HUNTER VALLEY 99ERS USERS GROUP

HOME COMPUTER NEWSLETTER

East Maitland "from the *Picturesque Atlas of Australasia*, 1886.

AUGUST 1989

"Main Street, Singleton."
1886.

RAILWAY STATION

NEW BATTERY

SAILORS HOME

"High Street, West Maitland from the *Picturesque Atlas of Australasia* 1886.

REGISTERED BY AUST POST
PUBLICATION No. NBG8023

Hunter Valley 99ERS USER GROUP Newcastle

YOUR COMMITTEE

PRESIDENT

Peter Smith
8 Glebe St.,
EAST MAITLAND 2322
Phone 336164 V/tl 49336164Ø

VICE PRESIDENT

John Paton
1 Parien Close,
RUTHERFORD 232Ø
Phone 326Ø14 V/tl 49326Ø14Ø

SECRETARY

Brian Woods
9 Thirlmere Pde.,
TARRO 2322
Phone 6623Ø7 V/tl 4966723Ø7Ø

TREASURER

Noel Cavanagh
378 Morpeth Rd.,
MORPETH 2321
Phone 333764

Software Librarian

Stewart Bradley
14 Hughes St.,
BIRMINGHAM GARDENS 2287
Phone 513246

EDITOR

Allen (Joe) Wright
77 Andrew Rd.,
VALENTINE 228Ø
Phone 46812Ø

PURCHASING/HARDWARE CO-ORDINATOR

Alan Franks
822 Pacific Highway,
MARKS POINT 228Ø
Phone 45917Ø

SOCIAL SECRETARY

Robert (Bob) MacClure
75 Deborah St.,
KOTARA SOUTH
Phone 437431

PUBLICATIONS LIBRARIAN

Ken Lynch
9 Hall St.,
EDGEWORTH 2285
Phone 585983

COMMITTEE MEMBERS

Don Dorrington
36 NELSON St.,
BARNSLEY 23Ø1
Phone 531228

Tim Watkins
36 The Ridgeway,
BOLTON POINT 2283
Phone 592836

CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.

Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX 8Ø or DIS/VAR 8Ø). A suitable Public Domain word processor program will be supplied if required by the club librarian.

Please include along with your article sufficient information to enable the file to be read by the Editor eg. File Name etc. The preferred format is 35 columns and page length 66 lines, right justified.

All articles printed in HV99 NEWS (unless notified otherwise) are considered to be Public Domain. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.

Articles for publication can be submitted to the Editor, ALL other club related correspondence should be addressed to The Secretary.

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY NINE USER GROUP.

Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result of applying such information.

The views expressed in the articles in this publication are the views of the author/s and are not necessarily the views of the Committee, Editor or members.

TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.

HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS.

From President PETE'S *Quill*

A profitable and friendly committee meeting was held at my place on 8/8/89 and the following points emerged...

1)The proposed visit to the **"Black Hole"** at Grawin, has been postponed till next year sometime (we hope) so start preparing and saving your money now.

2)The club is to purchase a pair of 1/2 ht 2s/2d drives for the librarian and his old ones to go to the treasurer.

3)The "Christmas Do" night has been changed and will consist of a 4 course dinner and fine live theatre on Sat. 18/11/89. A deposit of $15.00/head is required by Bob before 26/9/89 meeting.

4)Tony McGovern has developed a "challenge" to all budding EA users (and others). See his article elsewhere in this mag. Thanks Tony, a good idea which may be emulated by other language groups.

apologies go to those club members who had to wait so long for their magazines. I know there is nothing we can say which will fix the situation, however rest assured that all efforts will be made to ensure that it doesn't happen again and you can look forward to regularly reading this fine magazine once-a-month again.

Congratulations to Ruth and Noel Cavanagh on the birth of their third child , a lovely little girl.

At present we have only 24 members who have re-joined/joined for this year. We know that things have been busy but we do need **YOU** to rejoin and help us to help others. (and have a bit of fun as well).

I forgot to get personal resumes from those committee members who did not feature last month so you can wait with baited breath for next month.

The articles in Micropendium detailing the development of a PORTABLE 4A with ram disks and memory sure fires up the imagination. Have we any readers in our midst who would care to describe some project, possibly a little less ambitious? Please feel free to contact the editor and pass on the word.

SOFTWARE LIBRARY CATALOGUE.

About a year ago, John Paton,Brian Wood, Albert Anderson and I set out on an ambitious task to make the contents of our club's library more accessible to members. We reasoned that this ability would allow members to be more productive, better informed and equipped and happier to be in and use the club.

---Surely admirable ideals..but what a job???!!!

We set to work and decided that a data-base would be a good way of maintaining an up-to-date record, however there were a few provisors.

(1)Members would **not** need to have, or learn how to run, a data-base to have access to Records.

(2)The actual names of the programs as they appear when the program is run would be indexed and not just a file-name (as appears in a disk catalogue)

(3)The list must be "printable" hence limited to 80 chars.

(4)Information regarding how to load the program should be available

(5)A rough indication of the type of program should be indicated.

(6)All information must be able to be sorted and classified.

(7)Data must be able to be stored even if the constraints of disk-capacity are exceeded.(Multiple disks used to store data).

Having used a few data-bases previously we were in a reasonable position to select a suitable one and so we chose **PRB** as it allowed all the above facilities.

I designed a record format which would contain the following information..

1.Program name (15 chars). Actual name of running program.

2.Disk number (4 chars). Indicates the disk number in the club's library, on which the program can be found.

3.Type (5 chars). Indicates the following..

D/B (data-base); **W/P** (word

processor); **ASS** (assembler of some sort); **ART** (art/drawing program); **RTDAT** (data for picture in ART program); **GAME** ; **MODM** (communication program); **ADV** (adventure program); **MUSIC** (music prog or data for same); **COPY** (copying or back-up program); **SPRDS** (spreadsheet program or data); **DMAN** (disk manager/filer etc program or data); **UTILS** (any utility program).

NOTE Many programs fit in a couple of categories so selection was made on anticipated major area of usage. This is a personal assessment made by the person reviewing that program and so is a personal interpretation and may be open to debate. This is unfortunate, however without complicating the system unduly, the compromise selected should be sufficiently specific to guide members towards their required program.

4.Main file name (1Ø chars). Main file-name as appears on catalogue. If the files for a program are archived the name of the archived file is provided. (Generally when you request a program from the librarian, you will be provided with the **archived** program as you will be sure of having all (and only) those files necessary.

5.Number of files. (2 chars) If archived will be 1, if otherwise will indicate how many files needed to run the program.

6.Loading environment. (3 chars) Indicates the environment necessary to run the program. Following options are catered for..

B (basic); **XB** (Extended Basic); **MM** (mini memory); **EA3** (option 3 of editor assembler or a **DF8Ø** file able to be run by option 4 of F/Web loaders.); **EA5** (option 5 of editor assembler or a program file able to be run from option 3 of F/Web loaders.); **MP** (multiplan sheet); **TIW** (any DV8Ø file able to be loaded into TI Writer or F/Web Editor)

7.Comments. (22 chars) Indicates any special requirements and intent of program. may even indicate worth with a grade of from 1(good) to 3(poor) as char 22.

8.Distribution. (2 chars). P(ublic Domain); F(airware); C(opy-write)

9.Archived. Indicates if archived.

Generally these fields should provide enough information so that you can find a disk in the library which contains a program which **may** be the one you desire. You can then run that program and if desired (and subject to any copy-write or fairware constraints) copy it.

The job of going through all the disks in our library was too big for we few so thanks now go to those members who assisted at the last meeting and took home disks to examine.

The product of our labour will be incorporated into the data-base and all members will be supplied with a printed copy of the same, sorted on **"TYPE"** .

If required, members can have a disk of the files for examination using TIWriter or a copy of the actual files for manipulation with PRB.

Now here is a short programme for you to type in. Maybe somebody will work on it and make some needed improvements in screen presentation, ease of use and comments. If you do take up this challenge I am sure that Joe will be only to happy to reproduce your efforts in our Newsletter.

```
100 !SIMPLE LITTLE PROGRAM TO WORK
    OUT THE AVERAGE SCORE OF
    MULTI-ITEM TESTS (ALL SAME
    NUMBER OF ITEMS)
11Ø !TO END ENTERING DATA SIMPLY
    ENTER AN "E" AND PRESS ENTER
12Ø !TO SET THE ITEMS IN THE TEST
    RESPOND TO THE PROMPTS
13Ø CALL CLEAR :: INPUT
    "No. items/test":NO
14Ø CALL CLEAR
15Ø C=Ø :: TOTAL=Ø :: SCORE$=""
16Ø INPUT "SCORE=":SCORE$
17Ø IF SCORE$="E" OR SCORE$="e"
    THEN 24Ø
18Ø IF ASC(SCORE$)>57 OR
    ASC(SCORE$)<47 THEN 16Ø
19Ø IF VAL(SCORE$)>NO OR
    VAL(SCORE$)<Ø THEN 16Ø
2ØØ C=C+1
21Ø TOTAL=TOTAL+VAL(SCORE$)
22Ø PRINT "PROGRESSIVE TOTAL =";TOTAL
23Ø GOTO 16Ø
24Ø AV=INT((TOTALØ/(C*NO))+.5)
25Ø CALL CLEAR :: PRINT
   "TOTAL=";TOTAL;" TALLY=";C ::
    PRINT " AVERAGE=";AV ;"%"
26Ø PRINT
    "(A)nother,(R)estat,(E)nd ?"
27Ø CALL KEY(Ø,K,S)::
    IF S=O THEN 27Ø
28Ø IF K=69 OR K=1Ø1 THEN END
29Ø IF K=65 OR K=97 THEN 14Ø
3ØØ IF K=82 OR K=114 THEN 13Ø
31Ø GOTO 27Ø
```

Regards!

Pete.

Brian Woods REPORTS From the Secretary's ** Desk **

"I know that you believe you understand what you think I said, but I'm not sure you realise that what you heard is not what I meant."

source unknown

THE NEW COMMITTEE

The first meeting of the new committee took place on the 11th July, and already new ideas have come forth and a program of social activities roughed out. The full list of the new committee members appears on the inside front cover of the newsletter. Don't forget, if you have any ideas that may interest others in the group pass them on to a committee member so that it can be discussed and if suitable, arranged - it's your group so do your part in its running.

MEMBERSHIP DUES

Membership dues are now payable either by mail to the Secretary or to the Treasurer at the monthly meeting. Please return your membership forms as soon as possible, along with your money.

BOB CARMANY

One of our most active members, Bob Carmany from the USA is planning to visit us down under, probably February, so stay tuned for further details as Bob keeps us in touch. He is very keen to come along to one of the monthly meetings to meet as many HV99ers as possible, as well as seeing something of the area itself.

MAGAZINE ARTICLES

For the time being all articles for the newsletter should be given (or sent) to Joe Wright, who has volunteered to serve as Editor this year. Like our last Editor, Joe is always on the look-out for articles from our members.

OTHER COMPUTERS

It was mentioned at the committee meeting that some of our members have purchased other types of computers, and discussed the possibility of establishing a `register' of the owners of these computers. This will be made available to the members of the group so that they can get together and discuss their `other' machine. If you have a computer other than a TI get in touch with the Secretary and arrangements will be made to put you in touch with `like souls'. Please note - this IS NOT meant to undermine the basic reason for the group's existence - the support of the TI brand of computers!!

SOCIAL ACTIVITIES

Our `Social Secretary', Bob MacClure is in the process of arranging various activities for members & their families. One such activity is a Christmas party to be held at a local Theatre Restaurant on Saturday 18th November at a cost of $32 a head which includes a 4 course meal and the Show afterwards. If you are interested in this outing let Bob know as soon as possible.

If you have any ideas regarding a Christmas party for the kids, let someone on the Committee know.

TONY McGOVERN

Tony is currently fiddling with an E/A program, details of which appear in this month's Newsletter. Tony challenges any E/A programmers in the group to modify the program to run more efficiently.

Every entry in this `contest' gets a disk full of Assembly Language tutorials & the winning entry (to be judged by Tony) will win a super prize (to be announced).

SEPTEMBER RAFFLE

At the September meeting a Mini Memory Module donated by Joe Wright, a Parsec Module & one other Module will be raffled. All financial members will be in the draw - you do not have to be at the meeting to win - just make sure your dues are paid by then!

GENEALOGY RECORD KEEPER

No folks, Joe has not given up on this project - he has been flat out doing other things for the Group. Joe reports that the latest version should be released in a couple of months, incorporating many of the features that have been suggested by TIers from all over the world. His next step (after release) will be to translate the whole program to IBM format, so even those people will be able to use software developed on the TI!

BOOT PROGRAM

It was reported by Gary Cox in the July issue of TIdbits (Memphis UG) that the Miami UG is offering a program called BOOT which allows users to use the John Johnson's RAMdisk menu program without having a RAMdisk. the price is $US7.95 & is available from

Yvette McKenzie
6775 Tamiami Canal Road
Miami FL 33126

HOT BUG

Gary Taylor, writing in the May issue of the Pittsburgh UG Newsletter reports:

"I talked with Charles Earl at the Faire (Ottawa) about Press. He said that he is working hard on completing the program. he has had to develop some debugging tools of his own during the development of Press and decided to release "Hot Bug" as fairware. It is a new `pop-up' debugger offering step or realtime execution of programs. It comes complete with a Hex oriented calculator & will support remote debugging from another TI! It will load into a Supercart or Gram Kracker leaving a full 32K for your program.

It has a Fairware price tag of $20 and can be purchased from :

Charles Earl
34 McLeod Street
Ottawa, Ontario, Canada
K2P-0Z5

IF YOU WERE BORN BEFORE 1950

This came from the August issue of the Guilford 99er Newsletter, the author's name not being mentioned - perhaps to disguise his/her age??

We were born before TV, before penicillin. polio shots, frozen foods, Xerox, plastic, contact lenses, Frisbees and THE PILL.

We were born before radar, credit cards, split atoms, laser beams, ballpoint pens... before pantyhose, dishwashers, clothes dryers, electric blankets, air conditioners, drip-dry clothes and before man walked on the moon. We got married first and then lived together. How quaint can you get?

In our time, closets were for clothes, not for coming out of, Bunnies were small rabbits and rabbits were not Volkswagons. Designer jeans were scheming girls named Jeanne or Jean, and having a meaningful relationship meant getting along with our cousins.

We thought fast food was what you ate during Lent. We were before house-husbands, gay rights, computer dating, dual careers and computer marriages. We were before day-care centres, group therapy and nursing homes. We never heard of FM radio, tape decks, electric typewriters, artificial hearts, word processors, yogurt, and guys wearing earrings. For us, time-sharing meant togetherness - not computers or condominiums. A chip was a piece of wood, hardware was hardware and software wasn't even a word!

In 1940 making out referred to how you did on your exam... Pizzas, MacDonalds and instant coffee were unheard of. We hit the scene when there were 5 & 10 cent stores where you bought things for a nickel & a dime. Ice cream cones were 5 cents and for that price you could ride a street car, make a phone call, buy a Pepsi or enough stamps to mail one letter and 2 postcards. You could buy a Chevy coupe for $600, but who could afford one? A pity too, because gas was 11 cents a gallon.

In our day grass was mowed, Coke was a cold drink, pot was something you cooked in, rock music was Grandma's lullaby and AIDS were helpers in the Principals office. We were certainly not before the difference between the sexes was discovered, but we were surely before the sex change... we made do with what we had and we were the last generation that was so dumb as to think you had to have a husband to have a baby.

No wonder we are so confused and there is such a generation gap today... BUT WE SURVIVED!!!

ASSEMBLY SQUISH

TONY McGOVERN

This article poses a little puzzle and challenge for those HV99 members who know or are learning assembly language (the only way to fly on the TI-99). Perhaps there will be a small prize for the best successful attempt, though the main prize will the eternal glory from being recorded in these pages as the winner. Well ... it may not be precisely eternal, but it will do for as long as the TI-99 keeps going enough to have a HV99 UG interested in it.

The background to the problem is found in the SQUISH routine of the buffer manager in TI-Writer. With all the ROM space assigned in the TI-99's memory map, there isn't all that much RAM available, only 32K in the normal machine, and the Editor and all its text has to be squeezed in. One way TI-Writer does this is by squishing the text held in the memory buffer into a run length encoded form. A line number table grows down from the top of high memory, each entry being a pointer to the start of a squished record kept in a pile growing upwards from above the buffer manager code at the bottom of hi-mem (>A41Ø is the usual start). The records may be in any physical order in this pile and no gaps are ever left in it. Records are always added at the top, and if any are deleted the whole pile above that is dropped down.

We aren't going to attack all of that problem, even though TI's code could stand a lot of improvements if only there were a bunch more bytes to spare. Atrax R. is working on it. The immediate challenge or exercise is for you to see if you can do better than TI did in writing the SQUISH routine itself.

So what is it and what does it do ? It is a BL routine called from a couple of places that is given a pointer to a 8Ø character raw record (no length byte, just the 8Ø chars including all the blanks out to the end) in CPU RAM somewhere, and returns the squished version in a standard buffer area. The raw record may be the result of keyboard entry, or else a record read in from disk by LF. Records are stored in unsquished form on disk in DV/8Ø files and so may be shorter than 8Ø characters, in which case the raw record buffer is padded out with blanks to the full 8Ø bytes when the record is read back in.

A squished record consists of a length byte for the record which includes the length byte itself, followed by the line in an RLE form. Characters which are not repeated are stored just as they are, but repeated characters are followed by a byte containing the number of characters in that particular run of repeats. How does the unsquisher know a character is to be repeated ? SQUISH flags this by setting the most significant bit in repeated characters. As an example a repeated space becomes >AØ instead of >2Ø for a single space. This works because the normal ASCII characters accepted by the editor do not have the MS bit set. The SQUISH routine has to guard against full 8-bit bytes being slipped in from external disk records by stripping off the MS bit from all characters, otherwise the unsquisher could get dangerously confused.

A totally blank line encodes to the hex bytes

>Ø3,>AØ,>5Ø

A line consisting of A BB .. starting at the far left in column zero is stored in the text buffer as

>Ø7,>41,>2Ø,>C2,>Ø2,>AØ,>4C

You can see the sort of thing that happens from these examples . The squish buffer can accomodate the maximum possible total length of 81 bytes. This isn't necessarily the only way to do things, but it is the TI-Writer way so we will stick to it for present purposes.

The challenge then is to write a SQUISH routine that can replace TI's original version, but to do it better than TI did by making it shorter. SQUISH expects the pointer to the raw record 8Ø character buffer to be held at address BUFADR, and that the squish buffer starts at SQSBUF. To match the TI routine the following restrictions MUST be observed.

(1) The routine is called by BL **SQUISH**

(2) On exit R3 must point to SQSBUF.

(3) On exit R6 must contain the length byte of the squished record, right adjusted as a word.

(4) Registers RØ,R3,R6,R7,R8,R9,R1Ø,R12 are available for use. Contents of other registers must be preserved.

(5) If needed the data word >8ØØØ is available at address H8ØØØ.

(6) The MS bit must be stripped from all characters from the raw record.

(7) The maximum length allowed for the body of the routine is 45 words (>5A bytes).

As a matter of convenience for evaluation and comparison use R8 for your index into the raw record buffer and R9 as your index into the squish buffer. The assembly source will then have the form

```
SFIRST EQU  >AØØØ
SLAST  EQU   see text below
       DEF   SFIRST,SLAST
BUFADR EQU  >2Ø5E
SQSBUF EQU  >A198
H8ØØØ  EQU  >A1EE

       AORG >A22A

SQUISH EQU  $
       RT
```

Your code should be fully commented of course so that the judges don't have to strain their little brains too hard. Entries should also include a pseudo-code description of the program logic, and in the event of a tie on length achieved the decider will be quality of commenting in the source code. As a goal to shoot for, Atrax R. had it down to 38 words with no external data needed.

The SFIRST and SLAST EQUates and DEFs are included to allow you to test your code. To determine SLAST use DPATCH or other sector reader to inspect the first sector of the EE program file in your Funnelweb system. The second word is the file length and should be added to >AØØØ to get a value for SLAST. For instance on the AVPC version of EE I'm using now this length is >1422 so that SLAST would be >B422. The normal 4Ø-col version is a bit shorter. So the suggested procedure is;

(1) Make up a working disk containing FW (or as UTIL1) configured for the drive you will have it in, and Editor files ED,EE. Your Funnelweb system disk itself should have LL as part of the usual system files, and FMSAVE should be available. It might be a good idea to put file protection on your ED,EE system files on your Funnelweb disk to prevent possible accidents.

(2) Write the source code for your SQUISH routine using the Funnelweb Programmer's Editor, and check its length by counting off the words in the source code.

(3) Assemble this code. If there are any errors go back to (2) and make the necessary corrections.

(4) Go to the Funnelweb Loaders screen and choose program file loader, Option 2.

(5) Load the EE file from your Funnelweb system disk. It will return immediately to the Loaders screen.

(6) Immediately select Option 6, Low-Loader and load the object file you created in part(2). If the name of your object file doesn't come up automatically, then it is high time to update your Funnelweb system. Then load FMSAVE. If you make any blunders start at (5) again.

(7) <Enter> to go to the DEF display and select the SAVE entry. SAVE say DSKx.EE where "x" is the working drive #. If you have only a single drive remember to switch disks. z

(8) Go to Loaders Option 2 and load FW (or as UTIL1) from your working disk. Then load the Editor ....

If all has been successful it will run normally. If not - back to the drawing board. This article will appear in the August 89 issue, and closing date for entries will be the last Tuesday in October to allow time for out of town members to participate. The winner will be announced at the November meeting, the last of the year. Atrax R. will be the judge. If there is at least a reasonable degree of interest in this kind of activity there will be future problems posed as little contests in the future. Perhaps one of our XB programmers might care to put up a similar kind of challenge/puzzle, say to write a sub-program to do some clearly delineated task with a host program to merge it into for test, so that a wider range of members can participate.

Happy coding !!!

FUNNEL WEB (Female): Approximately 30mm in body length. Black to very dark brown in colour. Each egg sac may contain up to 120 spiderlings. Constructs burrow in moist soil under houses, in rockeries, compost heaps etc. Identification point: Very similar in appearance, but more robust than male; has no spur on second front legs. Extremely venomous.

FEMALE FUNNEL WEB

SPIDERS

AUSTRALIAN SPIDERS: The majority of spiders in Australia are relatively harmless although most can inflict a painful bite which can cause infection. The venomous spiders (those which can cause death) are the Funnel Web (male and female) of which the male is the most deadly, and the Red Back Spider.

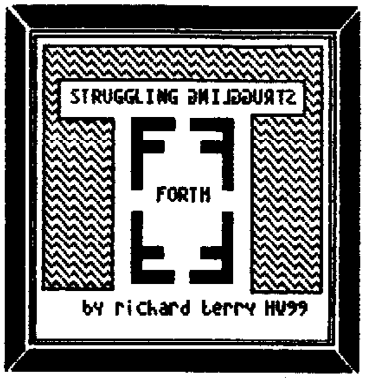
SCR #2

STRUGGLING FORTH

HV99ERS AUGUST ARTICLE 89

Program concepts

On Forth Possums Chardonnay and Hesse



As I write this, and I mean write as I don't have a computer at home, I'm looking out across the timber decking, over the tree canopy which follows the v-shaped line of the ridge, towards the distant hillside of the next valley where a few lights are beginning to twinkle through the evening winter mist. Mozarts Mass in C Minor is playing in the background.I'm sure his music is the nearest mortal **expression** of nearness to God. My silver grey persian cat is lying spreadeagled on her back in front of the heater, and my glass of '86 chardonnay is half finished. I've been fortunate to have recently moved from the concrete jungle of the inner city to more verdant surroundings. Any time now I'll be visited by some of the denizens of my new domain -nicer by far than the arachnids of Funnelweb Farm- my very own friendly neighbourhood brush tail possums (by 3). Cute as hell and seem to behave just like the Ewoks.

All very nice I hear you chorus, but what has all this got to do with programming. Everything! My mind is at peace, I have no distractions, I am existing in a mental and physical mileux conducive to letting my **mind** float free of constraints and conceptualise. I can gaze into nothingness and **reflect** .

This months article is probably one of the most difficult I've ever attempted to write, not so much because it contains heaps of source code which I had to strain my brain to write, but because once I started to think about the topic of the article I realised just how difficult it is to write about concepts. The Oxford dictionary defines concept as " a general notion","idea". How I conceive and execute an idea is bound to be different to the next person. It brings up all sorts of questions about just how we as human beings **think**, and the whole question of what **awareness** itself actually is. Are some people more **aware** than others of both their **self** and their environment. Has the human race always been aware, or has awareness evolved at some point in time because it bestowed an evolutionary advantage? In many ways our seperate realities are like overlapping mathematical subset diagrams - they allow us to appear to interact, yet much of each reality is different to the next. Pondering these and other earth shattering questions made me recall a rather nice quote from Herman Hesse, which I'll persuade our editor to print at the end of this article.

IDEAS.

Ideas for programs are something I've never lacked. To the contrary I find I've so many ideas I tend to write heaps of programs to a developed enough state for my own use, ie I'm happy to live with a few bugs, but not developed enough to release to the user who wants things perfect. Obviously a problem I've got to work on.

However, it never ceases to amaze me the apparent lack of drive and imagination present in members of society in general. How many people do you know, who are constantly complaining "there's nothing to do"! They have no interests, and once they lose their job, or retire they vegetate till they become ill and die. I'm sure this scenario is not related to intelligence either. More likely, it due to the fact that some children are lucky enough to grow up in an environment where their keepers love life and are constantly fascinated by the world around them. How long is it since you looked up at the landscape and felt a thrill of amazement surge through you just because its there? - what ever happened to your "joie de vivre"?

Many successful adults are so, in part, because they manage to keep the essence of a child, imagination and **awareness of self** and environment. What child doesn't spend time living in many a fantasy world, constantly exploring their many new ideas?

I've heard many people say they don't have any ideas for programming. I don't believe they are not capable of them. It is essential to re-capture some of that essence of being a child. The biggest killer of lack of success in life is to assume one is incapable of an achievement . The same applies to programming. Assume you will be able to work out the routines for any application and you surely will.

As for ideas - look around you. Many really useful programs **exist** for the Ti. Bathe yourself in the glow of your vdu one afternoon, ignoring all the theories about links between vdu radiation, sterility, cataracts, and leukemia. Forget the gloom and doom predictions about the impending crash of the economy or your 17%+ house mortgage . Flick through all your favorite programs, try and get the **feel** about what it is you like about them. What makes Funnelweb so good, what about the Horizon menu, how about Ti-Artist. What is bad about them -what don't you like, as a user; what features would you like enhanced. Do you like the screen displays, is the key usage easy or are you jumping about all over the place developing that non-existant repetitive strain injury syndrome; are your eyes sore, does the flicker of the colors cause you discomfort, or do you get straight out photic epilepsy! All in all try and become **aware** of the **feeling** you get when you use a good program and the feeling you get when you use a bad program.

Only once you've let your **senses** become aware should you then try and logically analyse **why** you feel those things about the program. Make a list of the good and not so good features of a program.

If you decide you don't want to write a program for everyone else, or feel too embarrassed by your efforts (which you shouldn't), do it for your **self**. I cannot stress too much the importance of writing something. No start is too small, even if, using the design editor and menu codes we will later develop, you write only a series of screen displays enabling you to construct a program which does nothing, expect lets you run up and down its menu's, but looks genuine. Seeing your design and name as the author will spur you on to greater things.

Try your hand at home record keeping, an inventory for serial numbers, a recipe program for your family, something for your tax, programs to help make programming easier etc. I've written ones to make my own phone book, act as a electronic directory, save the forth dictionary to Ti-writer file format for sorting, commenting etc.

Not all your ideas will end up in an ongoing useful program - but it doesn't matter. The **experience** gained in writing them is invaluable and will be **subconsciously** called on for your next effort.

Don't be averse to filching ideas written by other writers, or on other computers. There's nothing illegal about reverse software writing as long as you use the concept for another purpose, not just to mirror-image what you filched it from. Myriads of IBM programs abound. Don't be put off by the lack of 80 column display - your **imagination** can convert to 40 columns.

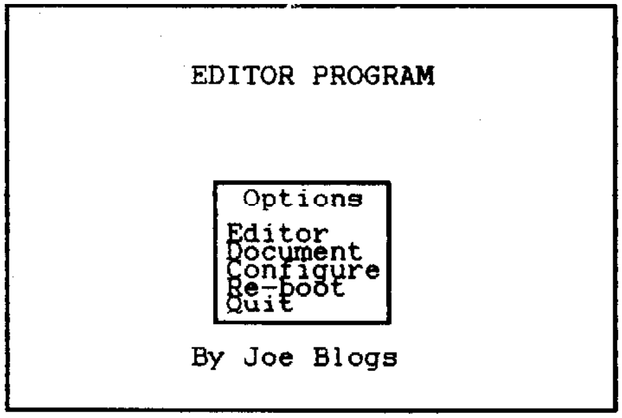
ENVIRONMENTAL INFLUENCES

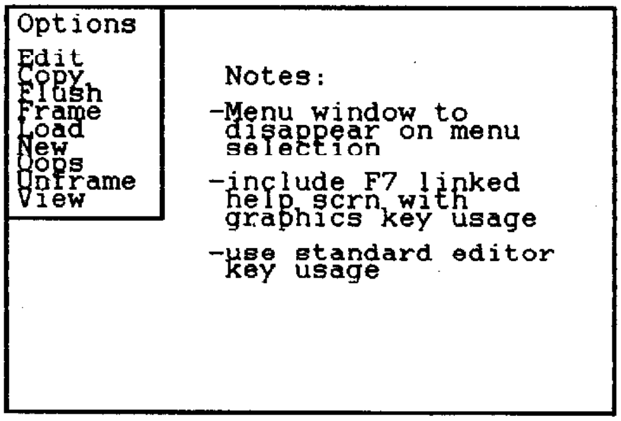
Probably the worst place to begin conceptualising your program is in front of the computer. The screen is far more limited than the uncharted expanses of the **mind**. I like a nice sunny afternoon, a comfy chair on the balcony, a chilled glass of dry white vino, the sunlight streaming through the trees keeping me warm, a wad of computer paper ,a pen or pencil, and mozart playing in the background. Trying to conceptualise whilst surrounded by irritating outside influences is doomed to failure.

Another good time is when the ideas hit you. I've woken up from sleep with a brilliant idea and written it down on the back of the box of tissues next to the bed to follow up next day, or got up, sat at the desk and spent an hour on the idea.

EXPANDING THE GERM OF THE IDEA.

Once you have decided on what the program is about, the next aspect is to decide what you want the program to do. If you are writing a personalised data-base, after a while you will find the formats for similar programs are the same, though what they do may be different. Other times with new ideas you will have to feel your way. I like to actually visualise what the Tv-screen will look like. I like to see my name in lights, so I usually sketch what I call the titlescreen of the program, and include a central menu of what I want the program to do. As an example I'll use the design editor program we may write. My initial sketch pad may have looked like:





Notes:

-Menu window to disappear on menu selection

-include F7 linked help scrn with graphics key usage

-use standard editor key usage

What I end up with then, is sort of like a screen display flow diagram, which is what the program will end up looking like as its running.

LIST WHAT THE PROGRAM HAS TO DO

We will go into this in much more detail in a future article. However, I sit and jot notes as to what capabilities I want my program to have. Make reference to your lists of the good and bad features you have previously analysed. Is it going to be a file oriented program, do I want it to be able to be accessed from the extended basic environment. Expanding on my concept of the screen displays above I may have just designed my new general ledger. I may decide:

1.It must use file based records for greater security as if I use screen based records they are easily erased.
2.Must be capable of data accept, change,printout etc. I may want a segment for statistical analysis.
3.Must be user friendly with on-screen help as my secretary will be using it.
4.Will have to have routines for date handling, daily date update, validation.....etc.
5.Will eventually occupy x no. of sectors on a disk.....is there going to be enough room?
6.Must have security entry clearance....etc.

And so on.

Each of these will again be broken down in turn to further specify what subroutines will be needed, but this will be discussed in a later article on program specific code.

Only when I'm fairly clear on the central idea of the program and its flow, do I turn to the computer.

Next month I'll discuss my next step - the prototype screen and window displays for my program, and will overview the concept of the editor program we will write in ensuing months.

We'll leave the last word to Herman Hesse:

Think of your being as a deep lake with a small surface. The surface is consciousness. Here all is brightness, and it is here that what we call thinking takes place. But this surface is infinitely small. It may be the most beautiful, most interesting part of the lake, for in contact with the air and light the water is transformed and enriched. But the particles of water at the surface change unceasingly. They rise, they fall; there are always currents, displacements, shifts; every particle of water wants to be on the surface at some time.... Now just as the lake consists of water, so our SELF or soul (the word does not matter) consists of thousands and millions of particles; of an ever-increasing, ever-changing store of possessions, memories, impressions. Of all this our consciousness sees only the small surface. The soul itself does not see the infinitely larger part of its content. A soul, it seems to me is rich and healthy and capable of happiness when there is constant exchange, a mutual renewal, between the great darkness and the small field of light. Most people have within them thousands and thousands of things that never rise to the bright surface, that rot in the depths and torment them. Because these things rot and bring torment, they are rejected time and again by consciousness, which distrusts and fears them. This is the substance of the morality that says: What has been recognised as harmful must not be allowed to rise to the surface! But nothing is harmful and nothing is useful, everything is good or everything is indifferent. Each one of us has within him things that belong to him, things that are truly his own, but are not permitted to emerge. If they did, says morality, a calamity would ensue! But quite possibly it would be a stroke of good fortune! Therefore let everything rise to the surface. The man who submits to a morality is impoverishing himself.

ADDRESS FOR CORRESPONDENCE

RICHARD TERRY
141 DUDLEY RD
WHITEBRIDGE
NSW 2290

(049) 436861

GOANNA TAILS

Bob Carmany

Ron has his famous "Black Hole" and Tony has all of those "cavorting beasties" that he named his software program after. What does that leave for "poor" Larry Reid way out there in the "Land of Banana Benders"? An interesting question, mate!

Just to the west of Brisbane we find Darling Downs long famous for livestock of all sorts. Swine, cattle, sheep -----and goannas. Lots and lots of goannas --big goannas, and small goannas. A real natural resource! That brings us to the beginning of this "Goanna tail (tale)".

Ever hear of Cecil Plains --- not the footballer, mate, but the crossroads about 200 KM west of Brisbane. When October is in "full bloom" so are the Goannas. Cecil Plains is host to the Sand Goanna Championships, an annual event soon to be the premier sporting event in all of Oz. The idea is simple, there is this 50 metre course with a tree at the end. Whichever goanna runs the course and climbs the tree in the shortest elapsed time is the winner. That is the easy part! The hard part is that each entrant has to catch his own goanna!

Sand goannas are interesting beasties --- a metre or so in length, muscular, quick, and with razor-sharp claws. Quite nasty customers, actually. There are two ways to catch one for the race. Both start with a large tub of ice with a copious supply of 'XXXX' (Foster's, Toohey's Draught, etc.) placed therein. You can grab a stout chaff bag and walk through the brush beating a couple of tree limbs together. The commotion will likely put a sleeping goanna to flight at which point you make a mad dash after him and try to get a grip behind the head with one hand and at the base of the tail with the other simultaneously. It is hot and dusty work and each foray after a goanna should be followed by a rest and a tin of 'XXXX'. The main hazard with this method is that one rarely succeeds the first time and each successive attempt gets a little tougher (it must be the heat --couldn't be the 'XXXX').

Why waste all of that energy running about after a lizard? The time-honored technique of the locals minimizes the exertion but still allows for the tin or two (?) of 'XXXX'. The first thing that you do is find a large clearing. Then you need a 'mate' (preferrably an IBM or Amiga User). You have your 'mate' stand in the centre of the clearing holding the chaff bag and standing very still so as not to alarm any passing goannas. From this point, the procedures are similar. You walk about with two tree limbs making as much noise as possible trying to scare a goanna into the clearing. The poor beast will undoubtedly thinks that your motionless 'mate' is a tree and will scramble up his leg and perch on the top of his head with those claws in a death-grip. All you have to do is stop by and pick him up and drop him in the chaff bag and you are ready for the race.

By now, you are thinking to yourself "what is the point of this yarn"? Absolutely nothing!! Is any of this true? Yes --- the Sand Goanna Championships are held in Cecil Plains in October and they do drink 'XXXX' and other types of "refreshments" there.

EDITOR'S NOTE!!!
WELL!! after that story I dare say we can expect a report in the near future from Bob claiming that he has a captive BUNYIP in his backyard. This proves that not only R.K. can handle the truth with considerable license.
To quote Henry Fielding;

"This story will never go down"

Joe.

Bits and Pieces with Joe Wright

Have you ever been into a rain forrest?. Have you ever touched a tree that is several hundred years old?. If you havn't then you will not yet have experienced the wonderful feeling of belonging which one gets when in such a place. Rain forrests, hold so many secrets that man has yet to learn, yet here we are tearing them down at an ever increasing rate. As we tear them down are we in fact tearing our own very existence down?.

One impulse from a vernal
 wood
 May teach you more of
 man,
Of moral evil and of good,
Than all the sages can.

ordsworth

SEARCH FOR A FORTH PROGRAMME
****
After reading Richard's article last month I decided to have a search for a programme written in Forth. The first place I looked was in a couple of boxes of discs I had bought of a person who was selling his system. I had remembered seeing something amoungst them. After a short search I found it; **J. P. GRAPHICS.** Not only is it written in forth but it also allows the users to write LOGO type routines in Forth to creates graphics designs.

J.P.GRAPHICS.
****
The programme and docs have no address for the writer J.P.Morin. Also I am not aware as to whether the programme is copyrighted or otherwise.

The programme allows the user to create graphics designs in the bitmap mode. The programme can be used in three mode;
1) BIT-MAP DRAWING
2) DESSIN
3) DESSIN2

BIT-MAP MODE.

In bit-map mode the user has 71 single key commands which can be used to create a design on screen. Some typical examples are;

Ø - Pick up pen
1 - Assign cursor a 1 pixel paint brush.
6 - Assign cursor a 8 pixel vertical brush.
b - Anchors beginning sprite for line, circle etc.
c - Draw a circle
d - Draw a right arc.
p - Paint command.
l - Draw a line between the beginning sprite and cursor.
T - Selects typing mode. displays large capitals.
Fctn 4 - Exit drawing mode to editor
. - Prints screen.
CTRL E - Clears the screen.

DESSIN

Selecting DESSIN mode places the user in the forth editor, Logo commands and the forth vocabulary can be used to create words to be executed in bit-map mode to create graphics designs.

DESSIN2.

Selecting DESSIN2 mode puts a 3 line editor section at the top of the bit-map screen. This allows words to be written as in DESSIN mode. The additional feature of DESSIN2 is that you can execute the words in command mode and see your graphics design on screen so you can tailor yourwords.

USING THE PROGRAMME.

To get the fullest use of this programme one needs a knowledge of forth. However after having said that I must point out that any user will find the programme very useful even with no knowledge of forth. The documentation is adequate enough for the programme to be put to good use. The outstanding feature of this programme is the ability to have large text included on the same screen as graphics characters. Currently only one set of large characters is available. I intend attempting to design some more fonts for use with the programme.

The programme's minimum requirements are; EXB. 32K,One disc drive, and of course a printer. Once booted you are asked if you would like a demo, (Y/N), the demo is very impressive but is a bit long winded. After you have seen it once you will then only see it again when you show some other person the programme. Be warned also that the programme had background music ALL the time it runs. That music is alright for about 45 seconds, after which I turn the volume down on the TV. The docs, thankfully, include the code to be entered in DESSIN mode to stop the music playing.

Once in the bit-map drawing mode it is then a simple matter of looking at the docs and trying to draw something using those single key commands described.

THE SCREEN.

The screen has a column of 6 two-digit numbers in the top right corner. These numbers are referred to as P1/P6. These parameters can be set by using shift 1 through to shift 6, the pointer is placed beside the parameter you select. A number can then be entered, the enter number mode is terminated by pressing any non numeric key. Immediately below these parameters is the 15 colour table, the colour can be changed by pressing the = key. In the bottom right corner is a three digit number, this is the angle value, this is set by pressing n then entering the angle required. A non numeric key terminates number entry. (Ø=right, 9Ø=down,18Ø=left). A sprite (the cursor) in the centre of the screen is the brush, it's shape changes to indicate the type of brush selected to draw. Joystick or arrow keys will move this sprite. A second sprite, called the beginning sprite can be anchored anywhere on the screen to draw lines, circles and other between it and the cursor.

FOR EXAMPLE TO DRAW A CIRCLE;

Using the = key move the pointer down the colour table on the right hand side of the screen to select the colour you want to use for drawing. Then using the arrow keys (or joystick), move the cursor to the left edge of where you would like the circle draw. Now press b, this anchors the beginning sprite. Again move the cursor to the opposite side of your proposed circle, then press c. The circle is now draw.

To draw a square (CTRL E clears the screen and a attaches the beginning and cursor sprites together) set the begging sprite as above and then move the cursor to the other end of the diagonal point in your square, then press s.

Once you get the hang of it, it is quite easy to use, joysticks can also be used to move the cursor as can FCTN W,E,R,S,D,Z,X,C. Each giving the normally expected movement direction, ie horizontal, vertical and 45 degrees.
Selecting the brush type then using the arrow keys you can create your own master piece, remember the Ø lifts the brush. To rub out simply select the same colour for your brush as the screen and run the brush over the section you want to remove. To prevent over running pressing D will place the cursor movement into single pixel step. Press D to return to normal drawing speed.

DESSIN2.
To get into DESSIN2 mode press FCTN 4 in the bit-map mode. Forth words can now be created using the standard forth vocabulary plus the additional LOGO type words included with the programme. Here are a couple of examples to type in, the docs include an example for a rectangle, this example is for a hexagon.

Press FCTN 4 to enter DESSIN2 mode and type the following;

```
: HXGON PD 1 GET 5 Ø DO DUP FD
6Ø LT LOOP DROP ; <press enter>
```

Type it in as if on one line, the forth editor will wrap the words around for you. This creates the forth word HXGON which draws the hexagon. That is you have entered a word (HXGON) on the forth dictionary. PD is the LOGO command to put the pen down. 1 GET places the value in P1 on the stack. 6 Ø DO sets up a FOR NEXT LOOP to run 5 times. FD takes the value taken from P1 off the stack and moves the cursor that number of pixels. 6Ø LT turns the cursor through 6Ø Degrees. LOOP terminates the FOR/NEXT LOOP and then DROP removes the unused P1 value from the stack.
Now type;

```
ASC R DFN HXGON <press enter>.
```

This section of code defines the R key to execute the word HXGON.

Now type DESSIN and press enter to return to bit-map mode. Press SHIFT 1 and then enter a value into P1, example 2Ø. Press any non numeric to terminate, position the cursor towards the left of the screen (about halfway). Select a colour using the = key. Now press R and the hexagon will be draw. Now change P1 again and press R, repeat this process with increasing values in P1 to see a 3-D effect. After completing this the screen can then be dumped to your printer by pressing the period key.

DESSIN MODE
This mode could also have been used to enter the above word. The advantage of DESSIN over DESSIN2 is that you have a full screen forth editor available.

CONCLUSIONS.
I like this programme, it is easy to use, and, gives reasonable graphics. It would be nice to have more fonts available . Perhaps somebody somewhere does have some??. In the mean time I hope to be able to get some going over the next few months. With regard to the writer, Jeanne-Pierre Moran, I am jealous of his knowledge of Forth. He has produced an easy to use well presented programme which I am sure that many people would be using if they had taken the time to look closely at it. I for one will be using it for diagrams in the future. Included on the disc which I have is the source code for his programme, Mr.Moran should be thanks for this also well. It is by including source code on discs of programmes that inexperienced programmers such as yours truly can learn new techniques and update my less efficient code.
Richard this month talks of the concepts of programming. What Mr. Moran has done has not only captured a concept. But! he has also proven himself capable of writing that concept into a working programme. Tony McGovern and the many other people who have produced quality software for the T.I. also fit into this same category. Tony to his eternal credit gives freely of his knowledge, we at the H.V.99'ers are lucky to have Tony readily available to us to give this knowledge. In Mr. Moran's case we don't have him here to talk to but giving the source code is definitely the next best.

Joe Wright

random bytes

with BOB CARMANY

I've been correspondi-
McGovern for >
--ever since I
FUNLWRITER Vn
time span, I've
the judiciously
by antiquated sy
the long run Tc
his remarks about
system. Now, hc
last laugh! I fina
full PE-Box and a
DSSD drives with an
after adding in wl
surplus standalone
sold. Ah, the lux
suppose that Tony
agitating for me to g
card and Quest 200. N

First of all a bit of
to the first install
PROGRAMS series. Of
quoted string should be
QUOTATION MARKS ra
parentheses. A ridiculou
it crept into the text in
my myopic proofreading.
didn't cause too much conf

Everyone seems to be concer
what's new for the TI.
some quality new software
but there is some equa
quality software around a
Most of it is stuff th
marketed commercially that e
has forgotten about in the m
to add new programs to
library. With this in mind,
take a look at some "oldies
goodies".

One of the most interesting pack
is the AMERISOFT extension of
Like most the the XB extensions,
uses a series of A/L routines t
can be LINKed from the
environment. Unlike most of the
it includes a large number o
graphics commands. In fact, almos
the entire lot of the European APESOFT graphics are present. You can construct windows, draw circles or ellipses, and other shapes from XB just by entering the necessary parameters. There are screen dumps and other utilities to go along with the package as well. It is well worth a second look!

Another software package worth a second look is the COMPILER from Ryte Data. It has some limitations but it can be of significant value for some of your XB programs. It will increase execution speed and it is very easy to use.

Before I end this column for this month, let's take a look at some values that can be used in CALL LOADS and A/L programs.

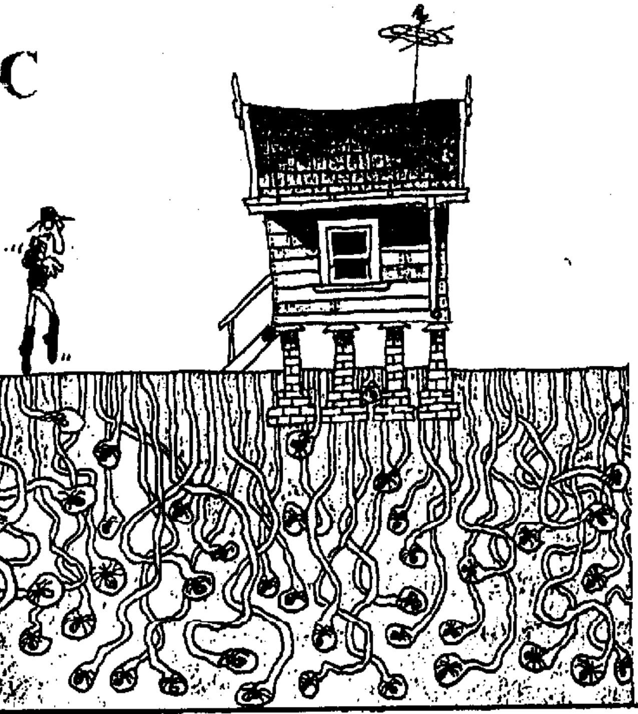
-31970	>831E	Step in NUM mode
-31888	>8370	Pointer to the end of VDP RAM --start of disk buffers
-31878	>837A	# of sprites in motion.
-31884	>8374	Keyboard # to scan (0 - 5)
-31882	>8376	Joystick Y from scan
-31881	>8377	Joystick X from scan
-31880	>8378	Random number generator.
-31808	>83C0	Random number seed

You can POKE and PEEK with these values and see what you can come up with.

One more thing worth mentioning before I get a "buffer full" message. Remember that all of those disk drive cleaning disks are abrasive. As such, they will wear down your disk heads. With this in mind, only use them as a LAST RESORT. In one of his newsletters Craig Miller said that he used his TI 7 hours a day 6 days a week and never found the need to use a disk driver cleaner more than once a year. Unless you use your drives more than that, you may never have to use one.

Once again it is getting close to the end of a column. There always seem to be "honey-do's" waiting --- you know, "honey do this"and "honey do that". Those of you who are married can relate to it well. Any suggestions for future topics to be covered in this column would be greatly appreciated. Send them on to Brian and he will forward them to me. 'Til next month . . .

PART III
EXTENDED BASIC TUTORIALS REVISITED
by TONY McGOVERN



Our next example will be a good start on a non-trivial utility program for printing out TI BASIC or XB listings on a 80 column printer in two side by side columns which preserve the normal screen listing format. If you just LIST "RS232.BA=...." then the computer sends it out in DIS/VAR 80 format and it is up to you to tell the printer how to handle it. Something approaching screen image format is only obtained (with extra paper consumption) with the printer margins set way in. 80-col printout beats none at all by miles but let's try to be fancier. If you don't have disk or printer then this lesson won't be of immediate use, but will still be a good example to work through as a programming exercise. We might as well do something useful.

First we figure out what needs to be done, and work out a set of procedures that can be CALLed as needed. The program will do only the minimum necessary to do the job properly. Bells and whistles can be added later. In one or two places we shall make provision for adding extras (bells and whistles have nothing on speech) by dummy subprograms which can be filled in later. For a good discussion of the use of such "stubs" see the excellent book by R. Mateosian, "Inside Basic Games". The detailed coding examples in this book are in Apple or Trash-80 Basics (does anyone remember any more what a TRS-80 was ?), but Mateosian develops ideas in a form much more in tune with a TI XB subprogram realisation than with these less capable Basics.

So let's start designing our program by deciding what we want it to do. We want the output nicely formatted on the page with top and bottom margins, in 2 columns each in screen image (28 char/line) format. More columns (assuming the output device will handle them) are no problem -- once you can count to 2 then 3 is easy. Lines of Basic are not to be split from from one column to the next or from one page to the next. Some things commonly encountered in printed listings, such as indenting of FOR-NEXT loops don't fit at all well with the multi-statement lines of XB (but might with TI Basic listings) so will not even be thought about here. On the other hand insertion of spaces before REM or SUB statements greatly improves the readability of XB listings, without doing violence to the idea of being screen list compatible. Page numbering is no big deal to add (a console only XB program can fill 6 pages).

At the other end of the business the LISTing to be printed is assumed taken from a disk file such as DSK1.LIST where it has been written by LIST "DSK1.LIST". A trivial difficulty easily taken care of is the blank first record written by LIST. The real problem is that LIST doesn't care about preserving XB lines as distinct entities. Each XB line starts out as a separate print record and if it is less than 80 characters long stays in one piece. XB lines can easily extend into 2 print records and more (Basic lines much less frequently), but LIST places no markers to show which print records contain the start of XB lines. So if we are going to meet our specification that XB lines be treated exactly as in a screen list then something more subtle than a simple LINPUT is needed. There's one of our most important building blocks identified --- SUB BASICLINE(...).

Any utility program needs title and advice screens so there's SUB TITLES to keep all the details from cluttering the main program. The program will also need SUB OPTIONS(...) to handle file and device name entry and print options which might be offered.

Now the real core of the program is the way in which it must assemble a whole page before printing anything because line feed moves ever on (at least it does on our TI-99 printer). So we need SUB PAGEBUFFER(....) to take the output of BASICLINES, chop it into screen format hunks and decide where these are to be located on the page. Then we need SUB PRINTPAGE(...) to massage the completed pages and ship them off to the printer. That about sums up the sub-programs that are called directly from the main program, and all that is necessary is to figure out the initialisation -- DIMs, default filenames etc etc, and to write the logic for program flow.

Before we start writing any code we should decide what utility sub-programs are to be used by those already defined. As the list is written into columns SUB WRITECOL(...) is a good candidate for repeated use, and SUB WRITEPAR(...) to take a line of BASIC and return it chopped up into 28 character lines to WRITECOL. Since BASICLINE fetches the input records it is the appropriate place to detect End Of File. We might as well use PRINTPAGE to wipe the slate clean before writing a new page.

Let's dress up the input of filenames and Yes/No responses a little as SUB FILENAME(...) and SUB YN(...) , with SUB MORE(...) to end it all. Other useful utility sub-programs which will be included are SUB TXTCOL(..) to change display colors in one CALL, SUB KEYCON to carry the burden of "press any key to continue", and SUB DELAY(..) is always handy.

That about finishes the roster of procedures necessary to make up the listing program, and now the detailed coding can start after some thought on the necessary chains of parameter passing. The principle that you should plan your programs from the top down and code them from the bottom up is just as valid in Extended Basic as it is in TI-LOGO or TI-FORTH where the form of the language makes it difficult to do otherwise. Sub-programs make it possible to go the same way in XB with ease. Less capable dialects of Basic make it a lot harder to keep your thoughts organised and your code on the rails.

The actual program will now be listed piece by piece and commented on in detail. The listing has been transferred into this TI-Writer file from a working copy of the program using a more elaborate version. The present program is actually a simplified version of the one originally written, but is powerful enough to do a useful job.

```
100 REM ** SIMPLIST **
110 REM * PRINTER LIST *
120 REM ** FROM DISK **
130 REM -FUNNELWEB FARM-
140 OPTION BASE 1 :: DIM PRL
N$(66,2)

150 REM * DEFAULT VALUES *
160 CALL TITLES :: SFIL$="DS
K1.LIST" :: PDEV$="RS232.BA=
4800"
170 CALL KEYCON
```

The first part of the main program shown here sets default values and DIMensions the string array PRLN$ for two columns of 66 lines each. The top and bottom few lines will be left blank so that page format is obtained without sending printer control codes. A 66 line/page, 80 col. printer is assumed.

```
180 REM * New File Entry *
190 CALL OPTIONS(SFIL$,PDEV$
):: ENDFILE=0 :: LINPUT #1:N
EW$
```

```
200 REM * New Page Entry *
210 CALL PAGEBUFFER(PRLN$(,)
,ENDFILE)
220 CALL PRINTPAGE(PRLN$(,),
PDEV$):: IF ENDFILE=Ø THEN 2
1Ø

230 REM * End OR Next *
240 CLOSE #1 :: CLOSE #2 ::
CALL MORE(NM):: IF NM THEN C
ALL SPEAK("GOODBYE"):: GOTO
250 ELSE 190
250 STOP
```

OPTIONS returns file and device names as entered there, and the remainder of line 190 resets the End of File flag, and throws away the first line of the list-file. At new page entry the page buffer is filled and then printed out repeatedly until it runs out of listing, and then it asks if you are finished. That's all there is to the main program folks. And now to the sub-programs that do all the work.

```
260 SUB TITLES
270 CALL CLEAR :: CALL SCREE
N(11):: DISPLAY AT(12,6)BEEP
:"PRINTER LISTING"
280 SUBEND

290 SUB OPTIONS(S$,P$):: DIS
PLAY ERASE ALL :: CALL TXTCO
L(16,5)
300 CALL FILENAME(1,2,"Edit
as needed and ENTER","N?")
310 CALL FILENAME(4,4,"Sourc
e file for listing",S$)
320 CALL FILENAME(8,4,"Print
er devicename",P$)
330 CALL YN(" Change mind ?"
,"N",22,5,I):: IF NOT(I)THEN
CALL HCHAR(22,1,32,64):: GOT
O 300
340 DISPLAY ERASE ALL :: IF
S$="" OR P$="" THEN DISPLAY
AT(1,2)BEEP:"NO INPUT/OUTPUT
POSSIBLE" :: CALL DELAY(500)
:: GOTO 300
350 OPEN #1:S$,DISPLAY ,INPU
T ,VARIABLE 80 :: OPEN #2:P$
,DISPLAY ,OUTPUT,VARIABLE 80
360 SUBEND
```

TITLES here is little more than the barest stub, but you can fill that out to your own fancy. OPTIONS takes down the file names, does some checking, and opens the files.

```
370 SUB PAGEBUFFER(PRLN$(,),
EFL)
380 REM * New Col Entry *
390 PLN=6 :: COL=COL+1 :: IF
COL>2 THEN COL=Ø :: SUBEXIT
E LSE PRINT "":"** Reading c
olumn #";COL:"":""

400 REM * New Para Input *
410 IF EFL THEN PRINT "":" *
":"*** END of FILE ***":" *"
:"" :: SUBEXIT ELSE CALL BAS
ICLINE(NEW$,EFL):: PRINT NEW
$:""
420 CALL WRITECOL(PLN,COL,PR
LN$(,),NEW$)
430 IF NEW$="END of COL" THE
N 390 ELSE 410
440 SUBEND
```

The new column entry in PAGEBUFFER resets the line counter PLN to top of page with a margin, increments the column count, and exits back to the main program if the page is full. If not it tells BASICLINE to fetch a new program line and WRITECOL to enter it in the page buffer. If BASICLINE says it has read the last line it exits and lets the main program worry about that, otherwise it gets another Basic line or starts a new column. A stub here, CALL SKIPLINE(NEW$,SK), could have uses.

```
450 SUB BASICLINE(N$,E)
460 N$="" :: IF NX$="" THEN
LINPUT #1:NX$
470 N$=N$NX$ :: IF LEN(NX$)
<80 OR EOF(1)THEN NX$="" ::
E=EOF(1):: SUBEXIT ELSE LINP
UT #1:NX$
480 PX=POS(NX$," ",1):: IF P
X<2 OR PX>6 THEN 470
490 P=POS(N$," ",1):: IF PX<
P THEN 470
500 NR=-1 :: FOR I=1 TO PX-1
:: C=ASC(SEG$(NX$,I,1)):: N
R=NR AND C>47 AND C<58 :: NE
XT I :: IF NOT(NR)THEN 470
510 IF SEG$(N$,LEN(N$),1)="
" THEN 470
520 IF VAL(SEG$(NX$,1,PX-1))
<VAL (SEG$(N$,1,P-1))THEN 47
Ø

530 REM ** Check Quotes
540 N Q,I=Ø
550 I-POS(N$,CHR$(34),I+1)::
IF I THEN NQ=NQ+1 :: GOTO 5
50 ELSE IF NQ<>2*INT(NQ/2)TH
EN 470
560 SUBEND
```

The procedure BASICLINE which retrieves complete lines of Basic code from the LIST-file is the only part of the program with decision flow complex enough to warrant planning out separately on paper beforehand. I am not going to reproduce this here, but you can work out your own and see if it leads to similar code. The problem comes when the procedure has read in a line exactly 8Ø characters long. Does the next LIST record then represent a continuation of the same line of Basic or is it the start of a new Basic line? This difficulty can't be ignored if screen list format is to be preserved since 28 into 8Ø does not go exactly. The procedure provides a cascade of tests each of which checks whether the record being scrutinised should be appended as a continuation of the previous Basic line. A few more rare cases could be tested for along the same lines. There is at least one (that I know of) unlikely case which BASICLINE cannot resolve even in principle. Can you spot it ? It does seem to work well already though. The intricate input code is needed since a VARIABLE file can only be read sequentially, and if the battery of tests says that the last record LINPUTted does start a new Basic line, then this must be saved till BASICLINE is called the next time.

Just be thankful for static variables in XB subprograms ! You also have to take care not to set off the End of File alarm prematurely.

```
57Ø SUB WRITECOL(P,C,P$(,),N
$):: IF NC THEN P=6 :: NC=Ø
58Ø IF P>=57 THEN N$="END of
COL" :: NC=-1 :: SUBEXIT
59Ø CALL WRITEPAR(P,C,P$(,),
N$)
6ØØ SUBEND
```

Now that WRITECOL has the line of Basic it sends it off to be formed into a paragraph. This simplified program handles coming to the end of a column in a slightly wasteful way that is very simple to program. A normal XB program line lists at most on 5 screen lines, and no matter how tricky you are in entering longer lines the program has already limited it to a string variable (max length 255 or 1Ø screen lines) or has crashed with an error. The simple minded solution is to exit with End of Col message if the proposed starting line for the new paragraph is past a fixed place somewhat short of the end of the column. The value entered, line #57, is a compromise between making the program totally bulletproof or wasting space. A better approach is to print as far as possible, testing each new paragraph to see if it fits, and if not, holding it over for the next column. If you wondered why the string was called NEW$, then spare a thought for OLD$ which which vanished without trace during program simplification for tutorial purposes.

```
61Ø SUB WRITEPAR(P,C,P$(,),N
$)
62Ø P=P+1 :: IF LEN(N$)>28 T
HEN P$(P,C)=SEG$(N$,1,28)::
N$=SEG$(N$,29,LEN(N$)-28)::
GOTO 62Ø ELSE P$(P,C)=N$ ::
N$=""
63Ø SUBEND
```

Sub-program WRITEPAR almost was called SALAMI as it slices up NEW$ and assigns the slices to successive printlines. Once entered line 62Ø loops on itself recursively until the remaining piece fits on a screen line. It assumes range checking has been done before entry. In retrospect, 5 years later, that still looks a fine line of code.

```
64Ø SUB PRINTPAGE(P$(,),D$):
:PRINT "":"** Page print sta
rted"
65Ø PRINT "":"** Assembling
printlines":" and printing t
o" :: PRINT "":" ";D$
66Ø FOR I=1 TO 66 :: PRINT #
2:TAB(9);P$(I,1);TAB(45);P$(
I,2):: P$(I,1),P$(I,2)="" ::
NEXT I
67Ø SUBEND
```

Not much needs be said about PRINTPAGE beyond noting that line 66Ø formats a single print record from the two column entries and erases the page buffer as it goes.

```
68Ø SUB YN(A$,B$,R,C,X)
69Ø DISPLAY AT(R,C)BEEP:A$"
(Y/N) "B$ :: ACCEPT AT(R,C
+LEN(A$)+7)VALIDATE("YN")SIZ
E(-1)BEEP:A$ :: X=A$=B$ :: R
=R+2 :: SUBEND

7ØØ SUB KEYCON :: DISPLAY AT
(24,6)BEEP:"ANY KEY TO PROCE
ED"
71Ø CALL KEY(3,I,ST):: IF ST
=Ø THEN 71Ø ELSE DISPLAY ERA
SE ALL
72Ø SUBEND
```

```
730 SUB FILENAME(R,C,M$,D$)
740 DISPLAY AT(R+1,C):RPT$("
-",LEN(M$)):: DISPLAY AT(R,C
):M$ :: IF D$<>"N?" THEN DIS
PLAY AT(R+2,C):D$ ELSE SUBEX
IT
750 ACCEPT AT(R+2,C)SIZE(-15
)BEEP:D$ :: SUBEND

760 SUB MORE(NM):: DISPLAY E
RASE ALL :: CALL TXTCOL(3,12
):: CALL YN("More listings",
"N",16,2,NM):: SUBEND

770 SUB DELAY(A):: FOR A=1 T
O A :: NEXT A :: SUBEND

780 SUB TXTCOL(A,B):: CALL S
CREEN(B):: FOR I=0 TO 12 ::
CALL COLOR(I,A,B):: NEXT I :
: SUBEND
```

The FILENAME routine writes an underlined heading, DISPLAYs the default response, and ACCEPTs the reply. If it is asked no question, "N?", it expects no answer. The other SUBs just do their job when called. YN acts like input routines familiar in other TI modules.

```
790 SUB SPEAK(A$):: CALL PEE
EEK(-28672,SP):: IF SP=96 TH
EN THEN CALL SAY(A$) ELSE CA
LL DELAY(5*LEN(A$))
800 SUBEND
```

This is a last little goodie tagged on so that you may add speech prompts to your program where desired. A bald CALL SAY has the annoying behaviour that it seems to take forever in giving up the attempt if no speech synthesizer is attached. Line 800 checks that speech is connected and line 820 substitutes a controlled delay if not. CALL SPEAK("....") can then be inserted anywhere it is wanted in the program.

So there we have it, a worked out example of a non-trivial and useful program that makes essential use of the sub-program facility of XB. It shows that the XB programmer can, with a style that finds natural expression in the language without undue contortions, follow the general principles of "structured programming" without getting hung up in the Swiss straight-jacket so beloved by some proponents. The program as presented is a cut-down version of the all-singing, all-dancing model, COLIST, which has now grown to >22K and uses 48 subprograms. In all the versions, subprograms have been an essential tool for program development. Now it's time to take retrospective look at what we have done and chase a few more subtleties

EDITOR'S LAMENT!
****

You will have noticed that I have a new cover for this month. I hope to be able to have a new cover each month depicting a particular theme. Next month will be either our unique animals, or Newcastle harbour in the 1800's. This month's cover emphasises Maitland and Sinlgeton. Your comments would be welcome.

CHRISTMAS

"Christmas is comming and the geese are getting fat" etc, etc. Time for you to start thinking and doing something about your contribution to the Christmas Bumper newsletter. I couldn't is not an acceptable excuse! This newsleter and the previous one have been good because of the good articles presented. The only problem here is that they are the same names producing them each month. The newsletter should reflect the group as a whole, not just the active few.

ARTICLES.

They can be presented in any form, preferably on disc but hand written, or printed copy is acceptable. I do need some articles of one column length, to place between the longer more complex article. This gives the reader a bit of a spell!. When you write (notice I said when not IF) some one basic rule will make the task easier for you. Firstly don't use long sentence that wander all over a topic doing that will cause the reader to loose track of what you are tdding to say to him which in turn causes him to loose interest in what you are writing about this sentence is one such example. Don't make them. So short that. You loose the meaning . Of what you want. To say.

Give it a try you will find it interesting.

QUOTATION

"Every quotation contributes something to the stability or enlargement of the language.

Samual Johnson
--Preference to dictionary.

AL'S

MARKET PLACE UPDATE

WITH *Alan Franks*

If someone told you, that you could buy brand new Mitsubishi double sided half height drives for sixty dollars each (yes its not a miss print), you would say they had to much sun. But the good news is you can! an advertisement in saturdays paper read, "Disc drives for sale from sixty dollars". So I gave the number a ring and now have a pair up and running in my PE box. Apparently they come "factory line" installed in IBM clones. When people ask for 1.3 meg drives these are left surplus.
So! if you are interested you can ring Col on (Ø49)54Ø317 to attain what, up until now, would have set you back at least four hundred dollars.
However, don't panic, Col has them wall to wall, so to speak.
Installing them is no real drama either. All you need is one thirty four pin ribbon connector which you can clamp on your existing disc drive ribbon cable. Also a four pin power plug the same as the one running to your present drive. Which you connect in parallel with four short pieces of wire from your present plug. The only other modification is to configure the drives by unscrewing and removing the sheet metal cover on the drives. You will then see eight pins running opposite each other with numbers beside them. So to configure drive one just slip the bridge connector over the two pins numbered Ø. Drive two is set by slipping the bridge connector over the two pins numbered one.

SECOND HAND MARKET.
In regards to the second hand hardware and software market the following bargains are available.

No (1)

Stewart Bradley still has a console plus PE box with 32k and disk controller cards for sale. You can contact stewart at 513246 with any reasonable offer but be quick as they are getting to be a scarce item.

No (2)

Dave Kaiser has a console, extended basic, and assorted programs for sale. Ring 565292 and ask for Dave.

No (3)

Paul Slowey has a console and the following modules for sale parsec, buckrogers, tunnels of doom, hangman, ti-invaders. You can ring paul on 591569.

No (4)

Eddy has a console with original TI joysticks plus a Computec joystick with the TI adapter along with the following modules fractional numbers, reading on, meteor multiplication, demolition division, hunt the wumpus, TE 2, touch typing, munchman, alligator mix, phys fitness, car wars, ti- invaders, attack, number magic, minus mission, multiplication, personal record keeper, dragon mix, alien addition, blasto, division 1, addition subtraction 2, tombstone city, early learning fun, hangman, video games 1 and a few cassette programs. All the modules are in a special carry case and the manuals have been put in a folder. You can make Eddy an offer at 485598.

No (5)

Bill has a console, joysticks, ex basic and about four assorted modules plus a book on basic programing. Just ring 438Ø24 and make Bill an offer.

No (6)

Robert Adams has a console with joysticks, ex basic, video games 1, hangman, video chess, touch typing, plus graphics pack on cassette. Ring Robert on 665631 and make an offer.

No (7)

ME! I have now got a Burrows box and one full height double sided drive for sale $1ØØ.ØØ ono. You can haggle with me on 45917Ø.

THE INFORMATION PAGE

GENERAL MEETING.
A good number of people took home library discs after the July meeting to catalogue. More discs will be available at the August meeting for you to take home and catalogue. We hope to have the software library FULLY catalogued in the near future. We will then be making available discs of the complete catalogue to our members. The August meeting will see the start of an Assembly language competition as proposed by Tony McGovern. If you read Tony's article else where in this newsletter you will get some more information.

FORTH.
The Forth SIG is on again first tuesday of september at Richard Terry's Surgery. Last one was a real success. Even if you are not into forth come along and sit in, you will be welcomed. This next meeting we will get into some programming and also hopefully set a few goals.

COMMITTEE MEETING.
Second tuesday 12/9/89. This month the meeting will be help at the Ambulance station Boolaroo. No! we have not gone there to be near to medical attention because our meetings get violent. We will be trying different locations each meetings. Personaly I prefer Pete Smith's place. (His wife put on a great after meeting feed for the troops!!). Once again drop in if you wish.

EXTENDED BASIC.
Gary Jones continues extended basic again at Bob McClure's house, third tuesday 19/9/89. If you go along to this meeting a good peace offering is a packet of Arnott's biscuits, any type so long as they are chocolate!. Gary must have set some kind of record for these classes, 5 years!.

GENERAL MEETING.
Once again at Jesmond Community Centre at 7:00 pm 26/9/89. Pete is puting together another trivia night for this meeting.

MEMBERSHIP RENEWALS 1989.
****

Annual membership fees are due for 1989/1990, if you have not yet sent your money in do so soon.

AUSTRALIAN MEMBERSHIP IS $25:00
OVERSEAS MEMBERSHIP IS $40:00 Australian.

Address membership renewals to Brian Woods, his address is inside the front cover.

CALENDAR 1989.
****

Our social secretary Bob McClure has booked 20 tickets for a Dinner/show at Kings theatre for 18/11/89. This is a 4 course dinner and a music hall show to follow all inclusive for $32:00 per head. Good night guaranteed. One of our members, Merv Walker is involved in the production. This show is instead of our usual christmas get-to-gether at one of the many fine local restraunts in the area.

October will be our month for another SPECIAL INTEREST DAY. Tim Watkins is really keen to get this organised. Location is yet to be decided.

GLOSSARY OF TERMS.
****

BASE ADRESS.
****
A given adress from which an absolute adress is derived by combination with a relative adress.

BAUD.
*****
A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example,one baud equals one half dot cycle per second in morse code,one bit per second in a train of binary signals,and one 3_bit value per second in a train of signals each of which can assume one of eight different states.

BCD.
****
Binary coded decimal notation.

BENCHMARK PROBLEM.
****
A problem used to evaluate the performance of hardware or software or both.

BINARY.
****
1.Pertaining to a characteristic or property involving a selection,choice,or condition in which there are two possibilities. 2.Pertaining to the number representation system with a radix of two.

BINARY CODED DECIMAL (BCD).
****
A binary numbering system for coding decimal numbers in groups of 4 bits. The binary value of these 4_bit groups ranges from ØØØØ 1ØØ1,and codes the decimal digits "Ø" through "9". To count to 9 takes 4 bits;to count to 99 takes two groups of 4 bits;to count to 999 takes three groups of 4 bits, etc.

BLOCK DIAGRAM.
****
A diagram of a system,instrument,or computer in which the principal parts are represented by suitable associated geometrical figures to show both the basic functions and the functional relationships among the parts.

BLOCK TRANSFER.
****
The process of transmitting one or more blocks of data where the data are organized in such blocks.

BOOTSTRAP.
****
A technique or device designed to bring itself into a desired state by means of its own action,eg.,a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device.

BORROW.
****
An arithmetically negative carry.

BRANCH.
****
1. A set of instructions that is executed between two successive decision instructions.
2. To select a branch as in definition 1.
3. A direct path joining two nodes of a network or graph.
4. Loosely,a conditional jump.

BRANCHING.
****
A method of selecting,on the basis of results,the next operation to execute while the program is in progress.

BREAKPOINT.
****
A place in a routine specified by an instruction,instruction digit,or other condition,where the routine may be interrupted by external intervention or by a monitor routine.

BUFFER.
****
An isolating circuit used to avoid reaction of a driven circuit on the corresponding driver circuit. Also,a storage device used to compensate for a difference in the rate of flow of information or the time of occurrence of events when transmitting information from one device to another.

BUS.
****
One or more conductors used for transmitting signals or power.

BYTE.
*****
A sequence of adjacent binary digits operated upon as a unit and usually shorter than a computer word. Usually 8 bits.

C#97

Mr R Carmany
15Ø4 Larson St
GREENSBORO NC 274Ø7
USA