 CLS:PRINT"CHANGES"DO
980 PRINT"1. Add records":PRINT"2. Chan
ge records":PRINTDO"   Which ?";
990 ZZ$=INKEY$:IFZZ$=""THEN990
1000 A=VAL(ZZ$):IFA<1ORA>2THEN990
1010 PRINTA:IFA=2THEN1130
1020 FORX=N+1TO200
1030 PRINTDO;USING"(###)";X;:PRINTDO
1040 FORY=0TOM
1050 PRINTA$(0,Y)TAB(15)": ";
1060 INPUT A$(X,Y)
1070 IF A$(X,Y)="*" THEN Y=Y-1:PRINTUP;
UP;:GOTO 1050
1080 IF A$(X,Y)="stop"ORA$(X,Y)="STOP"
THEN 1100
1090 NEXT Y,X
1100 N=X-1
1110 PRINTDO"Total number of records :"
N
1120 GOTO 450
1130 GOSUB 2410
1140 PRINTDO:INPUT"Change which field (
0=stop) ";Y:PRINT
1150 IF Y=0 THEN 450
1160 IF Y<0 OR Y>M+1 THEN PRINTUP;UP;UP
;:GOTO 1140
1170 PRINT A$(0,Y-1)TAB(15)":  "A$(X,Y-
1):PRINTTAB(16);UP;
1180 INPUT A$(X,Y-1)
1190 GOTO 1140
1200 CLS:PRINT"Search":C=0
1210 GOSUB 2320
```

```
1220 PRINTDO:INPUT"Search for";A$:CLS:P
RINTA$" searching...":PRINT
1230 FORX=1TON
1240 IFLEFT$(A$(X,Y-1),LEN(A$))=A$THENC
=1:GOSUB1280
1250 NEXT X
1260 IFC=0THENPRINTA$": NOT FOUND"
1270 GOTO 450
1280 PRINTDO;USING"(###)";X;:PRINTDO
1290 FORA=0TOM
1300 PRINTA$(0,A)": "A$(X,A)
1310 NEXT A
1320 RETURN
1330 CLS:PRINT"MENU"
1340 PRINTDO"1. Input data in a new fil
e"
1350 PRINT"2. See record contents"
1360 PRINT"3. List per field"
1370 PRINT"4. Change or add records"
1380 PRINT"5. Search on a field"
1390 PRINT"6. Sort (alphabetic) per fie
ld"
1400 PRINT"7. Print record contents"
1410 PRINT"8. Delete records"
1420 PRINT"9. Save file"
1430 PRINT"H. Help - list of commands"
1440 PRINT"E. End"
1450 GOTO 450
1460 CLS:PRINT"LIST"DO
1470 GOSUB 2320:CLS:PRINT"List on ";:LR
$=RIGHT$(A$(0,A-1),LEN(A$(0,A-1))-2)
1480 PRINTLR$:PRINT
1490 FOR X=1TON
1500 PRINTX":"A$(X,Y-1)
1510 ZZ=ZZ+1:IFZZ=20THENZZ=0:GOSUB2530:
CLS:PRINT"List on "LR$:PRINT:GOTO1520
1520 NEXT X:ZZ=0:GOTO 450
1530 CLS:PRINT"PRINT RECORDS ON PRINTER
"DO
1540 GOSUB 2560
1550 :
1560 :
1570 :
1580 :
1590 :
1600 :
1610 GOTO 450
1620 :
1630 :
1640 CLS:PRINTDO"SAVE"
1650 PRINTDO:INPUT"File name ";B$
1660 OPEN B$ FOR OUTPUT AS #1:PRINT"Sav
ing : "B$
1670 PRINT #1,N
1680 PRINT #1,M
1690 FOR X=0TON
1700 FOR Y=0TOM
1710 A$=A$(X,Y):PRINT #1,A$
1720 NEXT Y,X
1730 CLOSE 1
1740 PRINTDO;:INPUT"VERIFY (y/n) ";A$:I
FA$=" N"ORA$="n" THEN 450
1750 REM:PRINTDO"REWIND CASSETTE & PRES
S ANY KEY"
1760 REM:A$=INKEY$:IFA$=""THEN1760
1770 OPEN B$ FOR INPUT AS #1:PRINTDO"CO
NTROL "B$
1780 INPUT #1,N:INPUT #1,M
1790 FORX=0TON:FORY=0TOM:INPUT #1,A$
1800 IF EOF(1)=-1 THEN 1840
1810 IF A$(X,Y)<>A$ THEN PRINTDO"WRITE
ERROR, PLEASE SAVE AGAIN":GOTO 1840
1820 NEXT Y,X
1830 PRINT"FILE O.K."
1840 CLOSE #1:GOTO 450
1850 CLS:PRINT"DELETE"DO;DO
1860 INPUT "One record (1), Several rec
ords (2) ";A$
1865 IF A$<"1" OR A$>"2"THEN PRINTUP;UP
:GOTO 1860
1866 A=VAL(A$)
1870 IFA=2THEN1980
1880 GOSUB2410
1890 A=X:B=X:C=1
1900 FORX=ATON-1
1920 FOR Y=0TOM
1930 A$(X,Y)=A$(B,Y)
1940 NEXT Y
1950 NEXT X
1960 N=N-C
1970 GOTO 450
1980 PRINTDO:INPUT "Record : Start, End
";A,B
1990 C=B-A+1
2000 GOTO 1900
2010 CLS:PRINT"SORT"DO
2020 GOSUB 2320
2030 Y=Y-1
2040 PRINTDO"File is being sorted"
2050 GOTO 2130:IF MID$(A$(0,A-1),LEN(A$
(0,A-1))-1,1)="a" THEN 2130
2060 C=0
2070 FORX=1TON-1
2080 IFVAL(A$(X,Y))<=VAL(A$(X+1,Y))THEN
2100
2090 GOSUB2210
2100 NEXT X
2110 IFC=1THEN2060
2120 GOTO 2190
2130 C=0
2140 FORX=1TON-1
2150 IFA$(X,Y)<=A$(X+1,Y)THEN2170
2160 GOSUB2210
2170 NEXTX
2180 IFC=1THEN2130
2190 PRINTDO"File has been sorted"
2200 GOTO 450
2210 FOR B=0TOM
2250 SWAP A$(X,B),A$(X+1,B)
2290 NEXTB
2300 C=1
2310 RETURN
2320 PRINTDO"These are the fields :"DO:
FORY=0TOM:PRINTA$(0,Y):NEXT Y:PRINT
2330 PRINT"On which ?":A$=INKEY$:IF A$
<"1"OR A$>"6" AND A$<>"S" AND A$<>"s" T
HEN PRINTUP;:GOTO 2330
2331 A=VAL(A$)
2335 IF A$="S" OR A$="s" THEN 450
2340 PRINT A$(0,A-1)
2350 PRINTDO"Correct (y/n) "
2360 A$=INKEY$:IFA$=""THEN2360
2370 IFA$="n"ORA$="N"THENCLS:GOTO 2320
2380 IFA$<>"y"ANDA$<>"Y"THEN2360
2390 Y=A
2400 RETURN
2410 PRINTDO:INPUT"Which record nr. (0=
stop) ";X
2420 IF X=0 THEN 450
2430 IF X<1 OR X>N THEN PRINTUP;UP;:GOT
O2410
2440 PRINTDO;USING"(###)";X;:PRINTDO
2450 FORY=0TOM
2460 PRINT A$(0,Y)": "A$(X,Y)
2470 NEXT Y
2480 PRINTDO"Is this the one (y/n) ?"
2490 A$=INKEY$:IFA$=""THEN2490
2500 IFA$="n"ORA$="N"THEN2410
2510 IFA$<>"y"ANDA$<>"Y"THEN2490
2520 RETURN
2530 PRINTDO"Press any key"
2540 ZZ$=INKEY$:IFZZ$=""THEN2540
2550 RETURN
2560 PRINTDO;DO"Which record(s) do you
want to see"
2570 PRINTDO;DO"1. All":PRINT"2. One":P
RINT"3. Range"
2580 PRINTDO"Which ?"
2590 ZZ$=INKEY$:IFZZ$=""THEN2590
2595 IF ZZ$<>"1" OR ZZ$<>"2" OR ZZ$<>"3
" THEN PRINTUP:PRINTUP:GOTO 2580
2600 A=VAL(ZZ$):IFA=1ORA=2ORA=3THEN2620
2610 GOTO2590
2620 ONAGOSUB930,940,960
2630 CLS:PRINT"Printing..."
2640 FORX=ATOB
2650 LPRINT USING "Record number (###)"
;X:LPRINT
2660 FOR Y=0TOM
```

CONTINUED ON PAGE 13