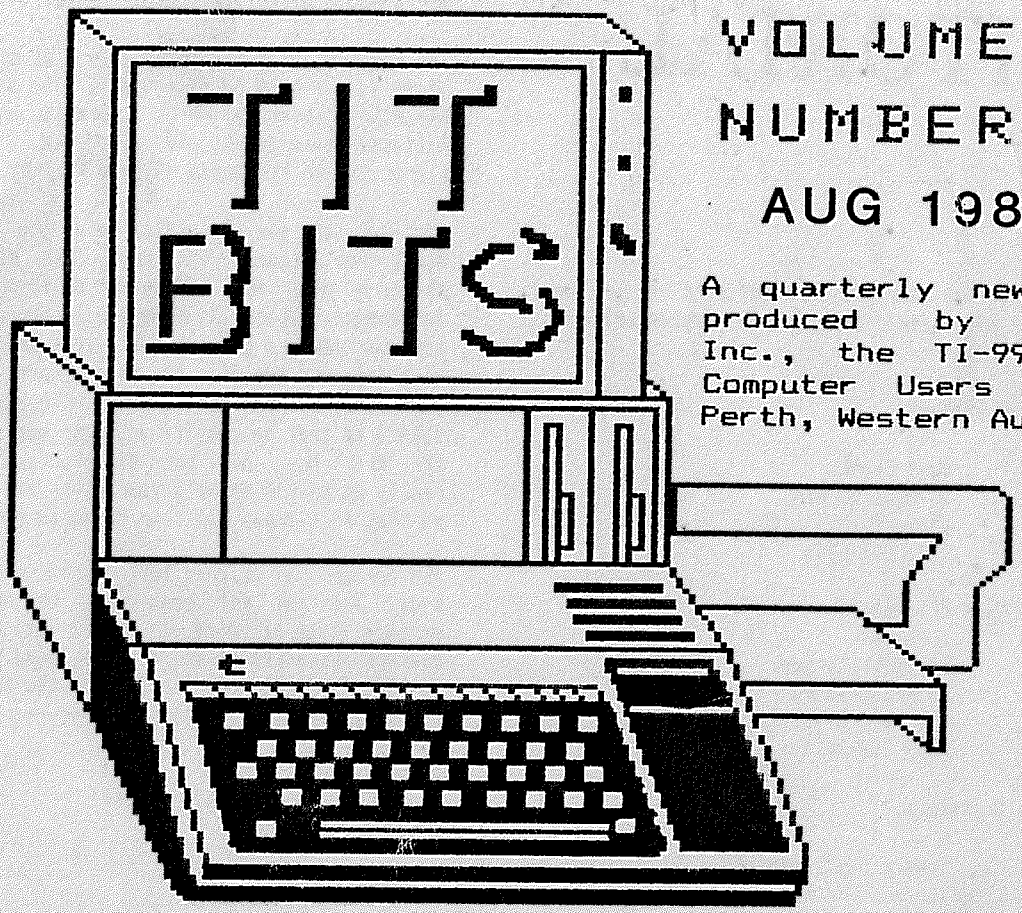


VOLUME-5
NUMBER-1
AUG 1986

A quarterly news letter produced by T.I.U.P. Inc., the TI-99/4A Home Computer Users Group of Perth, Western Australia.

T.I.U.P. Inc. is a non-profit group of TI-99/4A Computer Users not affiliated in any way with Texas Instruments.

Subscription :
$15.00 (Aus) per annum, full member.
$10.00 (Aus) per annum, Newsletter member only.

99 TIUP

CONTENTS

Letters to the Editor may be address to :-

The Editor
PO BOX 246
Mt Lawley
Western Australia.
6050

EDITORIAL

Another year has come and gone and with it a new committee has been elected. The members of your committee are:

President.

Dave Clark.
56 McGlinn Way
Cloverdale. 6105.
(09) 277-5296

Vice President.

Steve Wilkinson.
38 Homestead Rd
Gosnells 6110.
(09) 398-2079

Secretary.

Robert Hollingsworth.
16 Jennifer Way
Rossmoyne 6155
(09) 457-1538

Treasurer.

Phil West.
56 Risby St
Gosnells 6110
(09) 398-6592

Newsletter Editor Disk Librarian.
Les Twiss.
32 Diosma Way
Forrestfield 6058
(09) 453-6837

Committee.
Frank Graham.
20 hudson St.
Bayswater 6053.
(09) 271-5972.

Dave Whitney. Module Librarian.
261 Railway Rd
Subiaco 6008
(09) 381-9370

Erik Van Garderen.
9 Corser St
Kewdale 6105
(09) 362-4779

As you can see from the above list the committee has not changed drastically from last year. The only new member is Frank Graham. I would like to welcome him officially to our ranks.

The position that has changed is that of President, which is now held by Dave Clark. The positions of Cassette Librarian, Disc Librarian and Technical Librarian have yet to be allocated. The module Library is in the hands of Dave Whitney.

As you can see I still hold the position of Newsletter Editor. The newsletter will not be changing very much in content or format, at least for the next year. As this newsletter is a forum for our members to air their views and news on the TI-99/4A, and the only way we can carry on is for you to send in your GRIPES, PROBLEMS, HINTS and IDEA'S so that we can try and help our members get the best they can out of this machine. It is really up to you to help yourselves and help others so that this Group can keep going in the future.

The Group has at long last become an Incorporated body. This may not mean much to many of our members but it does at least give the Office Bearers indemnity if the Group is sued. Our thanks to Steve Wilkinson for seeing the Incorporation through, and handling all the paperwork that was involved.

Les Twiss.

FREEWARE

New in from the states is several disks of FREEWARE. If you are new to this concept, Freeware is software that can be copied freely, if you like the program you send the author a donation, which is usually $5 or $10. The author's name and address usually appears on the title screen of the program. This is a good concept, so don't abuse it. If you like the program send the author the suggested donation. He or She may even be tempted to write more programs. This is one way of helping our Computer to stay alive.

Below is a copy of some of the FREEWARE programs that the Group has available.

1.* DM1000-Ottawa 99/4 U.G. ....P.O. Box 2144, Station D, Ottawa, ONT. K1P5W3. This disk manager rivals CORCOMP'S version, and will make your TI Disk Manager II worthless!

Requires a complete disk.

2.* ENV. AND CATALOG PGM-Trio+ Software.... P.O. Box 115, Liscomb, Iowa 50148. An excellent disk cataloging program that allows you to make your own disk envelopes with your own comments about the individual programs on that disk. Be sure to READ the "Hardcopy Instructions" before trying to use it. The envelope for my freeware (Item 13 above) was done with this program.

3.* MASS-TRANSFER-Stuart Olson .... 25322 W. Wayside Place, Lake Villa, IL 60046. An assembly language Terminal Emulator, menu drive, x-modem transfers, capable of multiple X-modem transfers all at once. Asking $10 plus your disk and a reuseable mailer.

Requires a complete disk.

4.* FUNLWRITER-Funnelweb Farm ....Tony & Will McGovern, 215 Grinsell St., Kotara, NSW 2288 AUSTRALIA. Requires 2 disks for documents and programs. Version 3.2 was issued in March 1986. I have tried this one, and it will do everything TI-WRITER will do, plus more! And it uses Extended Basic rather than the TI-WRITER cartridge. Tony & his son Will says "you will end up leaving your TI-WRITER cartridge in the drawer"! This one contains version 2.3 of DM1000, DISKO, and E/A all commingled and interconnected-don't attempt to separate them.

5.* NOTEBOOK

6.* DATABASE PROGRAM. Written by Phil West of TI-UP. Requires 2 disks for documents and programs. This program is an excellent data base program written in X-Basic and using Assembly language routines. If you have ever tried to keep a mailing list of any type, then you know the problems that are associated with that task. This program will solve those problem. The Disk comes complete with the assembly language source code, the data base program and 2 associated programs that are useful for converting other types of data base files so that this one can read from them or write to them. These programs are reviewed and listed in the program section of this newletter.

All the above Disks are available from the User Group. To get Your Copy send $5.00 (Aus) for the each disk required, this is to cover postage, the disk itself and copying charges. Any residue amount will go into the Clubs resources so that further Free ware can be brought in from the States.

All enquiries, regarding these disks should be made to the :

Disk Librarian TIUP.
P.O. Box 246.
Mt Lawley
West Australia.
6050.

Or contact Les Twiss, by phone on (09) 4536837.

Please allow time for copying and postage of the return disks. This may be up to two weeks from the time YOU send your request.

NOTE !!! All disks will be copied Single Sided Single Density, so that all TI USERS will be able to load and run these files.

PRINTER

The CLUB has a printer and RS232 interface that may be hired out by any member. The only requirements to use this printer is that you have an extra power point or two and a TI.99/4A computer. Its really easy to set up on your machine, all you have to do is plug it in the right hand side.

To hire the printer, contact Rob Hollingsworth, on 09-457 1538. The cost of hire is a measly $2.00 per fortnight. This minimal charge cover the cost of replacement ribbons and maintenance. Paper requirements are :

a) Any 80 column form feed paper.
b) A4 or Quarto single sheet.

The printer will use either of these with ease. Cost of hire does not include paper.

NOTE!

You must be a financial member of TI-Users Perth Inc. to make use of this club facility.

CASSETTE LIBRARY

The club has a number of new cassette tapes that you may either purchase for the cost of $3.00 per tape, or bring along your own and get it copied free of charge at the meeting. All the programs from the last two TIT-BITS are now available for you.

Some problems have occured in recent times with tapes copied by the Club, these problems cannot be traced to the tape copier. It seems that if the tape that you want copied to is not ABSOLUTELY CLEAN then the new programs are copied over the existing ones and that's where the problems will occur. So if you bring along your own tapes for copying make sure that they are either new or have had all the previous data that was stored on them is erased.

NEWSLETTERS

The Club receives a number of other club magazines each month. These contain many type of programs and articles that may never get into our own newsletter, we like to keep ours as original as possible, so if you want to get copies of these programs the club photocopier is available at meetings to copy these programs, articles etc. at a nominal charge of 10c per sheet. The Newsletters are held in the Technical Library and are at each meeting.

NOTE!

These newsletters must not be removed from the meeting, for any reason what-so-ever.

MODULE LIBRARY

The MODULE LIBRARY has been going great guns since it was introduced last year. If you haven't made use of it yet, then you are missing out on many a good game or applications package. It has been so successful that the Club has bought a quantity of new modules recently to add to the already impressive line-up.

Country members don't feel left out. YOU can also borrow these modules. Just drop a line to the Module librarian:

Dave Whitney
c/o TI-99 Users Perth Inc.
P.O. Box 246
Mt Lawley 6050

or Phone Dave on (09) 381-9370

The cost of hire of these module still remains a $1.00 per month, with a fine of $2.00 per month for late return. If for some unforeseen reason you are unable to attend the meeting, contact any committee members, and the module you have will be booked out to you for another month. This could save you $1.00 on the fine.
Country members, and additional charge for the cost of postage and packaging will have to be added to the normal hire charge. At current rates this is approx. $1.20, for 1 module and $2.00 for two modules.

NOTE !

No more than 2 modules per member per month may be loaned out. This is to ensure that everyone has a fair go. A list of the Club modules can be found elsewhere in this newsletter.

DISK LIBRARY

The Club disk library has just received a number of new disks. These include many of the FREEWARE disks that are listed in this newsletter. If you want copies of these disks or to find out what's new, then contact the DISK LIBRARIAN :

Les Twiss
c/o TI-99 Users Perth Inc
P.O. Box 246
Mt Lawley 6050.

or phone Les on (09) 453-6837.

The club charges $3.00 per disk, and a copying fee of 50c per side. If you bring along your own disks it will cost you only 50c per side copied.

When ordering disks by mail, an added $1.00 per disk is charged to cover all postage and packing charges. All disc mailed will be sent Airmail where ever possible.

NOTICE OF MOTION

At the last general meeting a motion was passed, that the names and addresses of all members is to be circulated via the TIUP FLYER. A period for objection to your name appearing in this list was granted, this period expires on the 20th Sept 1986.

If you object to your NAME and ADRESS being circulated in this fashion send a LETTER of OBJECTION to the President :

The President
TI-99 Users Perth Inc
P.O. Box 246
Mt Lawley 6050.

All objections must be in writing and any correspondence received after this date will be deemed void.

10X CHARACTER SET

CHARACTER GENERATOR FOR GEMINI 10X/15X PRINTERS

Author Steve Wilkinson, T.I.U.P.
38 Homestead Rd,
Bosnells,
Western Australia.
6110.

This program is designed to make full use of the Gemini 10X/15X " Downloadable Character Set " option. It gives you the ability to create and save, to disk, all your special characters. Previously created files can be added to or modified.
A 9 X 9 grid is displayed on which the cursor can be moved around filling in or deleting squares as you go. Commands listed at the bottom of the screen can be called up by selecting the first letter of the command.
The " Saved File " can be loaded to the printer on it's own or loaded over the standard character set. Characters undefined will be null.
The ability to provide characters with decenders has been included.
Characters can be tested at the time of creation and also when loading the file to the printer. I had initially contemplated recalling the characters from the file and displaying them on screen. I feel that for the number of times this option would be used it was not worth the effort.
Besides the help file another file 10X-CHRS has been included with a few characters on file.
The program has been configured to address the PIO port but could easily be changed to the RS.232 port.
I have given some little thought to modifying this program for the Epson FX-80 and I think it may be possible to fully or partially implement it. As I do not have an FX-80 to play with it will stop their at this stage.

Options and Key Functions

" Change Character # " this option is to alter the ASCII number of a character that has been created previously in a file and give it a new number , the old number is deleted.. For example the original character number is ASCII # 36, the new number is ASCII # 120, now ASCII # 36 is deleted and ASCII # 120 takes on ASCII # 36 s attributes.

* Note ASCII characters must be between 32 and 126.

" Create Character " this option allows the user to create any ASCII character between # 32-126.

Keys used :
FCTN (2) = " Insert ".To change the cursor from transparent to black.
FCTN (1) = " Delete ".To change the cursor from black to transparent.
FCTN (E) = Move cursor UP.

FCTN (X) = Move cursor DOWN.

FCTN (D) = Move cursor RIGHT.

FCTN (S) = Move cursor LEFT.

* Note the cursor will wrap around the grid top, bottom and sides.

Only the first letter is to address the " ON SCREEN " commands. ie : "C"= Clear , "M"=Menu.

These words are :

" Clear " this clears any existing pattern on the screen and returns the cursor to the top left.

" Decenders " is only to be selected once you are sure the character created is correct. On pressing "D" the pattern will effectively be moved down two places then immediately saved to file.

" Menu " returns to the Main Menu.

" Save " saves the created character to file. On completion of " Save ", options are given to allow you to do the Next,Previous,Repeat and New numbers, also you may test the character direct to the printer. * Note a slight difference in the

vertical position of the character is given as the std. character set and the download set may not be printed on the same line at the same time, so a line feeds of 1/144" has been introduced to mask this effect.

" Load Printer " gives you the ability to load a previously created file into the printer " RAM " with or without the std. character set.

* Beware if you have just created some characters and tested them to the printer then load a " Saved File " the tested characters will be in RAM. To be absolutely sure the loaded character set will not be corrupted switch off the printer prior to loading.

Jan '86.

FUEL

TANK

Capacity Calculation

The reason that this program was created, was that I needed to know the volume of the fuel tank of my '57 Daimler Freeline bus.
The amount of fuel in the tank is quite important, as with any vehicle. You don't just tow a 9 tonne vehicle to the nearest garage, especially on the Nullabour Plain.
To fit a new fuel gauge would be expensive and would probably not be linear. The non linearity is due to the fact that the tank is a cylinder on it's side, (like a 44 gallon drum).
Any gauge that I fitted would need to be calibrated anyway. So even if I did get a gauge I would still need to know the relationship between the level in the tank and the amount of fuel in the tank.
The program is really straight forward as you can see. Find the area of the ends of the tank covered by the fuel and multiply it by the length.
The point that I found intrigueing was that the result SIN and COS of a ratio is expressed in Radians, (Radians !! I have not touched them since High school), not in degrees. I suppose I was expecting to find the same response from the computer as I would get from my humble $10:00 K-Mart Scientific calculator. Maybe some one could explain why T.I. did it this way ?
Anyway having come to terms with radians the problem then sort of fell into place.

As I do not have Graphix I have been unable to draw you the necesssary diagrams to fully explain the following program.

DATA-BASE

FOR TI

INTRODUCTION.

This program requires the following equipment or hardware to be execited fully.

1) 32k Memory expansion.
2) Extended Basic.
3) At least one disk drive.
4) RS232 interface and printer are optional but most desirable.

DIMENSIONS

Approx 1000 records per file.
15 User definable flags.

FILENAMES

All files associated with this program are prefixed with the characters 'DB'.

Data filenames are prefixed with 'DBD'.
Flag filenames are prefixed with 'DBF'.

Command filenames are prefixed with 'DBX'.
Data files are in DISPLAY FIXED 128 FORMAT.
Command and Flag files are in DISPLAY VARIABLE 80 FORMAT.

COMMANDS

All commands are 2 characters long.
Some commands use a third character to pass a parameter and some commands use an additional parameter separated from the command by a '/'.

This program was designed mainly as a Club Address Register. Other uses are possible but modifications to the main program have to be made.

GETTING STARTED.

There are two ways in which this program may be loaded :

1) By entering RUN "DSK1.DBPROGRAM"
2) From the menu program supplied with this disk.

After the DBPROGRAM has started executing there is a slight delay as the program takes a few seconds to initialize.

Once the main screen is set up the cursor will appear in the bottom left hand corner. It is at this point that data may start being entered. All commands used are two character commands that may require some parameter to be set. These parameters are separated from the command by the " / ".

On the initial use of this program it is required that a file be opened so that your data can be read and written to that file.

EXAMPLE.

GD/TEST

The command "GD" is the abbreviated form of "Get Data", the "/" is the delineator, and "TEST" is the name of the data file that is to be used.

This causes the system to look for a file called "DBDTEST" in drive 1. The message "LOADING DATA FILE" will appear in the command prompt line. If this file exists, it will be loaded and the message "XX RECORDS FOUND" will appear. If the file does not exist it will be created, and the message "0 RECORDS FOUND" will appear at the command prompt line.
The file that is created is named "DBDTEST".

NOTE !!!

No data entry or manipulation can occur until a data file is loaded or created.

Once a data file is loaded, or created, further record editing or record creation can take place.

CREATING RECORDS.

To create a record that is to be added to the data file the command "CR" is used.

Up to 28 characters can be entered in the name field.

WRITING RECORDS.

Once a record has been created by the "CR" command, that record must be written to the file. This is accomplished by using the "WR", Write Record, command. If this command is not used immediately after the record has been :

a) Created
b) Edited.

Then that record will not be :

a) Written to the file. (ie lost)
b) The newly edited record will not be written to the file, and the old record will be retained.

EDITING RECORDS.

This command is similar to the "CR" command except that it allows the editing of an existing record. The record to be edited must be displayed on the screen. This can be accomplished in one of two ways:

a) By using the "FR" command.
b) By using the "GR" command.

FINDING RECORDS.

This command allows a record to be found. A parameter must also be given. For example :

FR/T

Will find all the records in that file starting with the letter "T".

NEXT RECORD

This is used after the "FR" command has been used. It will display the next record in sequence that has been found with the "FR" command. It can be used to locate a record for editing or deletion.

PREVIOUS RECORD.

This is the opposite to the "NR" command. It will display the previous record that was found using the "FR" command.

DELETEING A RECORD.

This command will delete a record from the file. The record to be deleted must be displayed on the screen by using the "FR" or "GR" commands.

RECOVERING A RECORD.

This command will attempt to recover all deleted records that may exist in the file.

GETTING A RECORD.

The "GR" command is used to display a given record by using a number parameter. It takes the form of :
GR/12

DEFINE PRINTER.

The "DP" command is used to define the printer.

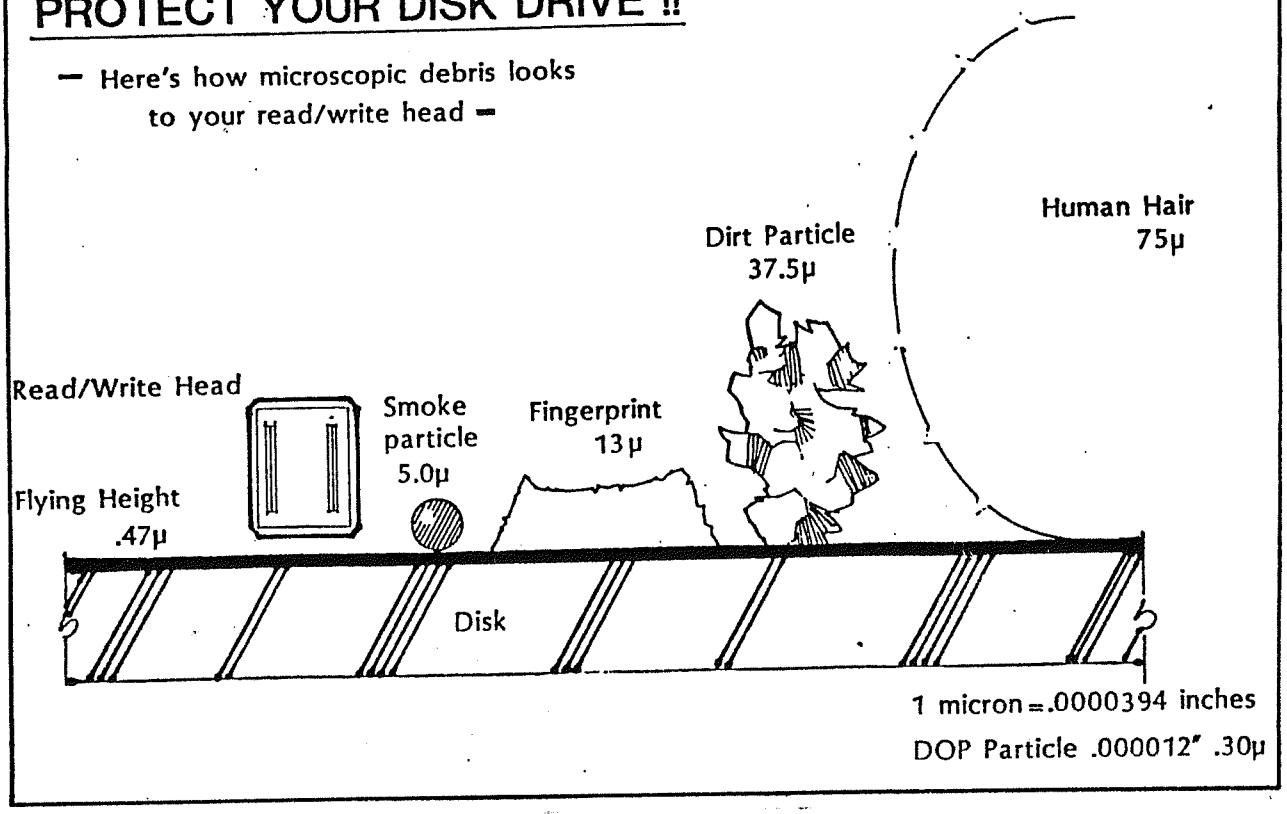
DP/RS232.BA=4800

For more information of the operation of this program send for our freeware disk.

This disk contains a comprehensive manual on the complete operation of this program.

PROTECT YOUR DISK DRIVE !!

— Here's how microscopic debris looks to your read/write head —



OF DISKS AND DRIVES.

Did you realise that your disk drive is a relatively fragile device. If you have a good look at the above diagram, you will see the size relationship of different dirt particles compared to the disk and the read write head. These are actual size representations so care in handling your precious disks is imperative.

WHAT IS A DISK.

A disk is manufactured from a thin sheet of mylar plastic, which is coated with a ferromagnetic material, in the same way as a cassette tape. This this disk of mylar is protected from the outside world by a tough flexible plastic case. Between this case and the disk itself is a layer of lint free material that is used to clean the polished surface of the disk to ensure that it stays relatively free of dirt.

It is a bad policy to to modify any disk in the manner of a "FLIPPY-FLOPPY", that is to turn the disk over and access the second side by punching the appropriate notches in the disk cover.
This causes the disk to turn in the opposite direction to it's normal rotational direction, and causes the lint free layer to release the dirt from the other side of the disk into your system. Although this practice is quite widely used it could cause you serious problems.

DISK DRIVE PROBLEMS.

There are many causes for disk problems on the TI, listed are a few that may occur regularly:

1) Disk cannot be READ. This may be caused by a number of simple faults, such the disk drive door is not shut or not shut fully. There is no disk in the disk drive, the disk is not formatted or the named file does not exist on the disk in that drive.

2) The disk cannot be WRITTEN TO. This may be caused by the disk drive door being open, no disk in the disk drive, the disk having a WRITE PROTECT TAB on, the disk has been filled with data or the disk has not been formatted.

If either of the above problems occur then the appropriate action should be taken. If that fails to cure the problem, then there may be other causes that are harder to fix. These may include:

1) Data on the disk has been corrupted by the disk coming near a magnetic or electromagnetic source. This will cause TOTAL loss of data that is NOT recoverable.

2) The READ-WRITE slot on the disk has been touched by SOMEONES finger causing irretrievable loss of data.

3) The wanted file has been accidently erased. This may be recoverable.

4) The Disk has been dropped onto a hard surface. This also can cause irretievable loss of data.

Generally disks are hardy devices but all care should be taken to protect them from:

1) High temperatures.

2) Dust and dirt.

3) Bumps and knocks.

4) Excessive bending.

REGULAR MAINTENANCE.

Disk drives contrary to popular belief do not require reuglar maintenance. Most manufacturers do not even recommend that the read write head is cleaned. This could actually cause more problems that it cures.

If you are having problems with your disk drives and the above remedies do not work then your disk drive may need alignment. This is a task that involve accurately setting the read-write head so that it is correctly positioned to read the data on the disk. This is a job for a specialist and should not be attempted by the average handy person.

TI DISK DRIVES.

The disk drives fitted to the TI-99/4A are IBM compatible drives. That is they are set up in a configuration that is the same as that used by IBM. It does not mean that they will reads disks that are for IBM use only.

They have 48 TPI (Tracks per Inch), which is a recognised standard for single sided single density disk usage. A standard drive as was supplied by TI for their PEB, was only capable of storing 90 kbytes of data. The standard disk drive controller, is capable of operating on both single sided single density and double sided single density disk drive units. The TI standard controller can only handle a maximum of three disk drives per machine.
A disk drive controller maufactured by CORCOMP for the TI is capable of handling double sided and double density data formats, with the appropriate disk drive fitted and can handle up to four disk drives per machine.

DATA STORAGE.

The TI disk can be formatted in one of the following ways. Using the standard disk controller card supplied by TI.

1) Single Sided Single Density. (SSSD). Which gives 90 kbytes of data storage per disk.

2) Double Sided Single Density. (DSSD). Which gives 180 kbytes of data storage per disk.

With the Corcomp controller replacing the TI controller the following is possible.

1) As in 1 and 2 above.

2) Single Sided Double Density. (SSDD). Which gives 180 kbytes of data storage per disk.

3) Double Sided Double Density. (DSDD(. Which gives 360 kbytes of data storage per disk.

The SSSD Disks used by the TI are formatted with 40 concentric tracks. Each track is divided into 9 sectors or areas, each of these areas hold 128 bytes of data. That is the equivalent of 128 characters per sector.

DSSD disks are formatted with 40 tracks per side and again each track is divided into 9 sectors and holds 128 bytes of data.

SSDD disks are formatted with 40 tracks, which is divided into 9 sectors per track. Each sector can hold 256 bytes of data, which is twice the amount of data for SSSD

DSDD disks are formatted with 40 tracks per side. Each track is divided into 9 sectors and each sector can hold 256 bytes of data.

Disk drives that are capable of formatting DSDD can read any of the other lesser formats. DSSD drives can also read the lesser formats but not the DD types.

SSSD drives can only read a single side of a DSSD format and there own format.

Care must be exercised when transporting disks from drive to drive as they must be of the same format or of a lesser format so that they can be read correctly. Typicall all software is distributed in the SSSD format so that all systems can read the stored data.

LOVE TRIANGLE IN MT LAWLEY

Bern's "love affair" with computers began in the Spring of 1966. September to be exact. That was when he "casually picked up" an edition of "Scientific American" in a newsagency and thumbed through it. As it turns out, it just happened to be the annual "BUMPER" issue, which, in that particular year, was devoted entirely to computers. He bought it for the nominal sum of .65c Australian which was equal to .60c American (Compare that with the current exchange rate for the Dollar) took it home and read it. He was hooked!

Being only three years into marriage and with an active child underfoot, this tantalizing "affair" had to be put on hold, as the "mistress" was too expensive to be entertained. Meanwhile, the glossy magazines kept on piling up, and a man could always dream. Then in 1973 computer classes beckoned. Discreet meetings held at evening Technical College. Secret passwords were learned such as "LET", "GOTO" and "RUN". new languages were used and, in due course a "Data Processing Certificate" was born.

Coloured punch-cards and reams of Print-out decorated the lounge-room with gay abandon. The child (previously underfoot) was now attending computer classes at school and adding more punch-cards and paper to the general turmoil. This was all very pleasing, but "satisfaction" was a long way off. He had to have "her" materialize in the home. It wasn't going to be easy. Jealousy prevailed. His better-half was dragged into a shop on the pretext of buying some fruit, It was indeed an "APPLE" but they wanted $3000.00 for it! Wifey dragged him out again. This was no bargain, you couldn't even eat it, besides it gulped electricity and, though colourful, was cheap and FLASHY and made rude STATEMENTS!

Things went quiet for a while, but the brain was still whirring away softly. It was the lull before the storm. Then suddenly one night in 1981, the front door burst open with a flourish! Carton after carton magically appeared on the lounge room floor. We were all instructed to sit and watch and say nothing. This we dutifully did, as we observed plugs and sockets being married unceremoniously. Then she was all there, Miss TI 99/4, humming pretty tunes and displaying her best colours. We were mesmerized. He pounced. She was his!

The months grew, and so did she. Expensive gifts were piled upon her. She expanded sideways. Larger and still larger tables were brought into support her wares. She then demanded the largest room in the house - the MASTER bedroom! In the meantime five good people held a clandestine meeting in a Wembley unit. They and their wares could hardly be accommodated, but they enjoyed themselves, swapping programs, never-the-less. Next time they invaded our house, but the number had tripled. Miss T.I. was now playing BACH'S "Invention in F" symphony much to everyones delight. Everyone was talking at once. Wifey offered coffee and biscuits and listened and nodded approvingly. She understood nothing.

Then they came again, wave upon wave of them. The lounge-room was re-organised, with the family now banished to the rear of the house. The mob grew - and the lounge-room shrank. It was TIME. Time to establish an official TI CLUB and seek larger premises. This was duly done and T.I.U.P. was born on the 20th of March 1982. The first TIT-BITS was typed up on a derelict portable typewriter, and the rest of the story is well known.

The 1966 issue of "Scientific American" still occupies pride of place on the library shelf with its now humorous articles and prices still intact, and Miss T.I. just sits and gloats knowingly. She WON!

This is a sad tale but true of a love affair with a mistress to beat all mistress's. This tale was told to me by Moira Elsner, and I can vouch for its validity.

DATA TRANSFER

TEXT FILE TRANSFER FROM THE 4A TO THE PROFESSIONAL

OR

PATIENCE WINS OUT

By Wayne Worladge, Melbourne.

If readers are used to seeing articles from me at all, they would be used to seeing them in "Melbourne Times" or the late-lamented "Softex" magazine. There is a reason why this article has been sent to Perth, which will become apparent later.

I've had a TI Professional personal computer for over a year now, and have spent a fair bit of time on it, and other P.C.'s. Consequently, my faithful 99/4A has not had a lot of use, except for word processing. The reason for this is that, until recently, I had not discovered a word processor significantly better than "Writer". (I am not saying they do not exist - simply that those I had for the PC did not offer very much more.)

Added to this was the fact that I had several disks full of text files which are to be used to prepare my wife's M.Ed thesis, and there was no way I was going to retype all that!

Nonetheless, I felt there may be some benefit in having the files transferred to the Pro. This feeling was reinforced by odd people here and there saying it should be a piece of cake, (HA!) and reading a Letter to the Editor in a magazine from a person who had all his text files on an Apple, and wanted to transfer them to an IBM. The Editor did, to his credit, warn the enquirer that the path may be difficult, but it could be done.

So, I finally got around to doing something about it. Software at the 4A end would be TEII, of course, and at the PRO end, I had Open Aceess, which has a communications module.

A few phone calls to "experts" and the necessary cable connections were obtained, and then I had to learn the workings of the Open Access Communications module - easier said than done, for all the examples given were for communications with a database in the USA, using Hayes Smartmodems, though the option is there to configure one's own setup. There, of course, problems arose, for while I knew the usual parameters, there were literally dozens of others to be defined, and I had not the faintest idea which were necessary, which weren't, whether the defaults would do, and so on. In the end, I set the ones I knew, and used the defaults given for a Hayes 300 modem.

Add to that the fact that I was unsure about the cable connections, and you have all the ingredients for endless frustration. Fortunately, I had corresponded with TI in the US on another matter, and decided that seeing they had been so helpful in that case, and as they had designed both computers, they ought to be able to give me the correct wiring for the cable. Accordingly, I wrote again to them. A long time had passed, and I had nearly given up. During this time, nothing I had tried had worked, when out of the blue, came a letter from TI with the required information.

Joyously, I soldered up the cable, and tried it out. Well, I did get a line of <PE><FE> and so on, which may have meant something, but TEII stopped dead after that. I nearly cried, for by his time, I had found a perfectly wonderful word processor called "Word Perfect", version 4.1, which must have been designed with thesis writers in mind.

A quick diversion here is in order, for there may be readers who are considering word processors. Word Perfect has a lot of wonderful features, but a few are:

a) The ability to enter footnotes and endnotes into the text, and to have them printed out at the bottom of the page, or end of the article, (as the case may be), automatically numbered;
b) The ability to automatically generate an index and a title page of chapter headings;
c) On-screen columns of text, changeable at will - if three columns doesn't look right, change to two!;
d) A spelling checker that takes wild-card characters and phonetic spelling;
e) A thesaurus;
f) Mathematical capability;
g) The ability to create a database, and selectively choose parameters for reports;
h) Many variations on the header/footer theme;
i) The ability to work on two files concurrently, and see them both on-screen, and be able to cut and paste between them;
j) The "flush right" key - e.g. type in the date at the head of a letter, on the left margin, press the "flush right" key, and hey presto, the date is now hard up against the right margin;
k) The above is unneccessary, for the program will automatically enter the date, in any of several formats;

and much, much more.

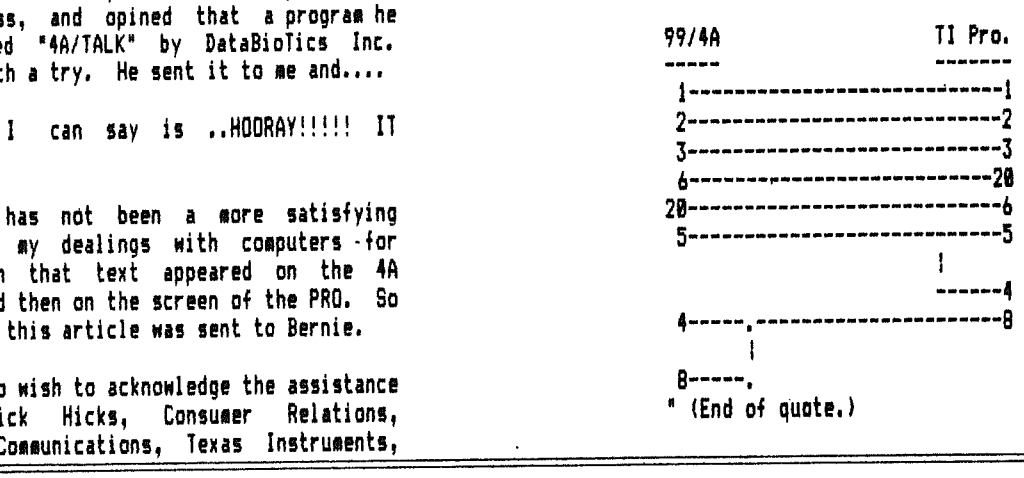
So, I was now in the situation where I really wanted to transfer those files, and in desperation, I did what I should have done ages ago, and that was to ring Bernie Elsner. Bernie, ever helpful, suggested that the protocols of TEII may not be acceptable to Open Access, and opined that a program he sells called "4A/TALK" by DataBioTics Inc. may be worth a try. He sent it to me and....

All I can say is ..HOORAY!!!!! IT WORKS!!!!!

There has not been a more satisfying moment in my dealings with computers -for years, when that text appeared on the 4A screen, and then on the screen of the PRO. So that's how this article was sent to Bernie.

I also wish to acknowledge the assistance of Patrick Hicks, Consumer Relations, Technical Communications, Texas Instruments, from whose letter I quote below:

"The following pin assignments are designed to connect a TI 99/4A Home Computer to a TI Professional Computer via a null modem cable.



" (End of quote.)

NEW

A number of new modules for the library has been purchased these are :

INTEGERS.

A self paced "tutor" which presents a variety of maths problems with both positive and negative numbers.

BURGERTIME.

You are Peter Pepper the chef. Ghastly pickles, terrifying hot dogs, and menacing eggs are invading your kitchen! Can you escape the villains in time to make your burgers?

MEASUREMENT FORMULAS.

A self paced "tutor" which presents a variety of geometric problems involving perimeter, area and volume measurement.

MATHS GAMES VI.

Five motivating maths games to help students reinforce and stregthen math skills.

ALPINER.

Climb six of the worlds tallest mountains ans evade dangerous obstacles with ALPINER. Be careful the ABOMINABLE SNOWMAN is waiting for you at the top.

EQUATIONS.

A self paced "tutor" which introduces children to the solution of mathematical equations.

LAWS OF ARITHMETIC.

A self paced "tutor" which presents mathematical principle to help your child develoop strong maths skills.

METEOR MULTIPLICATION.

An arcade game format provides fun and challenge while increasing math skills in multiplication of numbers from 0 to 9.

YAHTZEE.

Lets you play this popular game of chance while the computer rolls the dice and keeps the score for you. You concentrate on stategy.

MICROSURGEON.

Pilot the robot probe through arteries, veins and the lymphatic system. Eliminate deadly bacteria, tumors, cholestrol, tapeworms, tar deposits and other life threatening diseases as you go.

HOMEWORK HELPER.

Help set up and keep records of your assignments, etc for school. This one requires that you have at least 32k of memory and a disk based system. No operational manula is available.

```
##############################
#    DATA BASE PROGRAM BY    #
#  PHIL WEST, TI-USERS       #
#    PERTH INC.              #
#    REQUIRES 32K + DISK +   #
#    XBASIC.  PRINTER IS     #
#    OPTIONAL.               #
#    REQUIRES THE ASSEMBLY   #
#    LANGUAGE ROUTINES       #
#    PRINTED ELSEWHERE IN    #
#    THIS NEWSLETTER.        #
#    IF ENTERING THIS        #
#    PROGRAM SEEMS A         #
#    DAUNTING TASK, WE HAVE  #
#    MADE IT EASY FOR YOU.   #
#    JUST SEND $10.00 AUST   #
#    FOR THE TWO SINGLE      #
#    SIDED DISKS THAT HAVE   #
#    THIS PROGRAM AND THE    #
#    USERS MANUAL TO THE     #
#    FOLLOWING ADDRESS :     #
#                            #
#    DISK LIBRARIAN          #
#    TI-USERS PERTH INC      #
#    P.O. BOX 246            #
#    MT LAWLEY               #
#    WESTERN AUSTRALIA       #
#    6050                    #
#                            #
#    THIS SMALL CHARGE       #
#    INCLUDES THE COST OF    #
#    THE 2 DISKS, POSTAGE    #
#    AND PACKAGING, VIA      #
#    AIRMAIL.                #
#    ALSO INCLUDED ARE THE   #
#    ASSEMBLY LANGUAGE       #
#    SOURCE CODE AND TWO     #
#    ASSOCIATED PROGRAMS.    #
##############################
```

```
100 ! INITIALIZATION
110 COM$="FR.CR.ER.WR.LP.NR.
PR.QU.DR.IC.PC.??.CF.GR.DP.L
N.LH.LT.EX.RR.GD.SD.CD.IF.CI
.RP.JD.AS.DS.FS."
120 DECIMAL$="0123456789" ::
 ORDER$="NRTSPHBE"
130 FLG,TOT,QUI,LBFLG,GOTDAT
,AS,DS=0 :: INK=16 :: PAPER=
13 :: LABPTR=1 :: FS=60
140 DEF \=ABS(LEN(COMMAND$)>
3)*POS(COMMAND$,"/",1)
150 DEF @$=SEG$(COMMAND$,\+1
,LEN(COMMAND$)-\)
160 CALL INIT :: CALL LOAD(8
196,63,248):: CALL LOAD(-318
36,16)
170 CALL LOAD(16376,68,32,32
,32,32,32,37,0)
180 CALL LOAD(9472,2,224,47,
254,2,0,62,235,4,32,32,40,4,
254,131,124,2,224,131,224,4,
9)
190 CALL LINK("D"):: CALL PE
EK(12288,DRIVE):: DRIVE$="DS
K"&CHR$(DRIVE+48)&".DB"
200 DATA "NAME",25,"ROAD",25
,"TOWN",25,"STA",1,"POST",1,
"HOME PHON",1,"BUSN PHON",1,
"EXTN",1
210 DATA 1,28,29,28,57,28,85
,3,88,4,92,9,101,9,110,4
220 DIM FLAG$(14),FLGLAB$(14
),FOUND(90),LABEL$(3),LTAB(3
),FLGTST$(14),SEGPOS(8),SEGL
EN(8),HEAD$(8),HEDLEN(8)
230 GOTO 240 :: CALL INIT ::
 CALL LOAD :: CALL PEEK :: C
ALL CLEAR :: CALL KEY :: CAL
L SCREEN :: CALL COLOR :: CA
LL PEN :: CALL CHAR
240 GOTO 250 :: CALL NUMERD
:: CALL CHARPAT :: COM$ :: D
ECIMAL$ :: ORDER$ :: FLG ::
TOT :: QUI :: LBFLG :: GOTDA
T :: AS :: DS :: COL$
250 GOTO 260 :: INK :: PAPER
 :: LABPTR :: FS :: COMMAND$
 :: DRIVE$ :: A$ :: I :: FIL
E :: KEYWORD$ :: COMMAND ::
X :: NAME$ :: VALID :: S$
260 GOTO 270 :: SORT$ :: SEE
K$ :: NREC :: GET :: PASS ::
 COUNT :: LISTPTR :: COL ::
NL :: HL :: LT :: TS$ :: FIL
E$ :: PRINTER$ :: PRV$
270 GOTO 280 :: PWID :: CH :
: PRV :: REPT$ :: Z$ :: N$ :
: LIN :: CURRENT :: NA$ :: N
M$ :: ST$ :: TW$ :: AT$ :: P
C$ :: HP$ :: BP$ :: EX$
280 GOTO 290 :: ED :: K :: S
 :: J :: ON$ :: TABTAB :: CO
MMA :: JOIN$ :: !@P-
290 CALL CLEAR :: CALL SCREE
N(PAPER):: DISPLAY AT(24,1):
"initializing" :: CALL PEN(I
NK)
300 FOR I=1 TO 8 :: READ HEA
D$(I),HEDLEN(I):: NEXT I
310 FOR I=1 TO 8 :: READ SEG
POS(I),SEGLEN(I):: NEXT I
320 CALL CHAR(96,""):: CALL
CHARPAT(63,A$):: CALL CHAR(1
23,A$)
330 FOR I=48 TO 57 :: CALL C
HARPAT(I,A$):: CALL CHAR(I+7
6,A$):: NEXT I
340 FOR I=97 TO 122 :: CALL
CHARPAT(I,A$):: CALL CHAR(I,
SEG$(A$,3,14)):: NEXT I
350 CALL LOAD(DRIVE$&"CODE")
:: GOSUB 3090
360 ! COMMAND INTERPRETER
370 OPEN #5:DRIVE$&"XAUTOEXE
" :: IF EOF(5)THEN CLOSE #5:
DELETE ELSE FILE=1
380 IF FILE=0 THEN 410
390 IF EOF(5)THEN CLOSE #5 :
: FILE=0 :: GOTO 410
400 LINPUT #5:COMMAND$ :: DI
SPLAY AT(22,2):SEG$(COMMAND$
,1,27):: GOTO 420
410 ACCEPT AT(22,2)VALIDATE(
UALPHA,DIGIT,".=,/?"):COMMAN
D$ :: IF COMMAND$="" THEN 41
0
420 IF LEN(COMMAND$)<2 THEN
GOSUB 3330 :: IF FILE=1 THEN
 380 ELSE 410
430 DISPLAY AT(24,1):"" :: K
EYWORD$=SEG$(COMMAND$,1,2)
440 COMMAND=POS(COM$,KEYWORD
$,1):: IF COMMAND=0 THEN GOS
UB 3330 :: IF FILE=1 THEN 38
0 ELSE 410
450 COMMAND=INT(COMMAND/3)+1
 :: IF FILE=0 THEN 470
460 X=COMMAND :: IF X=2 OR X
=3 OR X=4 OR X=9 OR X=12 OR
X=13 OR X=19 OR X=20 THEN DI
SPLAY AT(24,1):"not`allowed`
in`command`file" :: GOTO 380
470 IF GOTDAT=1 THEN 500
480 X=COMMAND :: IF X=8 OR X
=10 OR X=11 OR X=12 OR X=15
OR X=16 OR X=17 OR X=18 OR X
=19 OR X=21 OR X=26 THEN 500
490 DISPLAY AT(24,1):"data`f
ile`not`loaded" :: IF FILE=1
 THEN 360 ELSE 410
500 IF COMMAND>21 THEN 530
510 ON COMMAND GOSUB 1140,15
70,1590,1790,1880,2050,2110,
2170,2220,2260,2300,2340,245
0,2560,2630,940,980,1020,120
0,910,570
520 GOTO 540
530 ON COMMAND-21 GOSUB 670,
790,1070,1120,1240,2870,2970
,3010,3050
540 IF FILE=1 AND QUI=0 THEN
 380
550 IF QUI=0 THEN 410 ELSE C
ALL CLEAR :: STOP
560 ! GD COMMAND
570 IF \ THEN NAME$=SEG$(@$,
1,7)ELSE 3350
580 IF GOTDAT=1 THEN CLOSE #
2
590 VALID,TOT,FLG=0 :: DISPL
AY AT(24,1):"loading`data`fi
le"
600 OPEN #1:DRIVE$&"F"&NAME$
 :: OPEN #2:DRIVE$&"D"&NAME$
,RELATIVE,FIXED 128 :: GOTDA
T=1 :: CALL LINK("NEW")
610 IF EOF(1)THEN 630 ELSE I
NPUT #1:FLGLAB$(FLG)
620 FLG=FLG+1 :: IF FLG<15 T
HEN 610
630 FLG=FLG-1 :: CLOSE #1
640 IF EOF(2)THEN 650 ELSE L
INPUT #2:A$ :: CALL LINK("AD
D",A$,TOT):: TOT=TOT+1 :: GO
TO 640
650 CALL NUMERD(TOT,S$):: DI
SPLAY AT(24,1):S$;"`records`
loaded" :: GOSUB 3290 :: RET
URN
660 ! SD COMMAND
670 IF \ THEN SORT$=SEG$(@$,
1,7)ELSE 3350
```

```
680 IF SORT$=NAME$ THEN 3400
690 VALID=0 :: SEEK$="" :: D
ISPLAY AT(24,1):"sorting`dat
a`file" :: CALL LINK("SORT")
:: NREC=0
700 OPEN #1:DRIVE$&"F"&NAME$
 :: OPEN #7:DRIVE$&"F"&SORT$
710 IF EOF(1)THEN CLOSE #1 :
: CLOSE #7 :: GOTO 730
720 INPUT #1:A$ :: PRINT #7:
A$ :: GOTO 710
730 OPEN #7:DRIVE$&"D"&SORT$
,FIXED 128
740 CALL LINK("GET",GET):: I
F GET=-1 THEN 770
750 LINPUT #2,REC GET:A$ ::
GOSUB 2660 :: IF PASS=0 THEN
 740
760 NREC=NREC+1 :: PRINT #7:
A$ :: GOTO 740
770 CLOSE #7 :: CALL NUMERO(
NREC,S$):: DISPLAY AT(24,1):
S$;"`records`sorted" :: GOSU
B 3290 :: RETURN
780 ! CD COMMAND
790 IF \ THEN SORT$=SEG$(@$,
1,7)ELSE 3350
800 IF SORT$=NAME$ THEN 3400
810 VALID=0 :: SEEK$="" :: D
ISPLAY AT(24,1):"copying`dat
a`file"
820 OPEN #1:DRIVE$&"F"&NAME$
 :: OPEN #7:DRIVE$&"F"&SORT$
830 IF EOF(1)THEN CLOSE #1 :
: CLOSE #7 :: GOTO 850
840 INPUT #1:A$ :: PRINT #7:
A$ :: GOTO 830
850 OPEN #7:DRIVE$&"D"&SORT$
,FIXED 128 :: RESTORE #2 ::
NREC=0
860 FOR I=0 TO TOT-1 :: LINP
UT #2:A$ :: GOSUB 2660 :: IF
 PASS=0 THEN 880
870 NREC=NREC+1 :: PRINT #7:
A$
880 NEXT I
890 CLOSE #7 :: CALL NUMERO(
NREC,S$):: DISPLAY AT(24,1):
S$;"`records`copied" :: RETU
RN
900 ! RR COMMAND
910 CALL LINK("FIND"," ",FOU
ND(),COUNT):: IF COUNT=-1 TH
EN DISPLAY AT(24,1):"no`reco
rds`found" :: GOSUB 3290 ::
RETURN
920 VALID=1 :: LISTPTR=0 ::
GOSUB 3200 :: RETURN
930 ! LN COMMAND
940 IF \ THEN GOSUB 3140 ELS
E 3350
950 IF COL<1 OR COL>3 THEN 3
390
960 NL=COL :: RETURN
970 ! LH COMMAND
980 IF \ THEN GOSUB 3140 ELS
E 3350
990 IF COL<4 OR COL>12 THEN
3390
1000 HL=COL :: RETURN
1010 ! LT COMMAND
1020 IF \ THEN GOSUB 3140 EL
SE 3350
1030 IF COL<1 OR COL>104 THE
N 3390
1040 IF POS(SEG$(DECIMAL$,2,
3),SEG$(COMMAND$,\-1,1),1)=0
 THEN 3350
1050 LT=ASC(SEG$(COMMAND$,\-
1,1))-48 :: LTAB(LT)=COL ::
RETURN
1060 ! IF COMMAND
1070 IF \ THEN GOSUB 3140 EL
SE 3350
1080 IF COL=-1 OR COL>FLG+1
THEN 3390
1090 TS$=SEG$(COMMAND$,\-1,1
):: IF POS("YN",TS$,1)=0 THE
N 3390
1100 FLGTST$(COL-1)=TS$ :: R
ETURN
1110 ! CI COMMAND
1120 FOR I=0 TO 14 :: FLGTST
$(I)="" :: NEXT I :: RETURN
1130 ! FR COMMAND
1140 IF \ THEN SEEK$=@$ ELSE
 3350
1150 CALL LINK("FIND",SEEK$,
FOUND(),COUNT)
1160 IF COUNT=-1 THEN VALID=
0 :: GOSUB 3290 :: DISPLAY A
T(24,1):"no`records`found" :
: RETURN
1170 VALID=1 :: LISTPTR=0 ::
 CALL NUMERO(COUNT+1,N$)
1180 DISPLAY AT(24,1):N$;"`r
ecords`found" :: GOSUB 3200
:: RETURN
1190 ! EX COMMAND
1200 IF \ THEN FILE$=SEG$(@$
,1,7)ELSE 3350
1210 OPEN #5:DRIVE$&"X"&FILE
$ :: IF EOF(5)THEN CLOSE #5:
DELETE :: GOTO 3400
1220 FILE=1 :: RETURN
1230 ! RP COMMAND
1240 IF PRINTER$="" THEN 336
0
1250 IF \ THEN REPT$=@$ ELSE
 3350
1260 SEEK$="" :: GOSUB 3290
:: DISPLAY AT(24,1):"printin
g`report" :: PRV$="" :: PRV=
0
1270 OPEN #1:PRINTER$,VARIAB
LE 132
1280 PWID=0 :: FOR I=1 TO 8
:: CH=POS(REPT$,SEG$(ORDER$,
I,1),1):: IF CH=0 THEN 1300
1290 PWID=PWID+SEGLEN(I)+1
1300 NEXT I
1310 IF PWID>80 THEN PRINT #
1:CHR$(15)
1320 GOSUB 1520 :: CALL LINK
("SORT")
1330 CALL LINK("GET",GET)::
IF GET=-1 THEN 1490
1340 LINPUT #2,REC GET:A$ ::
 GOSUB 2660 :: IF PASS=0 OR
ASC(A$)=32 THEN 1330
1350 IF AS=0 THEN 1400
1360 IF PRV=0 THEN PRV=1 ::
GOTO 1400
1370 IF SEG$(A$,1,1)=PRV$ TH
EN 1400
1380 PRV$=SEG$(A$,1,1):: PRI
NT #1 :: LIN=LIN+1
1390 IF DS THEN PRINT #1 ::
LIN=LIN+1
1400 FOR I=1 TO 8 :: IF POS(
REPT$,SEG$(ORDER$,I,1),1)=0
THEN 1440
1410 IF I<>1 THEN 1430
1420 Z$=A$ :: GOSUB 2700 ::
PRINT #1:N$;" ";:: GOTO 1440
1430 PRINT #1:SEG$(A$,SEGPOS
(I),SEGLEN(I));" ";
1440 NEXT I :: PRINT #1 :: L
IN=LIN+1
1450 IF DS=0 THEN 1470
1460 PRINT #1 :: LIN=LIN+1
1470 IF LIN>FS THEN GOSUB 15
20
1480 GOTO 1330
1490 IF PWID>80 THEN PRINT #
1:CHR$(18)
1500 CLOSE #1 :: RETURN
1510 ! PAGE HEADER ROUTINE
1520 PRINT #1:CHR$(12)
1530 FOR I=1 TO 8 :: IF POS(
REPT$,SEG$(ORDER$,I,1),1)=0
THEN 1550
1540 PRINT #1:HEAD$(I);RPT$(
" ",HEDLEN(I));
1550 NEXT I :: PRINT #1 :: P
RINT #1 :: LIN=2 :: RETURN
1560 ! CR COMMAND
1570 GOSUB 3290 :: CURRENT=-
1 :: SEEK$="" :: VALID=1
1580 ! ER COMMAND
1590 IF VALID=0 THEN 3380
1600 ACCEPT AT(2,1)SIZE(-28)
VALIDATE(UALPHA,DIGIT," ,/'"
):NA$ :: IF NA$="" THEN VALI
D=0 :: RETURN
1610 FOR I=1 TO LEN(NA$):: I
F SEG$(NA$,I,1)<>" " THEN 16
30
1620 NEXT I
1630 NM$=SEG$(NA$,I,LEN(NA$)
-I+1):: DISPLAY AT(2,1):NM$
1640 ACCEPT AT(4,1)SIZE(-28)
VALIDATE(UALPHA,DIGIT," ,'/"
):ST$ :: CALL LINK("UP",PASS
):: IF PASS THEN 1600
1650 ACCEPT AT(6,1)SIZE(-28)
VALIDATE(UALPHA,DIGIT," ,/'"
):TW$ :: CALL LINK("UP",PASS
):: IF PASS THEN 1640
1660 ACCEPT AT(8,1)SIZE(-3)V
ALIDATE(UALPHA):AT$ :: CALL
LINK("UP",PASS):: IF PASS TH
EN 1650
1670 ACCEPT AT(10,1)SIZE(-4)
```

```
VALIDATE(DIGIT):PC$ :: CALL
LINK("UP",PASS):: IF PASS TH
EN 1660
1680 ACCEPT AT(12,1)SIZE(-9)
VALIDATE(DIGIT):HP$ :: CALL
LINK("UP",PASS):: IF PASS TH
EN 1670
1690 ACCEPT AT(14,1)SIZE(-9)
VALIDATE(DIGIT):BP$ :: CALL
LINK("UP",PASS):: IF PASS TH
EN 1680
1700 ACCEPT AT(16,1)SIZE(-4)
VALIDATE(DIGIT):EX$ :: CALL
LINK("UP",PASS):: IF PASS TH
EN 1690
1710 FOR I=0 TO FLG
1720 DISPLAY AT(17,1):FLGLAB
$(I):: DISPLAY AT(17,LEN(FLG
LAB$(I))+2):FLAG$(I)
1730 ACCEPT AT(17,LEN(FLGLAB
$(I))+2)SIZE(-1)VALIDATE("YN
"):FLAG$(I):: IF FLAG$(I)=""
 THEN 1730
1740 CALL LINK("UP",PASS)::
IF PASS AND I=0 THEN 1700
1750 IF PASS THEN I=I-1 :: G
OTO 1720
1760 NEXT I :: ED=1
1770 DISPLAY AT(17,1):FLGLAB
$(0):: DISPLAY AT(17,LEN(FLG
LAB$(0))+2):FLAG$(0):: RETUR
N
1780 ! WR COMMAND
1790 IF VALID=0 OR ED=0 THEN
 3380 ELSE DISPLAY AT(24,1):
"writing`data`to`file"
1800 IF CURRENT<>-1 THEN 182
0
1810 CALL LINK("FIND"," ",FO
UND(),COUNT):: IF COUNT=-1 T
HEN CURRENT=TOT ELSE CURRENT
=FOUND(0)
1820 IF CURRENT=0 THEN RESTO
RE #2 ELSE LINPUT #2,REC CUR
RENT-1:A$
1830 PRINT #2:NM$;TAB(29);ST
$;TAB(57);TN$;TAB(85);AT$;TA
B(88);PC$;TAB(92);HP$;TAB(10
1);BP$;TAB(110);EX$;
1840 FOR I=0 TO 13 :: PRINT
#2:TAB(114+I);FLAG$(I);:: NE
XT I :: PRINT #2:TAB(128);FL
AG$(14)
1850 IF CURRENT=TOT THEN TOT
=TOT+1 :: CALL LINK("ADD",NM
$,CURRENT)ELSE CALL LINK("MO
D",NM$,CURRENT)
1860 GOSUB 3290 :: RETURN
1870 ! LP COMMAND
1880 IF PRINTER$="" THEN 336
0
1890 IF NL=0 OR HL=0 THEN 33
70
1900 IF LTAB(1)=0 THEN LTAB(
1)=1
1910 ON NL GOTO 1940,1930,19
20
1920 IF LTAB(3)=0 THEN 3370
1930 IF LTAB(2)=0 THEN 3370
1940 IF \ THEN 1980
1950 IF VALID=0 THEN 3380
1960 LABEL$(LABPTR)=A$ :: LA
BPTR=LABPTR+1 :: LBFLG=1 ::
IF LABPTR<=NL THEN RETURN
1970 GOSUB 2760 :: RETURN
1980 IF @$<>"ALL" THEN 3380
1990 CALL LINK("SORT"):: NRE
C=0 :: GOSUB 3290
2000 CALL LINK("GET",GET)::
IF GET=-1 THEN 2030
2010 LINPUT #2,REC GET:A$ ::
 GOSUB 2660 :: IF PASS=0 THE
N 2000
2020 NREC=NREC+1 :: GOSUB 19
60 :: GOTO 2000
2030 CALL NUMERO(NREC,S$)::
DISPLAY AT(24,1):S$;"`labels
`printed" :: RETURN
2040 ! NR COMMAND
2050 IF VALID=0 THEN 3380
2060 IF SEEK$="" AND CURRENT
<TOT-1 THEN CURRENT=CURRENT+
1 :: GOTO 3210
2070 IF SEEK$="" THEN RETURN
2080 IF LISTPTR=COUNT THEN L
ISTPTR=-1
2090 LISTPTR=LISTPTR+1 :: GO
SUB 3200 :: RETURN
2100 ! PR COMMAND
2110 IF VALID=0 THEN 3380
2120 IF SEEK$="" AND CURRENT
>0 THEN CURRENT=CURRENT-1 ::
 GOTO 3210
2130 IF SEEK$="" THEN RETURN
2140 IF LISTPTR=0 THEN LISTP
TR=COUNT+1
2150 LISTPTR=LISTPTR-1 :: GO
SUB 3200 :: RETURN
2160 ! QU COMMAND
2170 QUI=1 :: IF GOTDAT=1 TH
EN CLOSE #2
2180 IF FILE=1 THEN CLOSE #5
2190 IF LBFLG=0 THEN RETURN
2200 NL=LABPTR-1 :: GOSUB 27
60 :: RETURN
2210 ! DR COMMAND
2220 IF VALID=0 THEN RETURN
2230 IF CURRENT=0 THEN RESTO
RE #2 ELSE LINPUT #2,REC CUR
RENT-1:Z$
2240 PRINT #2:" ";SEG$(A$,2,
127):: CALL LINK("MOD"," ",C
URRENT):: GOSUB 3290 :: RETU
RN
2250 ! IC COMMAND
2260 IF \ THEN GOSUB 3140 EL
SE 3350
2270 IF COL<2 OR COL>16 OR C
OL=PAPER THEN 3390
2280 INK=COL :: CALL PEN(INK
):: RETURN
2290 ! PC COMMAND
2300 IF \ THEN GOSUB 3140 EL
SE 3350
2310 IF COL<2 OR COL>16 OR C
OL=INK THEN 3390
2320 PAPER=COL :: CALL SCREE
N(PAPER):: RETURN
2330 ! ?? COMMAND
2340 OPEN #3:DRIVE$&"HELP"
2350 IF EOF(3)THEN CLOSE #3
:: DISPLAY AT(24,1):"help`fi
le`not`available" :: RETURN
2360 CALL CLEAR
2370 FOR I=1 TO 22 :: IF EOF
(3)THEN 2400
2380 LINPUT #3:A$ :: DISPLAY
 AT(I,1):A$
2390 NEXT I
2400 DISPLAY AT(24,1):"press
`any`key`to`continue"
2410 CALL KEY(0,K,S):: IF S=
0 THEN 2410
2420 IF EOF(3)=0 THEN 2360
2430 CLOSE #3 :: GOSUB 3090
:: RETURN
2440 ! CF COMMAND
2450 IF \ THEN FLG$=@$ ELSE
3350
2460 IF FLG=14 THEN 3340
2470 FLG=FLG+1
2480 FLGLAB$(FLG)="" :: FOR
I=1 TO LEN(FLG$):: CH=ASC(SE
G$(FLG$,I,1))
2490 IF CH=32 THEN CH=96
2500 IF CH>47 AND CH<58 THEN
 CH=CH+76
2510 IF CH>64 AND CH<91 THEN
 CH=CH+32
2520 FLGLAB$(FLG)=FLGLAB$(FL
G)&CHR$(CH):: NEXT I
2530 OPEN #1:DRIVE$&"F"&NAME
$,APPEND
2540 PRINT #1:FLGLAB$(FLG)::
 CLOSE #1 :: RETURN
2550 ! GR COMMAND
2560 IF \ THEN GOSUB 3140 EL
SE 3350
2570 SEEK$=""
2580 IF COL=-1 OR COL>TOT-1
THEN GOSUB 3290 :: GOTO 3390
2590 CURRENT=COL :: GOSUB 32
10
2600 IF SEG$(A$,1,1)=" " THE
N GOSUB 3290 :: GOTO 3390
2610 VALID=1 :: SEEK$="" ::
RETURN
2620 ! DP COMMAND
2630 IF \ THEN PRINTER$=@$ E
LSE 3350
2640 RETURN
2650 ! FLAG TEST ROUTINE
2660 PASS=1 :: FOR J=0 TO FL
G :: IF FLGTST$(J)="" THEN 2
680
2670 IF SEG$(A$,114+J,1)=FLG
TST$(J)THEN PASS=1 ELSE PASS
=0
2680 NEXT J :: RETURN
2690 ! NAME REVERSAL ROUTINE
2700 IF POS(SEG$(Z$,1,28),",
",1)=0 THEN N$=SEG$(Z$,1,28)
:: RETURN
2710 COMMA=POS(SEG$(Z$,1,28)
```

```
,",",1):: N$=SEG$(Z$,COMMA+1
,28-COMMA)
2720 FOR J=LEN(N$)TO 1 STEP
-1 :: IF SEG$(N$,J,1)<>" " T
HEN 2740
2730 NEXT J
2740 N$=SEG$(N$,1,J)&" "&SEG
$(Z$,1,COMMA-1):: N$=N$&RPT$
(" ",28-LEN(N$)):: RETURN
2750 ! LABEL PRINT ROUTINE
2760 OPEN #1:PRINTER$,VARIAB
LE 132
2770 FOR I=1 TO NL :: Z$=LAB
EL$(I):: GOSUB 2700
2780 TABTAB=LTAB(I):: PRINT
#1:TAB(TABTAB);N$;:: NEXT I
:: PRINT #1
2790 FOR I=1 TO NL :: TABTAB
=LTAB(I)
2800 PRINT #1:TAB(TABTAB);SE
G$(LABEL$(I),29,28);:: NEXT
I :: PRINT #1
2810 FOR I=1 TO NL :: TABTAB
=LTAB(I)
2820 PRINT #1:TAB(TABTAB);SE
G$(LABEL$(I),57,28);:: NEXT
I :: PRINT #1
2830 FOR I=1 TO NL :: TABTAB
=LTAB(I)
2840 PRINT #1:TAB(TABTAB);SE
G$(LABEL$(I),85,3);TAB(TABTA
B+5);SEG$(LABEL$(I),88,4);::
 NEXT I :: PRINT #1
2850 FOR I=5 TO HL :: PRINT
#1 :: NEXT I :: CLOSE #1 ::
LABPTR=1 :: LBFLG=0 :: RETUR
N
2860 ! JD COMMAND
2870 IF \ THEN JOIN$=SEG$(@$
,1,7)ELSE 3350
2880 IF JOIN$=NAME$ THEN 340
0
2890 LINPUT #2,REC TOT-1:A$
2900 OPEN #7:DRIVE$&"D"&JOIN
$,FIXED 128 :: IF EOF(7)THEN
 CLOSE #7:DELETE :: GOTO 339
0
2910 DISPLAY AT(24,1):"joini
ng`records`to`data`file" ::
NREC=0 :: SEEK$="" :: GOSUB
3290
2920 IF EOF(7)THEN 2950
2930 LINPUT #7:A$ :: GOSUB 2
660 :: IF PASS=0 THEN 2920
2940 PRINT #2:A$ :: CALL LIN
K("ADD",A$,TOT):: NREC=NREC+
1 :: TOT=TOT+1 :: GOTO 2920
2950 CLOSE #7 :: CALL NUMERO
(NREC,S$))DISPLAY AT(24,1):S
$;"`records`joined`to`file"
:: RETURN
2960 ! AS COMMAND
2970 IF \ THEN ON$=@$ ELSE 3
350
2980 IF ON$="ON" THEN AS=1 E
LSE AS=0
2990 RETURN
3000 ! DS COMMAND
3010 IF \ THEN ON$=@$ ELSE 3
350
3020 IF ON$="ON" THEN DS=1 E
LSE DS=0
3030 RETURN
3040 ! FS COMMAND
3050 IF \ THEN GOSUB 3140 EL
SE 3350
3060 IF COL<10 OR COL>100 TH
EN FS=60 :: GOTO 3390
3070 FS=COL :: RETURN
3080 ! PROMPTS ROUTINE
3090 DISPLAY AT(1,1)ERASE AL
L:"name" :: DISPLAY AT(3,1):
"road" :: DISPLAY AT(5,1):"t
own" :: DISPLAY AT(7,1):"sta
te"
3100 DISPLAY AT(9,1):"postco
de" :: DISPLAY AT(11,1):"hom
e`phone" :: DISPLAY AT(13,1)
:"busn`phone" :: DISPLAY AT(
15,1):"extn"
3110 DISPLAY AT(15,18):"reco
rd" :: DISPLAY AT(17,1):FLGL
AB$(0):: DISPLAY AT(22,1):">
" :: DISPLAY AT(19,1):"type`
((`for`help"
3120 VALID=0 :: RETURN
3130 ! VAL() ROUTINE
3140 COL=-1 :: COL$=@$
3150 FOR I=1 TO LEN(COL$)
3160 IF POS(DECIMAL$,SEG$(CO
L$,I,1),1)=0 THEN RETURN
3170 NEXT I
3180 COL=VAL(COL$):: RETURN
3190 ! DISPLAY REC ROUTINE
3200 CURRENT=FOUND(LISTPTR)
3210 LINPUT #2,REC CURRENT:A
$
3220 DISPLAY AT(2,1):SEG$(A$
,1,28):: DISPLAY AT(4,1):SEG
$(A$,29,28):: DISPLAY AT(6,1
):SEG$(A$,57,28)
3230 DISPLAY AT(8,1):SEG$(A$
,85,3):: DISPLAY AT(10,1):SE
G$(A$,88,4):: DISPLAY AT(12,
1):SEG$(A$,92,9)
3240 DISPLAY AT(14,1):SEG$(A
$,101,9):: DISPLAY AT(16,1):
SEG$(A$,110,4)
3250 CALL NUMERO(CURRENT,N$)
:: DISPLAY AT(15,25):N$
3260 FOR I=0 TO 14 :: FLAG$(
I)=SEG$(A$,I+114,1):: NEXT I
3270 DISPLAY AT(17,1):FLGLAB
$(0):: DISPLAY AT(17,LEN(FLG
LAB$(0))+2):FLAG$(0):: ED=0
:: RETURN
3280 ! CLEAR FIELDS ROUTINE
3290 DISPLAY AT(2,1):"" :: D
ISPLAY AT(4,1):"" :: DISPLAY
 AT(6,1):"" :: DISPLAY AT(8,
1):""
3300 DISPLAY AT(10,1):"" ::
DISPLAY AT(12,1):"" :: DISPL
AY AT(14,1):"" :: DISPLAY AT
(16,1):"" :: DISPLAY AT(15,2
5):""
3310 DISPLAY AT(17,1):FLGLAB
$(0):: VALID=0 :: ED=0 :: RE
TURN
3320 ! DISPLAY ERRORS
3330 DISPLAY AT(24,1):"unkwo
wn`command" :: RETURN
3340 DISPLAY AT(24,1):"all`f
lags`defined" :: RETURN
3350 DISPLAY AT(24,1):"inval
id`format" :: RETURN
3360 DISPLAY AT(24,1):"print
er`undefined" :: RETURN
3370 DISPLAY AT(24,1):"label
`dimensions`undefined" :: RE
TURN
3380 DISPLAY AT(24,1):"inval
id`command" :: RETURN
3390 DISPLAY AT(24,1):"bad`v
alue" :: RETURN
3400 DISPLAY AT(24,1):"bad`f
ilename" :: RETURN
3410 ! CONVERT NUM ROUTINE
3420 !@P+
3430 SUB NUMERO(NX,ST$):: ST
$="" :: S$=STR$(NX)
3440 FOR I=1 TO LEN(S$):: CH
=ASC(SEG$(S$,I,1))+76 :: ST$
=ST$&CHR$(CH):: NEXT I :: SU
BEND
3450 ! INK COLOUR ROUTINE
3460 SUB PEN(IK):: FOR I=0 T
O 8 :: CALL COLOR(I,IK,1)::
NEXT I
3470 FOR I=9 TO 13 :: CALL C
OLOR(I,1,IK):: NEXT I :: SUB
END
```

```
############################
#                          #
# Conversion routine to    #
# change any lower case    #
# character into it's      #
# upper case equivalent.   #
#                          #
# Can be stored in the     #
# MERGE format and used    #
# as a subroutine.         #
#                          #
############################
```

```
100 ! CONVERSION ROUTINE
110 NM$=""
120 FOR I=1 TO LEN(A$)
130 CH$=SEG$(A$,I,1)
140 CH=ASC(CH$)
150 IF (CH<123)AND(CH>96)THE
N 160 ELSE 170
160 CH=CH-32
170 NM$=NM$&CHR$(CH)
180 NEXT I
190 PRINT NM$
200 A$=""
210 RETURN
```

```
############################
#                          #
#  See the articles FUEL   #
#  TANK EXPLAINED, for     #
#  details on how this     #
#  program works.          #
#                          #
############################
```

```
100 !*********************
110 !*                   *
120 !*   STEVE WILKINSON *
130 !*       T.I.U.P.    *
140 !*VOLUME CALCULATION OF*
150 !*FUEL TANK ( cylinder *
160 !*horizontal )        *
170 !* NOTE Result of SIN  *
180 !* & COS is in Radians *
190 !*                   *
200 !*********************
210 DIM Z(3)
220 CALL CLEAR
230 DATA "INPUT RADIUS","INP
UT LENGTH","INPUT DEPTH"
240 FOR I=1 TO 3
250 READ IN$
260 !
270 ! INPUT DATA
280 !
290 DISPLAY AT(24,1)SIZE(14)
:IN$ :: DISPLAY AT(24,19)SIZ
E(2):"##"
300 ACCEPT AT(24,14)SIZE(5):
INPT(I)
310 !
320 ! CHECK FOR INPUT ERROR
330 !
340 IF INPT(3)>2*INPT(1)THEN
 300
350 IF INPT(I)<=0 THEN 300
360 NEXT I
370 RADIUS=INPT(1):: LENGTH=
INPT(2):: DEPTH=INPT(3)
380 !
390 ! AREA OF FULL CIRCLE
400 !
410 AREACIRCLE=PI*(RADIUS^2)
420 IF DEPTH=RADIUS THEN 720
430 X=RADIUS-DEPTH
440 !
450 ! PYTHAGORAS TO THE RESC
UE
460 !
470 EB=SQR((RADIUS^2)-(X^2))
480 !
490 ! FIND ANGLE FROM CENTRE
 TO ENDS OF CORD
500 !
510 SINANGLE=SIN(EB/RADIUS)
520 COSANGLE=COS(RADIUS/X)
530 !
540 ! Answer in RADIANS
550 !
560 ANGLE=ATN(EB/X)
570 IF DEPTH<RADIUS THEN 590
 ELSE 640
580 !
590 ! AREA OF MINOR SEGMENT=
AREA OF MINOR SECTOR - TRIAN
GLE OF CENTRE & CORD
600 !
610 AREA=.5*RADIUS^2*(2*ANGL
E-SIN(2*ANGLE))
620 GOTO 760
630 !
640 ! AREA OF MAJOR SEGMENT=
AREA OF MAJOR SECTOR + TRIAN
GLE OF CENTRE & CORD
650 !
660 ANGLE=ANGLE*-1
670 AREA=.5*RADIUS^2*(2*PI-(
2*ANGLE))+.5*RADIUS^2*SIN(2*
ANGLE)
680 GOTO 760
690 !
700 ! SPECIAL CASE HALF FULL
710 !
720 AREA=.5*AREACIRCLE
730 !
740 ! CALCULATE VOLUME IN LI
TRES
750 !
760 VOLLITRE=AREA*LENGTH*10^
-6
770 !
780 ! CALCULATE VOLUME IN GA
LLONS
790 !
800 VOLGALS=VOLLITRE*.219969
810 VOLLITRE=(INT(VOLLITRE*1
00))/100
820 VOLGALS=(INT(VOLGALS*100
))/100
830 PRINT VOLLITRE;" LITRES"
840 PRINT VOLGALS;" GALLONS"
850 RESTORE
860 PRINT :: PRINT :: PRINT
:: PRINT ;"PRESS ANY KEY TO
CONTINUE"
870 CALL KEY(0,K,S)
880 IF S=0 THEN 870
890 GOTO 220
```

```
############################
#                          #
#  GEMINI 10X Character    #
#  set constructor.        #
#                          #
#  With some alteration    #
#  this could be used to   #
#  define a downloadable   #
#  character set for most  #
#  types of printers.      #
#                          #
#  See details on how to   #
#  use this program, in    #
#  the article :           #
#  10X CHARACTER SET       #
#                          #
############################
```

```
100 DATA 1,2,4,8,16,32,64
110 REM ******************
120 REM *                *
130 REM *    GEMINI 10X  *
140 REM *  - CHARACTER   *
150 REM *    DOWNLOADER  *
160 REM *                *
170 REM *     By Steve   *
180 REM *     Wilkinson  *
190 REM *    TIUP PERTH  *
200 REM *                *
210 REM *                *
220 REM ******************
230 CALL INIT :: CALL LOAD(-
31806,16,0)!
   DISABLE QUIT
240 ANYK$="Press any key to
continue"
250 DIM TX(9),TX$(9),SC$(3)
260 DSK=1
270 REM
    MAIN MENU
280 CALL CLEAR
290 DISPLAY AT(2,4):"CHARACT
ER CUSTOMISER" :: DISPLAY AT
(3,4):"--------------------"
300 DISPLAY AT(5,1):"1  Inst
ructions" :: DISPLAY AT(7,1)
:"2  Change  character #" ::
 DISPLAY AT(9,1):"3  Create
 characters"
310 DISPLAY AT(11,1):"4  Loa
d printer" :: DISPLAY AT(13,
1):"5  Exit"
320 ACCEPT AT(10,27)BEEP VAL
IDATE(DIGIT)SIZE(1):MENU
330 GOTO 480 !
    MENU
340 !
    CREATE NEW FILE
350 CALL CLEAR :: DISPLAY AT
(10,1):"Create new FILE
Y/N ?  Y"
360 ACCEPT AT(10,28)SIZE(-1)
VALIDATE("YN")BEEP:NF$
370 IF NF$="N" THEN 1680
380 GOSUB 410 !
    CREATE NEW CHAR. FILE
390 FOR I=0 TO 95 :: PRINT #
1,REC I:"" :: NEXT I
400 GOTO 1700 !
    MODIFY CHR. ROUTINE
410 REM
    FILE ATTRIBUTES
420 CALL CLEAR :: DISPLAY AT
(6,6):"DATA FILE NAME"
430 DISPLAY AT(7,6):"-------
--------" :: DISPLAY AT(10,6
):"DSK";DSK :: ACCEPT AT(10,
10)SIZE(-1)VALIDATE("123")BE
EP:DSK
440 DISPLAY AT(12,6):"FILENA
ME ";NAME$
450 ACCEPT AT(12,15)SIZE(-10
)VALIDATE(UALPHA,NUMERIC)BEE
P:NAME$
460 OPEN #1:"DSK"&STR$(DSK)&
```

```
","&NAME$,RELATIVE 96,INTERN
AL,UPDATE,FIXED 34
470 RETURN
480 IF MENU=1 THEN 630
490 IF MENU=2 THEN 690
500 IF MENU=3 THEN 340
510 IF MENU=4 THEN 940
520 IF MENU=5 THEN 540
530 IF MENU>5 THEN 630
540 CALL CLEAR :: DISPLAY AT
(5,13):"EXIT" :: DISPLAY AT(
5,13):"----"
550 DISPLAY AT(10,1):"ARE YO
U SURE ? (Y/N)  N" :: ACCEPT
 AT(10,23)SIZE(-1)BEEP VALID
ATE("YN"):EXIT$
560 IF EXIT$="Y" THEN 570 EL
SE 270
570 CALL CLEAR :: DISPLAY AT
(5,9):"EXIT OPTIONS" :: DISP
LAY AT(6,9):"------------"
580 DISPLAY AT(8,1):"1  Prin
ter set to download     char
acter set."
590 DISPLAY AT(12,1):"2  Pri
nter returned to std.    cha
racter set."
600 DISPLAY AT(18,14):"1" ::
 ACCEPT AT(18,14)BEEP VALIDA
TE("12")SIZE(-1):OPTION$
610 IF OPTION$="1" THEN END
ELSE 620
620 OPEN #2:"PIO" :: PRINT #
2:CHR$(27);CHR$(36);CHR$(0):
: CLOSE #2 :: END
630 REM
    INSTRUCTIONS
640 CALL CLEAR :: DISPLAY AT
(3,9):"Instructions" :: DISP
LAY AT(4,9):"------------"
650 DISPLAY AT(8,1):"Load di
splay  variable  80" :: DISP
LAY AT(10,1):"File 'DOWNLDHE
LP' into"
660 DISPLAY AT(12,1):"TI-WRI
TER  or  ED/ASS."
670 GOSUB 900 !
    MESSAGE
680 GOTO 270 !
    MENU
690 REM
    CHANGE CHR. ASCII #
700 GOSUB 410 !
    FILE ATTRIBUTES
710 CALL CLEAR
720 DISPLAY AT(4,4):"Origina
l ASCII # =" :: ACCEPT AT(4,
23)SIZE(3)VALIDATE(DIGIT)BEE
P:OLDNUM
730 INPUT #1,REC OLDNUM-32:B
$
740 IF B$="" THEN 750 ELSE 7
90
750 CALL CLEAR :: DISPLAY AT
(10,1):"NO DATA ON FILE FOR"
760 DISPLAY AT(12,1):"ASCII
# ";OLDNUM
770 GOSUB 900 !
    MESSAGE
780 GOTO 710
790 DISPLAY AT(8,4):"New ASC
II #   = " :: ACCEPT AT(8,21
)BEEP SIZE(3)VALIDATE(DIGIT)
:NEWNUM
800 INPUT #1,REC NEWNUM-32:C
$
810 IF C$="" THEN 850 ELSE 8
20
820 DISPLAY AT(12,1):"Data a
lready in this record" :: DI
SPLAY AT(15,1):"DO YOU WISH
TO PROCEED ? Y/N"
830 DISPLAY AT(18,14):"N" ::
 ACCEPT AT(18,14)BEEP VALIDA
TE(UALPHA,"YN")SIZE(-1):CONF
IRM$
840 IF CONFIRM$="N" THEN 710
850 PRINT #1,REC NEWNUM-32:B
$ :: B$="" :: C$="" :: PRINT
 #1,REC OLDNUM-32:B$
860 CALL CLEAR :: DISPLAY AT
(4,1):"Another character ? Y
/N  Y"
870 ACCEPT AT(4,26)SIZE(-1)V
ALIDATE("YN")BEEP:CONFIRM$
880 IF CONFIRM$="Y" THEN 710
890 CLOSE #1 :: GOTO 270 !
    MAIN MENU
900 PRINT :ANYK$ !
    MESSAGE
910 CALL KEY(5,K,S)
920 IF S=0 THEN 910
930 RETURN
940 OPEN #2:"PIO" !
    LOAD PRINTER ROUTINE
950 CALL CLEAR :: DISPLAY AT
(2,10):"LOAD PRINTER" :: DIS
PLAY AT(3,10):"------------"
960 DISPLAY AT(7,1):"1  Load
 saved character's      all
others blank "
970 DISPLAY AT(11,1):"2  Loa
d saved character's      on
top of std. characters"
980 DISPLAY AT(15,1):"3  Mai
n menu "
990 ACCEPT AT(10,28)SIZE(1)B
EEP VALIDATE(DIGIT):PRINTER
1000 IF PRINTER=1 THEN GOSUB
 1110 !   LOAD SAVED CHR. SE
T ONLY
1010 IF PRINTER=2 THEN 1080
!    LOAD CHR. FILE
1020 IF PRINTER=3 THEN 1040
!    MAIN MENU
1030 IF PRINTER>3 THEN 950 !
    TRY AGAIN
1040 CLOSE #2 :: GOTO 270 !
    MAIN MENU
1050 REM
    SELECT DOWNLOADABLE CHR
. SET
1060 PRINT #2:CHR$(27);CHR$(
36);CHR$(1)
1070 RETURN
1080 REM
    LOAD CHARACTER SET FILE
1090 PRINT #2:CHR$(27);CHR$(
42);CHR$(0);!
    LOAD STD. CHR. SET INTO
 RAM
1100 GOSUB 1050 !
    SELECT DOWNLOADABLE CHA
RACTERS
1110 CALL CLEAR :: DISPLAY A
T(4,1):"PRINT CHARACTERS AS
LOADED" :: DISPLAY AT(5,1):"
-------------------------"
1120 DISPLAY AT(8,9):"Y/N =
Y" :: ACCEPT AT(8,15)SIZE(-1
)BEEP VALIDATE("YN"):PRINT$
1130 GOSUB 410 !
    FILE ATTRIBUTES
1140 FOR I=0 TO 95
1150 INPUT #1,REC I:A$
1160 IF A$="" THEN 1270 !
    NEXT CHAR.
1170 COUNTER=0 :: Y=0
1180 FOR X=1 TO LEN(A$)
1190 SC$(Y)=SEG$(A$,X,1)
1200 IF SC$(Y)="," THEN 1210
 ELSE 1230
1210 SC$(Y)="" :: TX$(COUNTE
R)=SC$(0)&SC$(1)&SC$(2):: CO
UNTER=COUNTER+1 :: Y=0 :: SC
$(0)="" :: SC$(1)="" :: SC$(
2)=""
1220 LAST=I+32 :: GOTO 1240
1230 Y=Y+1
1240 NEXT X
1250 GOSUB 1310 !
    PRINT SINGLE CHAR.
1260 PRINT "ASCII #";I+32: :
: PRINT "CODE   ";A$: :: SC$
(0)="" :: SC$(1)="" :: SC$(2
)=""
1270 NEXT I
1280 GOSUB 900 !
    MESSAGE
1290 CLOSE #1 :: GOSUB 2730
!   CLEAR ARRAY
1300 GOTO 950 !
   LOAD PRINTER MENU
1310 REM
    PRINT SINGLE CHAR.
1320 FOR Z=0 TO 9 :: TX(Z)=V
AL(TX$(Z)):: NEXT Z
1330 PRINT #2:CHR$(27);CHR$(
42);CHR$(1);CHR$(LAST);CHR$(
TX(0));CHR$(TX(1));CHR$(TX(2
));CHR$(TX(3));CHR$(TX(4));C
HR$(TX(5));CHR$(TX(6));
1340 PRINT #2:CHR$(TX(7));CH
R$(TX(8));CHR$(TX(9));
1350 IF PRINT$="Y" THEN 1360
 ELSE 1410
1360 PRINT #2:CHR$(27);CHR$(
36);CHR$(0)!
    CANCEL DOWNLD CHRS.
1370 PRINT #2:CHR$(27);CHR$(
51);CHR$(1)!
    ADJUST LINE FEED 1/144
1380 PRINT #2:"ASCII #";LAST
;"="
```

```
1390 GOSUB 1050 !
     DOWN LD. CHR. SET
1400 PRINT #2:TAB(15);CHR$(L
AST):: PRINT #2:CHR$(27);CHR
$(74);CHR$(16)!
     ONE TIME LF 16/144
1410 RETURN
1420 REM
     TEST ROUTINE
1430 OPEN #2:"PIO" :: PRINT$
="Y"
1440 GOSUB 1050 !
     DOWN LD. CHR.SET
1450 GOSUB 1310 !
     PRINT SINGLE CHR.
1460 CLOSE #2
1470 GOSUB 2730 !
    CLEAR ARRAY
1480 REM
     CHARACTER # REQUEST ROU
TINE
1490 CALL CLEAR
1500 DISPLAY AT(4,6):"-CHARA
CTER NUMBER-" :: DISPLAY AT(
5,6):"----------------"
1510 DISPLAY AT(13,6):"NEXT
    # = 1"
1520 DISPLAY AT(15,6):"PREVI
OUS # = 2"
1530 DISPLAY AT(17,6):"LAST
    # = 3"
1540 DISPLAY AT(19,6):"NEW
    # = 4"
1550 DISPLAY AT(21,6):"TEST
    # = 5"
1560 DISPLAY AT(23,6):"MENU
    # = 6"
1570 DISPLAY AT(8,10):"LAST
# ";LAST
1580 DISPLAY AT(16,24):CHR$(
32):: DISPLAY AT(16,24):CHR$
(49):: FOR X=1 TO 25 :: NEXT
 X
1590 CALL KEY(5,NEWNUM,S)
1600 IF S=0 THEN 1580
1610 IF NEWNUM=13 OR NEWNUM=
49 THEN LAST=LAST+1 :: GOTO
1700
1620 IF NEWNUM=50 THEN LAST=
LAST-1 :: GOTO 1700
1630 IF NEWNUM=51 THEN LAST=
LAST :: GOTO 1700
1640 IF NEWNUM=52 THEN 1700
1650 IF NEWNUM=53 THEN GOSUB
 1420 !
     TEST SINGLE CHAR.
1660 IF NEWNUM=54 THEN 2250
!    MENU
1670 GOTO 1590 !
     CALL KEY
1680 REM
     MODIFY CHARACTER ROUTIN
E
1690 GOSUB 410 !
    FILE ATTRIBUTES
1700 TX$(0)="0"
1710 IF LAST<32 THEN LAST=32
1720 IF LAST>126 THEN LAST=1
26
1730 RESTORE
1740 CALL CLEAR
1750 CALL COLOR(14,2,1)
1760 CALL CHAR(136,"FF818181
818181FF")
1770 FOR X=1 TO 7 :: READ DA
T
1780 DISPLAY AT(6+X,5):DAT :
: NEXT X
1790 DISPLAY AT(1,9):"Create
  CHR  # "
1800 DISPLAY AT(20,1):"* ADJ
ACENT HOR SQUARES MUST"
1810 DISPLAY AT(21,3):"NOT B
E FILLED."
1820 DISPLAY AT(23,1):"Clear
     Decenders    Menu"
1830 DISPLAY AT(24,1):"New n
umber    Save"
1840 DISPLAY AT(2,15):"Use #
's 32-126"
1850 GOSUB 1920 !
     CLEAR GRID
1860 IF NEWNUM<52 THEN 1870
:: ACCEPT AT(1,24)SIZE(3)BEE
P VALIDATE(DIGIT):LAST
1870 IF LAST<32 OR LAST>126
THEN 1860
1880 FOR DELAY=1 TO 100 :: N
EXT DELAY !  KEY PRESS DELAY
1890 GOSUB 1970 !
     DISPLAY LAST NUMBER
1900 V=7 :: H=8
1910 GOTO 2100 !
     CALL KEY
1920 REM
     CLEAR GRID ROUTINE
1930 FOR X=7 TO 15
1940 CALL HCHAR(X,10,136,9)
1950 NEXT X
1960 RETURN
1970 REM
     DISPLAY LAST NUMBER ROU
TINE
1980 LAST$=STR$(LAST)
1990 FOR Y=1 TO LEN(LAST$)::
 Y$=SEG$(LAST$,Y,1)
2000 CALL HCHAR(1,25+Y,ASC(Y
$),1)
2010 NEXT Y :: V=7 :: H=8 ::
 RETURN
2020 F=1 !
     POSITION CURSOR DARK
2030 DISPLAY AT(V,H)SIZE(1):
CHR$(136)
2040 DISPLAY AT(V,H)SIZE(1):
CHR$(30):: FOR X=1 TO 25 ::
NEXT X
2050 GOTO 2100 !
     CALL KEY
2060 F=0 !
     POSITION CURSOR LIGHT
2070 DISPLAY AT(V,H)SIZE(1):
CHR$(32)
2080 DISPLAY AT(V,H)SIZE(1):
CHR$(136):: FOR X=1 TO 25 ::
 NEXT X
2090 GOTO 2100
2100 CALL KEY(5,K,S)
2110 IF S=0 THEN IF F=1 THEN
 2020 ELSE 2060
2120 IF K=67 THEN GOSUB 1920
 !
     CLEAR GRID
2130 IF K=68 THEN 2260 !
     DECENDERS
2140 IF K=77 THEN 2250 !
     MAIN  MENU
2150 IF K=78 THEN NEWNUM=52
:: GOTO 1700 !
     NEW NUMBER
2160 IF K=83 THEN 2300 !
     SAVE
2170 IF K=11 THEN GOTO 2510
!    SHIFT UP
2180 IF K=9 THEN GOTO 2540 !
     SHIFT RIGHT
2190 IF K=8 THEN GOTO 2570 !
     SHIFT LEFT
2200 IF K=10 THEN GOTO 2600
!    SHIFT DOWN
2210 IF K=4 THEN 2020 !
     CURSOR
2220 IF K=3 THEN 2060 !
     CURSOR
2230 GOSUB 2770 !
    WRONG!
2240 GOTO 2100
2250 CLOSE #1 :: GOTO 270 !
     MAIN MENU
2260 REM
     DECENDERS ROUTINE
2270 FOR CLEAR=1 TO 5 :: CAL
L HCHAR(13+CLEAR,10,32,9)::
NEXT CLEAR
2280 FOR X=5 TO 6 :: CALL HC
HAR(X,10,136,9):: NEXT X
2290 TX$(0)="1"
2300 REM
     SAVE ROUTINE
2310 FOR H=10 TO 18
2320 COUNTER=1
2330 FOR V=7 TO 13 :: CALL G
CHAR(V,H,A):: CALL GCHAR(V,H
+1,B):: IF A+B=60 THEN 2630
!   TEST FOR ADJACENT SQUARE
S FILLED
2340 IF A=30 THEN TX(H-9)=TX
(H-9)+COUNTER
2350 COUNTER=COUNTER+COUNTER
2360 NEXT V
2370 TX$(H-9)=STR$(TX(H-9))
2380 FOR Z=1 TO 3
2390 IF SEG$(TX$(H-9),Z,1)="
" THEN 2410
2400 DISPLAY AT(Z+15,H-2)SIZ
E(1):CHR$(ASC(SEG$(TX$(H-9),
Z,1)))
2410 NEXT Z
2420 NEXT H
2430 REM
     PRINT RECORDS TO DISK
2440 TXTOT$=TX$(0)&","&TX$(1
)&","&TX$(2)&","&TX$(3)&","&
TX$(4)&","&TX$(5)&","&TX$(6)
```

```
&","&TX$(7)&","&TX$(8)&","&T
X$(9)&","
2450 PRINT #1,REC LAST-32:TX
TOT$
2460 GOSUB 2730 !
    CLEAR ARRAY VARIABLES
2470 GOSUB 1920 !
     CLEAR GRID
2480 GOSUB 1480 !
     CHARACTER # REQUEST
2490 GOSUB 1970 !
     DISPLAY LAST NUMBER
2500 GOTO 2100
2510 REM
     SHIFT CURSOR UP
2520 V=V-1 :: IF V=6 THEN V=
13
2530 IF F=0 THEN 2060 ELSE 2
020
2540 REM
     SHIFT CURSOR RIGHT
2550 H=H+1 :: IF H=17 THEN H
=8
2560 IF F=0 THEN 2060 ELSE 2
020
2570 REM
     SHIFT CURSOR LEFT
2580 H=H-1 :: IF H=7 THEN H=
16
2590 IF F=0 THEN 2060 ELSE 2
020
2600 REM
     SHIFT CURSOR DOWN
2610 V=V+1 :: IF V=14 THEN V
=7
2620 IF F=0 THEN 2060 ELSE 2
020
2630 REM
     ADJACENT SQUARES FILLED
2640 GOSUB 2770 !
    WRONG!
2650 FOR X=1 TO 10
2660 DISPLAY AT(20,1)SIZE(1)
:CHR$(32)
2670 FOR DELAY=1 TO 50 :: NE
XT DELAY
2680 DISPLAY AT(20,1)SIZE(1)
:CHR$(42)
2690 NEXT X
2700 V=7 :: H=8
2710 GOSUB 2730 !
    CLEAR ARRAY
2720 GOTO 2100
2730 REM
     CLEAR ARRAY VARIABLES
2740 V=7 :: H=8 :: FOR COUNT
=0 TO 9 :: TX(COUNT)=0 :: NE
XT COUNT
2750 FOR CLEAR=1 TO 2 :: CAL
L HCHAR(15+CLEAR,10,32,9)::
NEXT CLEAR
2760 RETURN
2770 CALL SOUND(300,150,5,18
0,5)!
     WRONG!(BEEP)
2780 RETURN
```

```
############################
#                          #
# A disk cataloger with a  #
# difference.  This one    #
# gives you the choice of: #
#                          #
#  1) Displaying catalog   #
#     on the screen.       #
#  2) Prinring the catalog #
#     to a printer.        #
#  3) As in 2 but adding   #
#     comments to the      #
#     listing.             #
#                          #
############################



100 CALL CLEAR
110 DIM TYPE$(5)
120 TYPE$(1)="DIS/FIX"
130 TYPE$(2)="DIS/VAR"
140 TYPE$(3)="INT/FIX"
150 TYPE$(4)="INT/VAR"
160 TYPE$(5)="PROGRAM"
170 INPUT "MASTER DISK(1-3)?
":A
180 A=INT(A)
190 IF A<1 THEN 170
200 IF A>3 THEN 170
210 INPUT "PRINTER LISTING Y
/N ":YN$
220 IF YN$="Y" THEN 230 ELSE
 300
230 YES=1
240 INPUT "COMMENTS REQUIRED
 Y/N ":YN1$
250 IF YN1$="Y" THEN 260 ELS
E 280
260 COM=1
270 GOTO 320
280 COM=0
290 GOTO 320
300 YES=0
310 GOTO 330
320 OPEN #2:"PIO"
330 OPEN #1:"DSK"&STR$(A)&".
",INPUT ,RELATIVE,INTERNAL
340 INPUT #1:A$,J,J,K
350 DISPLAY "DSK";STR$(A);"
-DISKNAME= ";A$:"AVAILABLE="
;K;"USED=";J-K
360 IF YES=1 THEN 370 ELSE 3
80
370 PRINT #2:TAB(8);"DISKNAM
E  =  ";A$:TAB(8);"AVAILABLE
 =";K;"USED =";J-K
380 DISPLAY :"FILENAME   SIZ
E     TYPE   P":"----------
---------- -------
                 ";
390 IF YES=1 THEN 400 ELSE 4
10
400 PRINT #2:TAB(8);"FILENAM
E   SIZE    TYPE     P   COM
MENTS  ";TAB(10);"----------
 ----  ---------  -  -------
----------------------------"
410 FOR LOOP=1 TO 127
420 INPUT #1:A$,A,J,K
430 IF LEN(A$)=0 THEN 590
440 DISPLAY :A$;TAB(12);J;TA
B(17);TYPE$(ABS(A));
450 IF YES=1 THEN 460 ELSE 4
70
460 PRINT #2:TAB(8);A$;TAB(2
0);J;TAB(25);TYPE$(ABS(A));
470 IF ABS(A)=5 THEN 520
480 B$="  "&STR$(K)
490 DISPLAY SEG$(B$,LEN(B$)-
2,3);
500 IF YES=1 THEN 510 ELSE 5
20
510 PRINT #2:SEG$(B$,LEN(B$)
-2,3);
520 IF A>0 THEN 550
530 DISPLAY TAB(28);"Y";
540 PRINT #2:TAB(36);"Y";
550 IF COM=1 THEN 560 ELSE 5
80
560 INPUT "COMMENT : ":COMME
NT$
570 PRINT #2:TAB(40);COMMENT
$
580 NEXT LOOP
590 CLOSE #1
600 IF YES=1 THEN 620 ELSE 6
40
610 PRINT #2:TAB(8);"-------
----------------------------
----------------------------
------"
620 PRINT #2:CHR$(12)
630 CLOSE #2
640 END



############################
#                          #
############################



100 REM ******************
110 REM * Reads data from *
120 REM *  a Dis-Var 80   *
130 REM *  and writes it  *
140 REM * to a Dis-Fix 128*
150 REM * file that can be*
160 REM *   read by the   *
170 REM *    DPPROGRAM    *
180 REM * Written by Phil *
190 REM * West.
200 REM *                 *
210 REM *    L.C.Twiss    *
220 REM *     June 1986   *
230 REM ******************
240 CALL CLEAR :: CALL SCRN
:: CALL CHR :: CALL WINDOW
250 DISPLAY AT(2,2):"This
program   converts  Dis-Var
  80   files   into  Dis-Fix
 128 files utilized  by the
DBPROGRAM."
260 DISPLAY AT(10,2):"Name o
f original Data File:" :: DI
SPLAY AT(11,2)SIZE(-5):"DSK1
." :: ACCEPT AT(11,5)SIZE(-1
```

```
):DN$
270 IF DN$="" THEN DN$="DSK1
."
280 ACCEPT AT(11,7)SIZE(-10)
:FN$ :: IF FN$="" THEN 270
290 DB$="DSK"&DN$&"."&FN$
300 DISPLAY AT(12,2):"Name o
f new Data Base file:" :: DI
SPLAY AT(13,2)SIZE(-5):"DSK1
." :: DISPLAY AT(13,7)SIZE(-
3):"DBD"
310 ACCEPT AT(13,5)SIZE(-1):
DNN$
320 IF DNN$="" THEN DNN$="DS
K1"
330 ACCEPT AT(13,10)SIZE(-7)
:FNN$ :: IF FNN$="" THEN 330
340 DBN$="DSK"&DNN$&"."&"DBD
"&FNN$
350 DISPLAY AT(18,2)SIZE(-24
):"Reading file : ";FN$ :: D
ISPLAY AT(22,2)SIZE(-24):"Wr
iting file : DBD";FNN$
360 OPEN #1:DB$
370 OPEN #2:DBN$,RELATIVE,FI
XED 128,OUTPUT
380 LINPUT #1:A$
390 IF EOF(1)THEN 660
400 ! 370 Ignores line if li
ne is blank.
      380 If first Ch is a s
psce line is ignored.
410 ! 390 If first Ch is a -
 the line is ignored.
      400 Reads the surname
from the first 14 Ch of line
420 ! 410 reads the First na
me from the next 12 Ch.
      420 Reads the Street f
rom the next 19 Ch of line
430 ! 430 reads the Town fro
m the next 15 Ch of line.
      440 Reads the Post Cod
e from the next 4 Ch of line
440 IF A$="" THEN 380
450 IF SEG$(A$,1,1)=" " THEN
 380
460 IF SEG$(A$,2,1)="-" THEN
 380
470 GO SUB 670
480 SURNAME$=SEG$(AA$,1,14)
490 FIRSTNAME$=SEG$(AA$,15,1
2)
500 STREET$=SEG$(AA$,27,19)
510 TOWN$=SEG$(AA$,46,15)
520 POSTCODE$=SEG$(AA$,61,4)
530 ! 460-510 Reads the post
 code and adds the state. Ed
iting may be necessary for
the ACT and NT.
540 IF SEG$(POSTCODE$,1,1)="
6" THEN STATE$="WA"
550 IF SEG$(POSTCODE$,1,1)="
2" THEN STATE$="NSW"
560 IF SEG$(POSTCODE$,1,1)="
3" THEN STATE$="VIC"
570 IF SEG$(POSTCODE$,1,1)="
4" THEN STATE$="QLD"
580 IF SEG$(POSTCODE$,1,1)="
5" THEN STATE$="SA"
590 IF SEG$(POSTCODE$,1,1)="
7" THEN STATE$="TAS"
600 ! 530 Reads the Phone nu
mber and prefix from the nex
t 9 Ch of line.
610 PHONE$=SEG$(AA$,69,9)
620 ! 550 Writes to DBPROGRA
M file in the correct format
 for that program to read th
e file.
630 DISPLAY AT(20,2)SIZE(28)
:FIRSTNAME$&SURNAME$
640 PRINT #2:TAB(0);SURNAME$
&","&FIRSTNAME$;TAB(29);STRE
ET$;TAB(57);TOWN$;TAB(85);ST
ATE$;TAB(88);POSTCODE$;TAB(9
2);PHONE$;
650 GOTO 380
660 CLOSE #1 :: CLOSE #2 ::
RUN "DSK1.LOAD"
670 ! CONVERSION ROUTINE
680 NM$=""
690 FOR I=1 TO LEN(A$)
700 CH$=SEG$(A$,I,1)
710 CH=ASC(CH$)
720 IF (CH<123)AND(CH>96)THE
N 730 ELSE 740
730 CH=CH-32
740 NM$=NM$&CHR$(CH)
750 NEXT I
760 AA$=NM$
770 A$=""
780 RETURN
790 SUB SCRN
800 CALL SCREEN(13)
810 FOR COL=8 TO 12
820 CALL COLOR(COL,16,13)
830 NEXT COL
840 SUBEND
850 SUB CHR
860 CALL CHAR(128,"00FFFFC0C
0C0C0C0",129,"00FFFF00000000
00",130,"00FFFF0303030303")
870 CALL CHAR(131,"C0C0C0C0C
0C0C0C0",132,"0303030303030303
03")
880 CALL CHAR(133,"C0C0C0C0C
0FFFF00",134,"0000000000FFFF
00",135,"0303030303FFFF00")
890 CALL COLOR(13,16,13)
900 SUBEND
910 SUB WINDOW
920 A=1 :: B=1
930 FOR AA=1 TO 3
940 CALL HCHAR(A,B,128):: CA
LL HCHAR(A,B+1,129,30):: CAL
L HCHAR(A,B+31,130):: CALL V
CHAR(A+1,B,131,5):: CALL VCH
AR(A+1,B+31,132,5)
950 CALL HCHAR(A+6,B,133)::
CALL HCHAR(A+6,B+1,134,30)::
 CALL HCHAR(A+6,B+31,135)
960 A=A+8
970 NEXT AA
980 SUBEND
990 SUB BOX
1000 CALL CHAR(136,"000000FF
FF000000"):: CALL COLOR(14,7
,13):: CALL HCHAR(19,9,136,2
3):: CALL HCHAR(21,9,136,23)
1010 CALL CHAR(137,"000000FF
FFC0C0C0",138,"C0C0C0FFFF000
000",139,"C0C0C0C0C0C0C0C0")
:: CALL HCHAR(19,9,137):: CA
LL HCHAR(21,9,138)
1020 CALL HCHAR(20,9,139)
1030 SUBEND


##############################
#                            #
##############################


100 ! **********************
110 ! *                    *
120 ! * This program will *
130 ! * read files from a *
140 ! * data base file and*
150 ! * is used to print  *
160 ! * mail wrappers.    *
170 ! *                    *
180 ! *    Written By      *
190 ! *    L.C.Twiss       *
200 ! *                    *
210 ! * To complement the *
220 ! * DBPROGRAM written *
230 ! * by Phil West.     *
240 ! *                    *
250 ! *TI-Users Perth Inc.*
260 ! *                    *
270 ! *    June 1986       *
280 ! *                    *
290 ! **********************
300 CALL CLEAR
310 CALL ENTER
320 CALL SCRN
330 CALL WINDOW
340 COUNT=0
350 !READS DB FILES IN DISPL
AY FIXED 128 FORMAT
360 DISPLAY AT(2,2)SIZE(-30)
:"This program  will read
" :: DISPLAY AT(3,2)SIZE(-30
):"Data  files written by th
e"
370 DISPLAY AT(4,2)SIZE(-30)
:"DBPROGRAM  and  generate a
" :: DISPLAY AT(5,2)SIZE(-30
):"mail wrapper with a retur
n"
380 DISPLAY AT(6,2)SIZE(-30)
:"address."
390 DISPLAY AT(8,2)SIZE(-30)
:"Enter your data here : " :
: DISPLAY AT(11,2)SIZE(-5):"
DSK1." :: DISPLAY AT(11,7)SI
ZE(-10):"DBDMEMBER"
400 ACCEPT AT(11,5)SIZE(-1):
DDNUM$
410 IF DDNUM$=" " THEN DDNUM
$="1"
420 ACCEPT AT(11,10)SIZE(-7)
:FILE$ :: IF FILE$="" THEN F
```

```
ILE$="MEMBER"
430 FILNAM$="DBD"&FILE$
440 DISPLAY AT(11,2)SIZE(-20
):"Printer :PIO"
450 ACCEPT AT(11,11)SIZE(-18
):PRINTER$ :: IF PRINTER$="
" THEN PRINTER$="PIO"
460 OPEN #2:PRINTER$,OUTPUT
470 DISPLAY AT(10,2)SIZE(-20
):"Return Address :" :: DISP
LAY AT(11,2)SIZE(-30):"TI-Us
ers Perth Inc."
480 DISPLAY AT(12,2)SIZE(-30
):"P.O.Box 246" :: DISPLAY A
T(13,2)SIZE(-30):"Mt Lawley"
 :: DISPLAY AT(14,2)SIZE(-30
):"W.A.   6050"
490 ACCEPT AT(11,2)SIZE(-30)
:CLNAM$ :: IF CLNAM$="" THEN
 CLNAM$="TI-Users Perth Inc.
"
500 ACCEPT AT(12,2)SIZE(-30)
:CLADD$ :: IF CLADD$="" THEN
 CLADD$="P.O. Box 246"
510 ACCEPT AT(13,2)SIZE(-30)
:CLTWN$ :: IF CLTWN$="" THEN
 CLTWN$="Mt. Lawley"
520 ACCEPT AT(14,2)SIZE(-30)
:CLPCD$ :: IF CLPCD$="" THEN
 CLPCD$="W.A.   6050"
530 OPEN #1:"DSK"&DDNUM$&"."
&FILNAM$,FIXED 128,INPUT
540 LINPUT #1:B$
550 IF EOF(1)THEN 980
560 IF SEG$(B$,1,1)=" " THEN
 540
570 A$=SEG$(B$,1,28)
580 FOR I=1 TO 28
590 IF SEG$(A$,I,1)="," THEN
 600 ELSE 630
600 LASTNAME$=SEG$(A$,1,I-1)
610 A=I
620 I=28
630 NEXT I
640 FOR J=A TO 28
650 IF SEG$(A$,J,1)=" " THEN
 660 ELSE 690
660 FIRSTNAM$=SEG$(A$,A+1,J-
A)
670 NAM$=FIRSTNAM$&LASTNAME$
680 J=28
690 NEXT J
700 CALL LINE
710 DISPLAY AT(18,2)SIZE(-30
):"Press 'Y' to print name."
720 DISPLAY AT(22,2)SIZE(-7)
:"COUNT :"
730 DISPLAY AT(20,2)SIZE(-4)
:"Name" :: DISPLAY AT(20,8)S
IZE(-22):NAM$
740 CALL KEY(0,K,S):: IF S=0
 THEN 740
750 IF (K=89)+(K=121)THEN 76
0 ELSE 540
760 COUNT=COUNT+1
770 DISPLAY AT(22,10)SIZE(-3
):COUNT
780 ADDRESS$=SEG$(B$,29,28)
790 SUBURB$=SEG$(B$,57,28)
800 ST$=SEG$(B$,85,3)
810 PCODE$=SEG$(B$,88,4)
820 LENGTH=LEN(ST$)
830 IF LENGTH>2 THEN 850
840 STATE$=ST$&"    "&PCODE$
 :: GOTO 870
850 STATE$=ST$&"   "&PCODE$
860 FOR I=1 TO 25 :: PRINT #
2 :: NEXT I
870 PRINT #2:TAB(30);NAM$
880 PRINT #2:TAB(30);ADDRESS
$
890 PRINT #2:TAB(30);SUBURB$
900 PRINT #2:TAB(30);STATE$
910 FOR I=1 TO 7 :: PRINT #2
 :: NEXT I
920 PRINT #2:TAB(5);CLNAM$
930 PRINT #2:TAB(5);CLADD$
940 PRINT #2:TAB(5);CLTWN$
950 PRINT #2:TAB(5);CLPCD$
960 PRINT #2:CHR$(12)
970 GOTO 540
980 CLOSE #1 :: CLOSE #2
990 END
1000 SUB SCRN
1010 CALL SCREEN(13)
1020 FOR COL=0 TO 12
1030 CALL COLOR(COL,16,13)
1040 NEXT COL
1050 SUBEND
1060 SUB ENTER
1070 CALL CHAR(128,"00FFFFC0
C0C0C0C0",129,"00FFFF0000000
000",130,"00FFFF0303030303")
1080 CALL CHAR(131,"C0C0C0C0
C0C0C0C0",132,"0303030303030
303")
1090 CALL CHAR(133,"C0C0C0C0
C0FFFF00",134,"000000000FFF
F00",135,"0303030303FFFF00")
1100 CALL COLOR(13,16,13)
1110 SUBEND
1120 SUB WINDOW
1130 A=1 :: B=1
1140 FOR AA=1 TO 3
1150 CALL HCHAR(A,B,128):: C
ALL HCHAR(A,B+1,129,30):: CA
LL HCHAR(A,B+31,130):: CALL
VCHAR(A+1,B,131,5):: CALL VC
HAR(A+1,B+31,132,5)
1160 CALL HCHAR(A+6,B,133)::
 CALL HCHAR(A+6,B+1,134,30):
: CALL HCHAR(A+6,B+31,135)
1170 A=A+8
1180 NEXT AA
1190 SUBEND
1200 SUB LINE
1210 CALL CHAR(136,"000000FF
FF000000"):: CALL COLOR(14,7
,13):: CALL HCHAR(19,9,136,2
3):: CALL HCHAR(21,9,136,23)
1220 CALL CHAR(137,"000000FF
FFC0C0C0",138,"C0C0C0FFFF000
000",139,"C0C0C0C0C0C0C0C0")
:: CALL HCHAR(19,9,137):: CA
LL HCHAR(21,9,138)
1230 CALL HCHAR(20,9,139)
1240 SUBEND


############################
#                          #
# A mathematics tutorial   #
# for either Basic or XB.  #
#                          #
############################


100 CALL CLEAR
110 CALL SCREEN(16)
120 PRINT "    M A T H A M A
 T I C S"
130 PRINT
140 PRINT
150 PRINT
160 PRINT
170 PRINT TAB(12);"BY VON KL
IMPEL"
180 PRINT
190 PRINT TAB(17);"TICHUG"
200 GOSUB 2260
210 INPUT "PLEASE ENTER YOUR
 NAME?":NAME$
220 GOSUB 2310 REM DELAY
230 CALL CLEAR
240 PRINT TAB(11);"OK ";NAME
$
250 PRINT
260 PRINT TAB(9);"YOUR CHOIC
E"
270 PRINT
280 PRINT
290 PRINT
300 PRINT "1. INSTRUCTIONS"
310 PRINT
320 PRINT "2. MULTIPLICATION
 TABLES "
330 PRINT
340 PRINT "3. RANDOM MULTIPL
ICATIONS"
350 PRINT
360 PRINT "4. ADDITION   "
370 PRINT
380 PRINT "5. SUBTRACTION"
390 PRINT
400 PRINT "6. DIVISION "
410 PRINT
420 PRINT
430 PRINT
440 PRINT
450 PRINT
460 PRINT "(1,2,3,4,5 OR 6)"
;
470 INPUT CHOICE
480 ON CHOICE GOSUB 500,700,
950,1170,1380,1620,2510
490 GOTO 2400
500 REM INSTRUCTIONS
510 CALL CLEAR
520 PRINT TAB(9);"INSTRUCTIO
NS"
530 PRINT
540 PRINT "    THIS PROGRAM
ALLOWS YOU TO CHOOSE THE MAT
```

```
HS FUNCTION "
550 PRINT
560 PRINT "    YOU WILL BE G
IVEN A     CHOICE OF FUNCTIO
NS AND ALL YOU HAVE TO DO IS
 PRESS THE APPROPRIATE ";
570 PRINT "ANSWER."
580 PRINT
590 PRINT "    TO STOP THE P
ROGRAM AT  ANY TIME JUST ENT
ER '000' AS YOUR ANSWER."
600 PRINT
610 PRINT "    TO RETURN TO
THE  INDEX LIST AND CHANGE T
HE MATHS   FUNCTION, JUST TY
PE IN THE  WORD 'INDEX' AS Y
OUR ANSWER"
620 PRINT
630 PRINT
640 PRINT
650 PRINT
660 PRINT "(PRESS ANY KEY TO
 CONTINUE)";
670 CALL KEY(0,KEY,STATUS)
680 IF STATUS=0 THEN 670
690 GOTO 230
700 REM MULTIPLICATION
       TABLES.....
710 CALL CLEAR
720 PRINT TAB(9);"RIGHT ";NA
ME$
730 PRINT
740 PRINT
750 PRINT "WHICH MULTIPLICAT
ION TABLE  WOULD YOU LIKE TO
 DO ";
760 INPUT TABLE
770 GOSUB 2310
780 CALL CLEAR
790 N=0
800 N=N+1
810 PRINT TAB(6);"TRY THIS O
NE ";NAME$
820 PRINT
830 PRINT
840 PRINT TAB(9);N;"X";TABLE
;
850 INPUT "=";ANS$
860 IF ANS$="INDEX" THEN 230
870 LET ANS=VAL(ANS$)
880 IF ANS=000 THEN 2400
890 IF ANS<>N*TABLE THEN 930
900 GOSUB 2080
910 PRINT
920 GOTO 800
930 GOSUB 1980
940 GOTO 840
950 REM RANDOM MULTIPLICATIO
NS
960 CALL CLEAR
970 PRINT TAB(6);NAME$;
980 PRINT " TRY THIS ONE";
990 PRINT
1000 PRINT
1010 RANDOMIZE
1020 FIRST=INT(10*RND)+1
1030 SECOND=INT(10*RND)+1
1040 PRINT TAB(9);FIRST;
1050 PRINT "X";
1060 PRINT SECOND;
1070 PRINT "=";
1080 INPUT ANS$
1090 IF ANS$="INDEX" THEN 23
0
1100 LET ANS=VAL(ANS$)
1110 IF ANS=000 THEN 2400
1120 IF ANS<>FIRST*SECOND TH
EN 1150
1130 GOSUB 2080
1140 GOTO 960
1150 GOSUB 1980
1160 GOTO 1040
1170 REM ADDITION
1180 CALL CLEAR
1190 PRINT TAB(6);"TRY THIS
ONE ";NAME$
1200 RANDOMIZE
1210 FIRST=INT(100*RND)+1
1220 SECOND=INT(100*RND)+1
1230 PRINT
1240 PRINT
1250 PRINT TAB(9);FIRST;
1260 PRINT "+";
1270 PRINT SECOND;
1280 PRINT "=";
1290 INPUT ANS$
1300 IF ANS$="INDEX" THEN 23
0
1310 LET ANS=VAL(ANS$)
1320 IF ANS=000 THEN 2400
1330 IF ANS<>FIRST+SECOND TH
EN 1360
1340 GOSUB 2080
1350 GOTO 1190
1360 GOSUB 1980
1370 GOTO 1230
1380 REM SUBTRACTION
1390 CALL CLEAR
1400 PRINT TAB(6);"TRY THIS
ONE ";NAME$
1410 PRINT
1420 PRINT
1430 RANDOMIZE
1440 FIRST=INT(100*RND)+1
1450 SECOND=INT(100*RND)+1
1460 IF SECOND>FIRST THEN 14
30
1470 PRINT TAB(6);FIRST;
1480 PRINT "-";
1490 PRINT SECOND;
1500 PRINT "=";
1510 IF FIRST=SECOND THEN 13
80
1520 INPUT ANS$
1530 IF ANS$="INDEX" THEN 23
0
1540 LET ANS=VAL(ANS$)
1550 IF ANS=000 THEN 2400
1560 IF ANS<>FIRST-SECOND TH
EN 1590
1570 GOSUB 2080
1580 GOTO 1390
1590 GOSUB 1980
1600 GOTO 1470
1610 RETURN
1620 REM DIVISION BY A SET
        FIGURE
1630 CALL CLEAR
1640 PRINT TAB(9);"OK ";NAME
$
1650 PRINT
1660 PRINT
1670 PRINT "WHICH NUMBER DO
YOU WISH TO DIVIDE BY";
1680 INPUT TABLE
1690 GOSUB 2260
1700 CALL CLEAR
1710 GOSUB 2310
1720 PRINT
1730 CALL CLEAR
1740 RANDOMIZE
1750 FIRST=INT(100*RND)+1
1760 IF FIRST/TABLE<>INT(FIR
ST/TABLE)THEN 1700
1770 PRINT TAB(9);"OK ";NAME
$
1780 PRINT
1790 PRINT TAB(8);"TRY THIS
ONE"
1800 PRINT
1810 PRINT
1820 PRINT
1830 PRINT TAB(3);FIRST;
1840 PRINT "DIVIDED BY";
1850 PRINT TABLE;
1860 PRINT "=";
1870 IF FIRST/TABLE<>INT(FIR
ST/TABLE)THEN 1730
1880 IF FIRST<=TABLE THEN 17
00
1890 INPUT ANS$
1900 IF ANS$="INDEX" THEN 23
0
1910 LET ANS=VAL(ANS$)
1920 IF ANS=000 THEN 2400
1930 IF ANS<>FIRST/TABLE THE
N 1960
1940 GOSUB 2080
1950 GOTO 1700
1960 GOSUB 1980
1970 GOTO 1830
1980 REM WRONG ROUTINE
1990 CALL SOUND(250,110,2,-6
,2)
2000 GOSUB 2260
2010 PRINT TAB(6);"YOU ARE W
RONG, ";NAME$
2020 PRINT
2030 PRINT
2040 PRINT TAB(6);"TRY THAT
ONE AGAIN"
2050 PRINT
2060 PRINT
2070 RETURN
2080 REM  CONGRATULATION
        SUBROUTINE
2090 CALL CLEAR
2100 PRINT TAB(8);"THAT'S CO
RRECT ",,;TAB(12);NAME$;,,TA
B(8);"CONGRATULATIONS"
2110 CALL SOUND(100,440,2)
2120 PRINT
2130 CALL SOUND(100,550,2)
2140 PRINT
```

```
2150 CALL SOUND(100,440,2)
2160 PRINT
2170 CALL SOUND(100,660,2)
2180 PRINT
2190 CALL SOUND(100,880,2)
2200 PRINT
2210 FOR I=3 TO 16
2220 CALL SCREEN(I)
2230 NEXT I
2240 CALL CLEAR
2250 RETURN
2260 REM SCROLLING ROUTINE

2270 FOR I=1 TO 8
2280 PRINT
2290 NEXT I
2300 RETURN
2310 REM     DELAY ROUTINE
2320 FOR DELAY=1 TO 50
2330 NEXT DELAY
2340 RETURN
2350 REM SWAP ROUTINE
2360 HOLD=FIRST
2370 FIRST=SECOND
2380 SECOND=FIRST
2390 GOTO 1470
2400 CALL CLEAR
2410 PRINT TAB(3);"HOPE YOU
ENJOYED YOURSELF"
2420 PRINT
2430 PRINT
2440 PRINT TAB(12);NAME$
2450 PRINT
2460 PRINT
2470 PRINT TAB(11);"GOOD BYE
"
2480 PRINT
2490 PRINT
2500 PRINT
2510 END
############################
#                          #
# Try this program out for #
# size.  The SPRITE demo   #
# is fantastic.            #
#                          #
#   EXTENDED BASIC ONLY    #
#                          #
############################


100 ! ***** SPIRCLE *****
110 !
120 ! BY GENE KRACKEDWITS
130 !
140 !
150 CALL CLEAR
160 CALL SCREEN(2)
170 CALL MAGNIFY(3)
180 A$="07182040408080808080
80404020180 7E0180402020101 01
0101010202041 8E0"
190 B$="071F3F7F7FFFFFFFFFFF
FF7F7F3F1F07E0F8FCFEFEFFFFFF
FFFFFFFEFEFCF8E0"
200 CALL CHAR(140,A$)
210 CALL CHAR(136,B$)
220 CALL SPRITE(#2,136,14,92
,124)
230 X=88
240 Y=127
250 R=80 :: P=2
260 FOR I=0 TO 2*PI STEP (2*
PI)/18
270 A=X+R*COS(I)
280 B=Y+R*SIN(I)
290 P=P+1
300 CALL SPRITE(#P,136,I*2+3
,A,B,0,0)
310 NEXT I
320 V=28
330 CALL SPRITE(#1,140,16,80
,90)
340 FOR I=0 TO 2*PI STEP (2*
PI)/36
350 A=INT(V*COS(I))
360 B=INT(V*SIN(I))
370 CALL MOTION(#1,A,B)
380 NEXT I
390 IF C=1 THEN 400 ELSE C=C
+1 :: GOTO 340
400 CALL DELSPRITE(#1):: P=1
4 :: V=2 :: C=0
410 FOR I=0 TO 2*PI STEP (2*
PI/18)
420 A=INT(V*COS(I))
430 B=INT(V*SIN(I))
440 P=P+1 :: IF P>20 THEN P=
3
450 CALL MOTION(#P,A,B)
460 V=V*1.05
 rs
470 NEXT I
480 IF C=1 THEN 490 ELSE C=C
+1 :: GOTO 410
490 C=0 :: GOTO 250


############################
#                          #
# The competition winning  #
# programs.  This one is   #
# by William Edmond, and   #
# won the first prize.     #
#                          #
############################


100 REM ** MOTORBIKE **
110 REM **  DEALING  **
120 REM **    BY     **
130 REM ** WILLIAM   **
140 REM **    EDMOND **
150 REM ** T.I.U.P.  **
160 REM ** DATE:     **
170 REM **   15/4/86 **
180 REM ** COMPETION **
190 REM **READ DATA**
200 DIM MODEL$(7),PRICE(7)
210 FOR ELEMENT=1 TO 7
220 READ MODEL$(ELEMENT),PRI
CE(ELEMENT)
230 NEXT ELEMENT
240 CALL CLEAR
250 CALL SCREEN(14)
260 PRINT TAB(7);"MOTOR BIKE
"
270 PRINT TAB(7);" DATABASE
"
280 PRINT
290 PRINT TAB(11);"BY"
300 PRINT TAB(8);"WILLIAM"
310 PRINT TAB(9);"EDMOND"
320 FOR INTERVAL=1 TO 9
330 PRINT
340 NEXT INTERVAL
350 CALL SOUND(2000,-4,5)
360 CALL SOUND(3000,-4,5)
370 CALL SOUND(1700,-8,5)
380 FOR TIME=1 TO 900
390 NEXT TIME
400 CALL CLEAR
410 CALL SCREEN(13)
420 PRINT TAB(7);"SELECTION
LIST"
430 PRINT TAB(7);"=========
====="
440 PRINT TAB(6);"(1)....INS
TRUCTIONS"
450 PRINT TAB(6);"(2)....SEE
 ALL BIKES"
460 PRINT TAB(6);"(3)....SEE
 THE PRICE OF"
470 PRINT TAB(4);"A CERTAIN
MODEL"
480 PRINT TAB(6);"(4)....SEE
 MODELS BELOW"
490 PRINT TAB(4);"A CERTAIN
PRICE"
500 PRINT TAB(6);"(5)....END
"
510 FOR SPACE=1 TO 3
520 PRINT
530 NEXT SPACE
540 CALL SOUND(100,440,6)
550 PRINT "YOUR CHOICE :"
560 CALL KEY(0,K,S)
570 IF S=0 THEN 560
580 IF K=49 THEN 650
590 IF K=50 THEN 860
600 IF K=51 THEN 950
610 IF K=52 THEN 1160
620 IF K=53 THEN 1540
630 GOTO 560
640 REM
650 REM **INSTRUCTIONS**
660 CALL CLEAR
670 CALL SCREEN(4)
680 PRINT TAB(6);"INSTRUCTIO
NS"
690 PRINT TAB(6);"==========
=="
700 PRINT "THIS PROGRAM WILL
 PROVIDE "
710 PRINT "YOU WITH ALL THE
RELEVANT"
720 PRINT "IMFORMATION DURIN
G THE"
730 PRINT "PROGRAM."
740 PRINT "YOU WILL NEED TO
ENTER"
750 PRINT "THE NAMES OF MODE
LS OR "
```

```
*****************************************************************************
*                                                                           *
* Source Code listing of the file DBCODE. Used to provide CALL LINK routines  *
* for the EXTENDED BASIC program called DBPROGRAM. These routines are used to *
* provide additional facilities to XBASIC or to speed up some sections of the *
* program.                                                                  *
*                                                                           *
* Author - Phil West  T.I.U.P.                                              *
*                                                                           *
* Version 1.2                                                               *
*                                                                           *
* Date 18th July 1986                                                       *
*                                                                           *
*****************************************************************************
*
* Define names of CALL LINK routines
*
       DEF  ADD,FIND,MOD,NEW,GET,SORT
       DEF  UP
*
* Extended Basic Equates
*
FAC    EQU  >834A                    Floating Point Accumulator
GPLWS  EQU  >83E0                    GPL interpreter workspace
NUMASG EQU  >2008                    Numeric Assignment entry
NUMREF EQU  >200C                    Numeric Reference entry
STRASG EQU  >2010                    String Reference entry
STRREF EQU  >2014                    String Assignment entry
XMLLNK EQU  >2018                    Extended M/L entry
ERR    EQU  >2034                    Error Routine entry
CFI    EQU  >1288                    Entry address floating point to integer
CIF    EQU  >0020                    Entry address integer to floating point
BASE   EQU  >A000                    Base address of data table
STATUS EQU  >837C                    GPL status byte
MEMFUL EQU  >0B00                    Error code for Memory Full
KEYBRD EQU  >8375                    Address of character read from keyboard

WORKSP BSS  >0020                    Workspace for all routines

SAVRET BSS  >0002                    Buffer used to save return address

BUFFER BSS  >0082                    Buffer for strings from X Basic

COUNT  BSS  >0002                    Buffer for counting # of matches found

PNTR   BSS  >0002                    Buffer for pointer to end of data

LIMIT  BSS  >0002                    Buffer for storing memory limit

SOPNTR BSS  >0002                    Buffer for pointer used by SORT routine

SWPFLG BSS  >0002                    Flag used by SORT routine

MAXLEN BYTE >80                      Maximum length of string from X BASIC
SPACE  BYTE >20                      Space character constant
CHEK   BYTE >08                      Length of string used by FIND routine
UPKEY  BYTE >0B                      FCTN 'E' character constant

       EVEN                          align code to even word boundary

*
* Subroutine to clear the string buffer
*
CLRBUF LI   R8,BUFFER+1              Point to 2nd character of buffer
```

```
760 PRINT "PRICES ALONG THE
WAY"
770 PRINT "AND THIS INFORMAT
ION CAN BE"
780 PRINT "GOT BY PRESSING -
2- IN THE"
790 PRINT "MAIN SELECTION LI
ST."
800 PRINT
810 PRINT
820 PRINT TAB(4);"PRESS ENTE
R TO CONTINUE"
830 CALL KEY(0,K,S)
840 IF S=0 THEN 830
850 GOTO 400
860 REM **ALL BIKES**
870 CALL CLEAR
880 CALL SCREEN(12)
890 PRINT TAB(4);"MODEL
  PRICE"
900 PRINT
910 FOR INDEX=1 TO 7
920 PRINT MODEL$(INDEX),PRIC
E(INDEX)
930 NEXT INDEX
940 GOTO 1360
950 REM **MODEL'S PRICE**
960 CALL CLEAR
970 CALL SCREEN(10)
980 INPUT "MODEL OF BIKE ":B
IKEMODEL$
990 N=0
1000 W=0
1010 FOR N=1 TO 7
1020 IF MODEL$(N)=BIKEMODEL$
 THEN 1110
1030 IF MODEL$(N)=BIKEMODEL$
 THEN 1130
1040 IF MODEL$(N)=BIKEMODEL$
 THEN 1150
1050 IF W<>0 THEN 1150
1060 NEXT N
1070 PRINT
1080 PRINT "MODEL UNLISTED"
1090 GOTO 1150
1100 PRINT
1110 PRINT MODEL$(N),PRICE(N
)
1120 GOTO 1030
1130 W=1
1140 GOTO 1050
1150 GOTO 1360
1160 REM **BELOW A CERTAIN
                PRICE     **

1170 CALL CLEAR
1180 CALL SCREEN(16)
1190 INPUT "WHAT IS THE SEAR
CH PRICE ":PRICETOFIND
1200 Y=0
1210 PRINT
1220 FOR N=1 TO 7
1230 IF PRICE(N)<PRICETOFIND
 THEN 1280
1240 IF PRICE(N)<PRICETOFIND
 THEN 1300
1250 NEXT N
1260 I
```

```
       LI   R9,>0080              Number of bytes to blank out
CLRCHS MOVB @SPACE,*R8+           Blank one character
       DEC  R9                    Decrement counter
       JNE  CLRCHS                If non-zero then continue
       RT                         Return
*
* LINK routine to initialize pointers
*
* No parameters required
*
NEW    MOV  R11,@SAVRET           Save return address
       LWPI WORKSP                Load own workspace
       LI   R1,BASE               Start address of data buffer
       MOV  R1,@PNTR              Reset pointer address
       MOV  @>8386,R1             Get HFA used by X BASIC
       AI   R1,>FFF4              Subtract 10
       MOV  R1,@LIMIT             Store in LIMIT
       JMP  GOOD                  Return to X BASIC
*
* LINK routine to add a new entry to the data buffer
*
* 1st Parameter = record number in file
* 2nd Parameter = the string to be added to buffer
*
ADD    MOV  R11,@SAVRET           Save return address
       LWPI WORKSP                Load own workspace
       BL   @CLRBUF               Clear the string buffer
       CLR  R0                    Simple variable R0=0
       LI   R1,>0002              Get second parameter in LINK list
       BLWP @NUMREF               Get the file record number
       BLWP @XMLLNK               Convert to integer
       DATA CFI
       MOV  @FAC,R5               Copy to R5
       DEC  R1                    now get the string from X BASIC
       LI   R2,BUFFER             Buffer for string
       MOVB @MAXLEN,*R2           Set maximum length string permitted
       BLWP @STRREF               Get the string
       MOV  @PNTR,R3              Get pointer to end of data in memory
       C    R3,@LIMIT             Has the limit been reached ?
       JLT  NOTFUL                If not then continue
       LI   R0,MEMFUL             Memory Full error
       BLWP @ERR                  Issue error and return to X BASIC
NOTFUL LI   R4,>0008              Number of bytes to use from record
       INC  R2                    Point to first byte to move
MOVENT MOVB *R2+,*R3+             Move a byte from string to data buffer
       DEC  R4                    Decrement counter
       JNE  MOVENT                If non-zero then continue
       MOV  R5,*R3+               Put record number of this record in buff
       MOV  R3,@PNTR              Modify data buffer pointer
GOOD   CLR  @STATUS               Clear GPL status byte
       LWPI GPLWS                 Restore GPL workspace
       MOV  @SAVRET,R11           Restore return address
       RT                         Return to X BASIC
*
* LINK routine to modify an existing entry in the data buff
*
* 1st Parameter = record number to be modified
* 2nd Parameter = new value of string to put in data buffer
*
MOD    MOV  R11,@SAVRET           Save return address
       LWPI WORKSP                Load own workspace
       BL   @CLRBUF               Clear the buffer
       CLR  R0                    Simple variable - R0=0
       LI   R1,>0001              First parameter in LINK list
       LI   R2,BUFFER             Buffer for string
       MOVB @MAXLEN,*R2           Set maximum length of string to read
       BLWP @STRREF               Get the string
       INC  R1                    Point to second parameter in LINK list
       BLWP @NUMREF               Get the record number
       BLWP @XMLLNK               Convert to integer
```

```
1270 GOTO 1360
1280 PRINT MODEL$(N),PRICE(N
)
1290 GOTO 1240
1300 Y=1
1310 GOTO 1250
1320 GOTO 1360
1330 PRINT "NO BIKES UNDER T
HAT PRICE"
1340 GOTO 1320
1350 END
1360 REM **SELECTIONS2**
1370 PRINT
1380 PRINT
1390 PRINT TAB(6);"SELECTION
S :"
1400 PRINT TAB(6);"    1)...
.RETURN TO MAIN     MENU"
1410 PRINT TAB(5);"2)....END
"
1420 PRINT
1430 CALL KEY(0,K,S)
1440 IF S=0 THEN 1430
1450 IF K=49 THEN 400
1460 END
1470 DATA "HONDA CB250",1750
1480 DATA "HONDA CB360",1820
1490 DATA "HONDA CB500",2340
1500 DATA "HONDA CB750",2560
1510 DATA "SUZUKI GL900",265
0
1520 DATA "SUZUKI GL1000",27
80
1530 DATA "SUZUKI GL1100",29
40
1540 END
1550 REM **END**
1560 REM IF MORE BIKES ARE
1570 REM TO BE ADDED,REFER
1580 REM TO USER'S MANNUAL
1590 REM FOR INSTRUCTIONS
1600 REM **BYE**
```

```
##############################
#                            #
# Competition entry #2       #
#                            #
# by Norm Bernhardt          #
#                            #
##############################
```

```
100 ! ********************
110 ! *                  *
120 ! *      SLEUTH      *
130 ! *                  *
140 ! ********************
150 !
160 ! BY NORMAN BERNHARDT
170 !   T.I.U.P.
180 !
190 ! ADAPTED FROM VIC-20
200 !
210 !
220 CALL CLEAR
230 CALL SCREEN(5)
240 RANDOMIZE
```

```
       DATA CFI
       MOV  @FAC,R5                    Move to R5
       LI   R4,BASE                    Start of data buffer
SEEK   C    R5,@>000B(R4)              Is this the correct record number ?
       JEQ  GOTNUM                     If yes then modify it
       AI   R4,>000A                   Point to next entry in data buffer
       C    R4,@PNTR                   Reached end of data buffer ?
       JNE  SEEK                       If not then continue
GOTNUM INC  R2                         Point to first character of string buffer
       LI   R3,>0008                   Number of bytes to move
MOVMOD MOVB *R2+,*R4+                  Move new value for this record
       DEC  R3                         Decrement counter
       JNE  MOVMOD                     If non-zero then continue
       JMP  GOOD                       Return to X BASIC
*
* LINK routine to find entries in data buffer
*
* 1st Parameter = string to find in data buffer
* 2nd Parameter = numeric array of records found
* 3rd Parameter = count of Record found - -1 if none found
*
FIND   MOV  R11,@SAVRET                Save return address
       LWPI WORKSP                     Load own workspace
       SETO @COUNT                     Initialize # matches found to -1
       CLR  R0                         Simple variable - R0=0
       LI   R1,>0001                   First parameter in LINK list
       LI   R2,BUFFER                  String buffer
       MOVB @MAXLEN,*R2                Set Maximum length of string to read
       BLWP @STRREF                    Get the string from X BASIC
       CB   *R2,@CHEK                  Is string less than 8 characters long
       JLE  SHORT                      If yes then skip next section
       MOVB @CHEK,*R2                  String >8 chars - only use 8 chars
SHORT  MOVB *R2+,R3                    Get length of search string
       SRL  R3,8                       Right justify
       LI   R5,BASE->000A              Point to start of data buffer-10
FINEX  AI   R5,>000A                   Point to next entry in data buffer
       C    R5,@PNTR                   Reached end of data buffer ?
       JEQ  ENLOOK                     Yes - return
       MOV  R5,R6                      Save data buffer pointer
       MOV  R3,R7                      Save string length
       MOV  R2,R8                      Save pointer to string character
FICOM  CB   *R8+,*R6+                  Compare the strings
       JNE  FINEX                      If not matching then check next entry
       DEC  R7                         Decrement pointer
       JNE  FICOM                      If non-zero then continue
       MOV  @>000B(R5),@FAC            Get the record number of the match
       BLWP @XMLLNK                    Convert to floating point
       DATA CIF
       MOV  @COUNT,R0                  Get # matches found counter
       INC  R0                         Increment by 1
       MOV  R0,@COUNT                  Adjust counter
       LI   R1,>0002                   Second variable in LINK LIST
       BLWP @NUMASG                    Assign record number to array element
       JMP  FINEX                      Keep searching
ENLOOK MOV  @COUNT,@FAC                Store # matches found in FAC
       BLWP @XMLLNK                    Convert to floating point
       DATA CIF
       CLR  R0                         Simple variable - R0=0
       LI   R1,>0003                   Third parameter in LINK list
       BLWP @NUMASG                    Assign number to X BASIC
       B    @GOOD                      Return
*
* LINK routine to sort the data buffer
*
* No parameters
*
SORT   MOV  R11,@SAVRET                Save return address
       LWPI WORKSP                     Load own workspace
       LI   R0,BASE                    Start of data buffer in high memory
       MOV  R0,@SOPNTR                 Store it
       MOV  @PNTR,R9                   Get pointer to end of data
```

```
250 REM DEFINE CHARACTERS
260 CALL CHAR(120,"0000827FD
FF0C0C0")
270 CALL CHAR(121,"000000606
")
280 REM TITLES
290 FOR D=1 TO 8
300 CALL COLOR(D,16,1)
310 NEXT D
320 DISPLAY AT(12,12):"SLEUT
H"
330 DISPLAY AT(14,7):"BY NOR
MAN BERNHARDT"
340 CALL MAGNIFY(2)
350 FOR I=1 TO 3
360 CALL SPRITE(#1,120,2,40,
40,0,0)
370 CALL SOUND(200,-7,2)
380 CALL SPRITE(#2,121,7,40,
50,0,50)
390 CALL COINC(#2,40,200,8,C
O)
400 IF CO<>0 THEN 420
410 GOTO 390
420 NEXT I
430 CALL DELSPRITE(ALL)
440 CALL CLEAR
450 REM INSTRUCTIONS
460 PRINT "          SLEUTH"
470 PRINT
480 PRINT "     YOU ARE A GU
EST STAYING AT THE WINDSOR E
STATE."
490 PRINT
500 PRINT "     WHILE YOU'RE
 ASLEEP    YOU HEAR A GUN SH
OT."
510 PRINT
520 PRINT "     YOU, THE BUT
LER, THE   MAID, COLONEL BUS
H, MISS RI-FFEL AND PROF. OF
FENHEIMER  ALL RACE DOWNSTAI
RS TO FIND MR WINDSOR DEAD D
N THE FLOO-R."
530 PRINT
540 PRINT "     IN HIS CHEST
 IS A      BULLET."
550 PRINT
560 PRINT
570 PRINT " PRESS ANY KEY TO
 CONTINUE"
580 CALL KEY(0,K,S)
590 IF S=0 THEN 580
600 CALL CLEAR
610 PRINT "     BEING A DETE
CTIVE, YOU NATURALLY WANT TO
 SOLVE THE MYSTERY OF THE KI
LLING OF MR WINDSOR. "
620 PRINT
630 PRINT "YOU ALSO HAVE TO
GUESS THE  LIAR"
640 PRINT
650 PRINT "     YOU ASK EVER
YONE IN THE ROOM QUESTIONS A
ND THESE   ARE THE ANSWERS Y
OU GOT."
660 PRINT
670 PRINT " PRESS ANY KEY TO
```

```
       AI   R9,>FFF6                  subtract 10 to point to last entry
SORT1  LI   R0,BASE                   Start of data buffer
       CLR  @SWPFLG                   Clear flag
SORT2  MOV  R0,R1                     Copy R0
       AI   R1,>000A                  Add 10 to copy
       MOV  R0,R2                     Save R0
       MOV  R0,R5                     And again
       MOV  R1,R3                     Save R1
       MOV  R1,R6                     And again
       LI   R4,>0008                  Number of characters to compare
SWAP2  CB   *R2+,*R3+                 Compare 2 entries
       JH   SWAP3                     1st entry > next entry - swap them
       JL   NOSWAP                    Entries in correct order
       DEC  R4                        Decrement counter
       JNE  SWAP2                     If non-zero then continue
       JMP  NOSWAP                    Entries are equal - don't swap
SWAP3  LI   R4,>0005                  # of words to swap
SWAP4  MOV  *R5,R2                    Store a word from 1st entry
       MOV  *R6,R3                    Store a word from 2nd entry
       MOV  R3,*R5+                   Swap them
       MOV  R2,*R6+                   Swap them
       DEC  R4                        Decrement counter
       JNE  SWAP4                     If non-zero then continue
       SETO @SWPFLG                   Set flag to indicate exchange performed
NOSWAP AI   R0,>000A                  Increment data buffer pointer
       C    R1,R9                     Finished this pass ?
       JNE  SORT2                     No - keep sorting
       MOV  @SWPFLG,@SWPFLG           Any swaps this pass ?
       JNE  SORT1                     Yes - do another pass
       B    @GOOD                     Sort finished - return to X BASIC
*
* LINK routine to get record numbers in sorted order
*
* 1st parameter=next record number , -1 if end of data
*
GET    MOV  R11,@SAVRET               Save return address
       LWPI WORKSP                    Load own workspace
       CLR  R0                        Simple variable - R0=0
       LI   R1,>0001                  Only parameter in LINK list
       MOV  @SOPNTR,R2                Get pointer to next entry in data buffer
       C    R2,@PNTR                  Reached end of data yet ?
       JEQ  GETEND                    Yes - return -1
       MOV  @>0008(R2),@FAC           Get next record number
GETASG BLWP @XMLLNK                   Convert to floating point
       DATA CIF
       BLWP @NUMASG                   Assign value to X BASIC
       AI   R2,>000A                  Point to next data entry
       MOV  R2,@SOPNTR                Store it
       B    @GOOD                     Return
GETEND SETO @FAC                      Store -1 in FAC
       JMP  GETASG                    Assign value to X BASIC
*
* LINK routine to determine key pressed to exit an ACCEPT AT statement
*
* 1st Parameter=0 if enter of FCTN X pressed,-1 if FCTN E  pressed
*
UP     MOV  R11,@SAVRET               Save return address
       LWPI WORKSP                    Load own workspace
       CLR  @FAC                      Clear at FAC
       CB   @KEYBRD,@UPKEY            Was the FCTN E key pressed ?
       JNE  UP1                       No - return a value of 0
       SETO @FAC                      FCTN E pressed - return -1
UP1    BLWP @XMLLNK                   Convert to floating point
       DATA CIF
       CLR  R0                        Simple variable R0=0
       LI   R1,>0001                  Only parameter in LINK list
       BLWP @NUMASG                   Assign value to X BASIC
       B    @GOOD                     Return
*
* End of source listing
*
       END
```

```
 CONTINUE"
680 CALL KEY(0,K,S)
690 IF S=0 THEN 680
700 CALL CLEAR
710 PRINT "    YOU ALREADY K
NOW YOU    DIDN'T DO IT"
720 PRINT
730 DEF M=INT(RND*5)+1
740 IF M=1 THEN 790
750 IF M=2 THEN 860
760 IF M=3 THEN 930
770 IF M=4 THEN 1000
780 IF M=5 THEN 1070
790 REM BUTLER DID IT
800 M$="BUTLER"
810 A$="MISS RIFFEL"
820 B$="PROF. OFFENHEIMER"
830 C$="THE MAID"
840 D$="COLONEL BUSH"
850 GOTO 1130
860 REM MISS RIFFEL DID IT
870 M$="MISS RIFFEL"
880 A$="BUTLER"
890 B$="PROF. OFFENHEIMER"
900 C$="THE MAID"
910 D$="COLONEL BUSH "
920 GOTO 1130
930 REM PROF. DID IT
940 M$="PROF. OFFEMHEIMER"
950 A$="THE MAID"
960 B$="THE BUTLER"
970 C$="COLONEL BUSH"
980 D$="MISS RIFFEL"
990 GOTO 1130
1000 REM COLONEL DUNNIT
1010 M$="COLONEL BUSH"
1020 A$="MISS RIFFEL"
1030 B$="THE BUTLER"
1040 C$="PROF.  OFFENHEIMER"
1050 D$="THE MAID"
1060 GOTO 1130
1070 REM THE MAID DUNNIT
1080 M$="MAID"
1090 A$="MISS RIFFEL"
1100 B$="PROF. OFFENHEIMER"
1110 C$="THE BUTLER"
1120 D$="COLONEL BUSH"
1130 PRINT ;A$;" SAID THAT T
HEY KNEW THAT ";B$;" HAD A G
UN."
1140 FOR I=1 TO 1000
1150 NEXT I
1160 PRINT
1170 PRINT ;B$;" ADMITTED TH
AT THEY HAD A GUN, BUT THEY
SAID IT WAS STOLEN THE NIGHT
 BEFORE THE MURDER."
1180 FOR I=1 TO 1000
1190 NEXT I
1200 PRINT ;M$;" SAID THAT T
HEY SAW ";C$;" TAKING THE GU
N."
1210 FOR I=1 TO 1000
1220 NEXT I
1230 PRINT
1240 PRINT ;C$;" DENIED TAKI
NG THE GUN AND SAYS THAT HE
HEARD ";D$;"'S BEDROOM DOOR
```

```
SLAM SHUT SECONDS AFTER THE
GUN SHOT."
1250 FOR I=1 TO 1000
1260 NEXT I
1270 PRINT
1280 PRINT ;D$;"CLAIMS THAT
HE WENT TO GET A GLASS OF WA
TER."
1290 FOR I=1 TO 1000
1300 NEXT I
1310 PRINT
1320 PRINT ;D$;" ALSO SAID T
HEY SAW ";M$;" IN THE CORRID
OR BEFORE THE GUNSHOT."
1330 FOR I=1 TO 1000
1340 NEXT I
1350 PRINT
1360 PRINT
1370 PRINT "THOSE ARE ALL TH
E STATEMENTS YOU RECIEVED."
1380 PRINT
1390 PRINT "CAN YOU SOLVE TH
E MYSTERY? "
1400 PRINT "WHODUNNIT"
1410 INPUT J$
1420 IF J$=M$ THEN 1520
1430 CALL CLEAR
1440 PRINT ;M$;" HAS ACCUSED
 YOU OF BEING   THE MURDERER
 AND PLAYING    THE DETECTIV
E PART TO COVER UP YOUR TRAC
KS, THEY ALL AG-REE AND YOU
ARE FRAMED "
1450 PRINT "FOR THE MURDER O
F MR WINDSO-R. YOU RECIEVE 5
0 YEARS IN JAIL."
1460 PRINT "WHILE IN JAIL YO
U REALIZE Y-OU'LL BE 101 YEA
RS OLD WHEN YOU GET OUT."
1470 PRINT "DO YOU WANT TO S
OLVE ANOTHER MYSTERY?"
1480 INPUT O$
1490 IF O$="Y" THEN 650
1500 END
1510 REM THEY GET IT
1520 PRINT "HOW DID YOU KNOW
 ";M$;" DUNNIT."
1530 PRINT "NOW WHO WAS THE
LIAR?'
1540 INPUT Y$
1550 IF Y$=M$ THEN 1590
1560 PRINT ;M$;" TURNS AROUN
D AND SHOOTS YOU WITH....."
1570 FOR I=1 TO 1500
1580 NEXT I
1590 REM THEY GET THE LIAR
1600 PRINT ;M$;" SAYS THAT H
E IS GOING TO GET YOU FOR TH
IS AND IS CARRIED OUT BY THE
 POLICE."
1610 END


##############################
#                            #
# Competition entry #4       #
#                            #
# by  Steve Summerell        #
#                            #
##############################


100 REM BY STEVE SUMMERELL
110 REM  17TH MARCH 1986
120 REM MEMBER OF T.I.U.P.
130 REM ^^^^^TI BASIC^^^^^
140 REM
150 CALL CLEAR
160 CALL SCREEN(5)
170 CALL COLOR(1,16,5)
180 CALL COLOR(2,16,5)
190 CALL COLOR(3,16,5)
200 CALL COLOR(4,16,5)
210 CALL COLOR(5,16,5)
220 CALL COLOR(6,16,5)
230 CALL COLOR(7,16,5)
240 CALL COLOR(8,16,5)
250 GOSUB 530
260 FOR A=1 TO 1000
270 NEXT A
280 PRINT : : : : : : : : :
: : :
290 INPUT " WHOSE PHONE NUMB
ER DO YOU   WANT ?":A$
300 RESTORE
310 FOR SEARCH=1 TO 6
320 READ NAME$,TELPHN$
330 IF NAME$=A$ THEN 390
340 NEXT SEARCH
350 PRINT : : : :
360 PRINT " SORRY !! I DON'T
 HAVE ";A$;"'S PHONE NUMBER.
"
370 PRINT : : : :
380 GOTO 430
390 PRINT : : : :
400 PRINT A$;"'S PHONE NUMBE
R IS "
410 PRINT TAB(10);TELPHN$
420 PRINT : : : :
430 INPUT " DO YOU WANT ANOT
HER PHONE   NUMBER ? (Y/N)":
B$
440 PRINT : : : :
450 IF B$="Y" THEN 290
460 STOP
470 DATA PHIL WEST,398-2079
480 DATA D.CLARK,277-5296
490 DATA B.ELSNER,271-8642
500 DATA LES TWISS,453-6837
510 DATA GARDEREN,362-4779
520 DATA WILKINSON,398-2079
530 CALL CHAR(97,"0000000000
010307")
540 CALL CHAR(98,"00003F7FFF
FFFFFF")
550 CALL CHAR(99,"0000FFFFFF
FF8F6F")
560 CALL CHAR(100,"0000FFFFF
FFFF1F6")
570 CALL CHAR(101,"0000FCFEF
FFFFFFF")
580 CALL CHAR(102,"000000000
080C0E0")
590 CALL CHAR(103,"070F1F1F0
0000000")
600 CALL CHAR(104,"FFFEFEFD0
1030307")
610 CALL CHAR(105,"FFFFFCF3E
FEFDFDF")
620 CALL CHAR(106,"FFFF3FCFF
7F7FBFB")
630 CALL CHAR(107,"FF7F7FBF8
0C0C0E0")
640 CALL CHAR(108,"E0F0F8F80
0000000")
650 CALL CHAR(109,"070F0F1F1
F3F3F7F")
660 CALL CHAR(110,"DFDFEFEFF
3FCFFFF")
670 CALL CHAR(111,"FBFBF7F7C
F3FFFFF")
680 CALL CHAR(112,"E0F0F0F8F
8FCFCFE")
690 PRINT "******PHONE DIREC
TORY******"
700 PRINT : :
710 CALL COLOR(9,9,5)
720 CALL COLOR(10,9,5)
730 CALL COLOR(11,9,5)
740 PRINT "          abcdef
                   ghijkl
                    mnop"
750 PRINT : : : :"
760 RETURN
950 A$="THE MAID"
960 B$="THE BUTLER"
970 C$="COLONEL BUSH"
980 D$="MISS PIFFEL"
```

FOR SALE

HELP I need to sell a "CENTRONICS" model 300 line printer, this is a good solid workhorse printer that has done me well. The only reason it has to go is so that my new EPSON FX86 can be paid for. If you would like to give this printer a good home here are a few details.

120 characters per second.
Draft and expanded character modes only.
80 characters wide and six lines per inch.
Bullet proof.
Solid as a Sherman tank.
Parallel interface only.

This piece of genuinly solid piece of equipment that is in good working order can be yours if you offer Geoff Warner a spectacular low figure of around the $120.00 mark.

For more details phone Geoff on (09) 291-7[illegible].