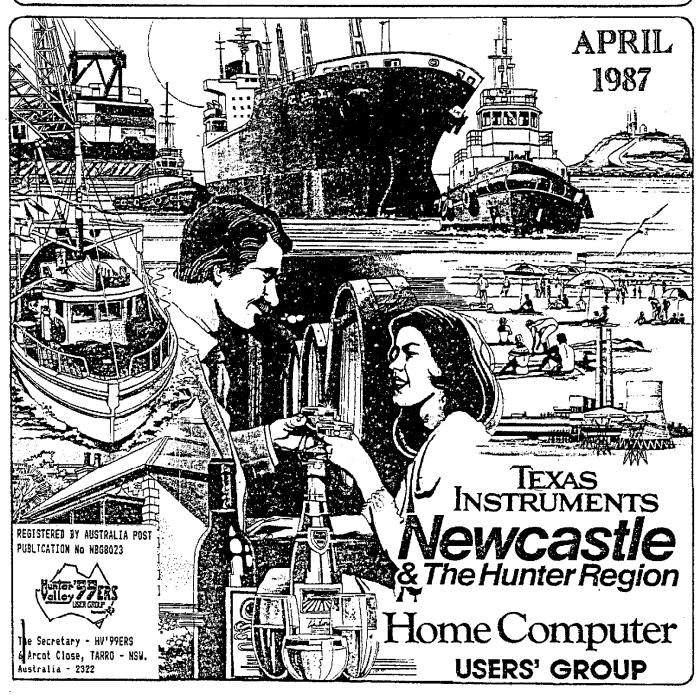
HUNTER VALLEY 99'ERS NEWS



TI 99/4A

HOME COMPUTER NEWSLETTER



YOUR COMMITTEE

ACTING PRESIDENT Jim Grimmond 31 Jarrett St., TORONTO 2283 Ph. 595751

SECRETARY
Albert Anderson
6 Arcot Close,
TARRO 2322
Ph. 662602
Viatel 496626020

TREASURER
Brian Rutherford
9 Bombala St.,
DUDLEY 229Ø
Ph. 498184

SOFTWARE LIBRARIAN Alan Lawrence 35 Bayview St., WARNERS BAY 2282 Ph. 484509

PUBLICATIONS LIBRARIAN
Paul Mulvaney
26 Marmong St.,
MARMONG POINT 2284
Ph. 583623

EDITOR Brian Woods 9 Thirlmere Pde., TARRO 2322 Ph. 462307

COMMITTEE
Bob MacClure
75 Deborah St.,
KOTARA SOUTH 2288
Ph. 437431

Gary Jones 53 Janet St., JESMOND 2299 Ph. 573744

Tony McGovern 215 Grinsell St., KOTARA 2285 Ph. 523162

Peter Coxon 25 Reserve Rd., WANGI WANGI 2267 Ph. 751930

CONTRIBUTIONS

Members and non members are invited contribute articles publication in HV99 NEWS. Any copy intended for publication may be typed, hand written, or submited on tape/disc media 35 files suitable for use with Ti Writer (ie. DIS/FIX 88 or DIS/VAR 88). A suitable Public Domain word processor program will be supplied if required by the club librarian Al Lawrence. Please include along with 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 MV99 WEWS (unless notified otherwise) considered to be PUBLIC DOMPIN. groups шishina Other user reproduce material from HV99 NEWS may feet free to do so as long 85 source and author 315 recognised. Articles for publication can be submitted to.

THE EDITOR
HV99 NEWS
9 THIRLEMERE PDE.,
TARRO 2322

General address for ALL other club

THE SECRETARY
HVSS USER GROUP
6 RECOT CLOSE,
TARRO 2322

INSTRUMENTS.

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY WINE USER GROUP. Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it technical, σf generai, or programming пο nature. responsibility can be accepted by HUSS NEWS as a result of applying such information. TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS. HV99 is a non profit group of computer nagray not T199/40 affiliated in any way with TEXAS SE

Ac HV99 our to I brea a Aus and the

exc man by pro for thi Aus

an as Da Gr th

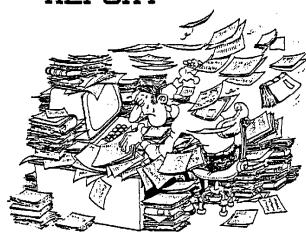
to sc we cu

y t

HPG

Garon

SECRETARYS
REPORT



FROM ALBERT ANDERSON

Again a busy month for us here HV99 and once again welcome to all Things are starting our supporters. to return to normal after the summer break and as the hemispheres undergo a season reversal it looks like its Australias turn to hit the keyboard and use the imagination to back up the recent excellent work carried on by our overseas associates. This excellent work which has come from many user groups has been spurred on by the rather hard to believe to come which continue products forth to enhance the utilization We here 4A machine. the Australia really miss this service and wish to thank such organisations as Myarc, Triton, Horizon, Ryte Data, Tenex, Tex-Comp, Millers Graphics, MICROpendium etc.etc.etc., list goes on, who have been absolutely marvellous when called on to supply goods and information from some 10,000k's at usually a 3 or 4 week turn-around. The pain οf currency conversion and government charges doesn't hurt so much when you are able to get hold of an item that you desperately want/need.

Memberships this month came from both near and far. Welcome to the HV99ers to Stephen Clucas of Rankin Park (local), Robert Carmony from in North Carolina, USA. Greensboro San Francisco, and to Ray Dukes of who is at present Ι California believe on military duty somewhere Ţ know that our Korea. in newsletter to some peculiar goes

places around the globe, but Korea? Well I hope it gets there for you Ray.

As usual extend to you we invitation write for YOUR to and have others benefit newsletter from your experiences, whatever they may be. SHARE it, see your name in print. You'll get a buzz out of it.

Whilst I am at it thanks to our regular contributors and our latest columnist/member Rob Carmony as its people like you that are helping our newsletter become the sought after item that it is.

During March, local radio station here in Newcastle has provided HV99 with an unusual method of saying thank you to those great people at MICROpendium. Radio 2KO ran a promotion to let the people of the Hunter Valley know that a sister arrangement had city established between Lake close Macquarie City (which in i 5 proximity to Newcastle) and a town just out of Austin, Texas - yes you quessed it; Round Rock, the home of Having heard of this MICROpendium. the 2KO, I then approached manager of 2KO, John McGann, who was more than interested that a group of actually Novocastrians connections already with new sister city. With that, I asked him pass on a complete set to back-issues of the HV99 Newsletter to John and Laura at Mp as a gesture of thanks for their support of HV99.

Mention was made last month of our starved. non local information members. This being the case, and in a move to overcome this, editor (Brian) and publications librarian (Paul) have got together a couple of packages of overseas newsletters and put them in the Australia Post system with one bundle going Tasmania and the Steve Taylor in other to Geoff Shipton in Adelaide. starting point and as the This i S 3.0 country there are some or SO would like to have you members Ι contact me 50 that а circulation listing can be drawn up and your name included should you wish it to The list will be sent to Geoff be. and Steve for inclusion parcel. As yet I haven't 'chain' had any other suggestions on how the system will operate so this is the

lied rian your

Dr

35

Ti

/VBR

word

n to the

umns ight NEWS

are Niù. to

85 35 376

be

. 1 u b

. . . .

ial LEY Ist the the

ic or no by lng

. M2 S C D

of not XNS

i f you Please members have any ideas on it, let me us! know, as it is for YOUR benefit.

Two local (Aust) newsletters that Cheers to all till next month. have changed in format with their latest issues is the TND(Sydney) and the Melbourne Times. Both of these have considerable content and are The Illawara well worth reading. Group produced TND covers virtually all aspects of use in a very gutsy issue and full marks should go to the TIsHUG members that contributed, quality articles subjects and is, from memory, virtually all local content. The difference in format with this one positions the new format. Anyway the content usual requirement of a seconder in the mag. is tops.

Still on goings-on in Aust, avid members from Canberra, Chan, has been hospitalised recently hands. with a rather concerning illness. We were looking forward to a visit from Charlie and his family this joining the Committee after the AGM, month, however this has now been you are cordially invited to attend postponed. Maybe the vineyards of the next Committee meeting to be the Hunter would aid in his recovery held on Tuesday 5th May to see how - the grapevines around here are we work (or something!). their medicinal properties. Anyway, hurry up and GET WELL CHARLIE! from all of us ANNUAL SUBSCRIPTIONS fall due on 1st HV99ers.

Locally, we have received several enguiries on the 3 Slot PE Box project and unfortunately for those people we have completely sold the stock of our original order of 20 PCB's. A re-order is possible on this, however this will be dependant on the number of firm orders placed one of our members, but due to his over the next couple of months.

Just a reminder to our members that the AGM is in June, so it's time to consider what you can do to problems often involve repetitive help in the running of HV99. This calculations for which a computer is goes for our country people as well. | well suited. Whilst wrestling with You may not be close physically but such problems is not an everyday this group is as much yours as it is chore, occassional experimentation mine. If you want something done or can result in some surprising facts changed or whatever, get on the and figures.

us a letter - tell write country phone or

> Anyway that's about iŧ for

> > Albert Anderson . 4a4me

NOTICE

The Annual General Meeting of and to Bob and his helpers who got the Hunter Valley 99ers Users Group it between the covers quite nicely. | will be held on Tuesday, 9th June The Melbourne Times also contains 1987 at the Warners Bay High School on many commencing at 7.00pm.

Nomination forms for Committee are enclosed in this however comes in its packaging. It months Newsletter. Please fill them comprises about 40 odd pages of in, sign them and return to the computer fan fold pages and on first Secretary by Tuesday, 12 May 1987. impressions, comes as a bit of a Members outside the immediate area On investigation however, are invited to nominate people to time and money play a big part in stand for the Committee without the the signature of the nominee.

> goings-on in Aust, we ALL members are urged informed that one of our attend the AGM - remember, members are urged the Charlie future of the Group lies in YOUR

> > If you are interested

reminded Members are that July, so please have your membership paid up by then.

FACTORIALS

This article was written by modesty would prefer to remain anonymous.

Probability statistics bui ran sim TI4 for T14

TI for For fac acq wri COM Sim see

fac REA 101 2.4 cal thr kee int of.

sha

pro out 'di fic sin eas rol in i≢ а

6 He (A AB(are 10 che

ar 26 ch

the

ch 24 26

ma

ar pe 1e Th wh tell

now.

οf roup June thoo l

ittee this them the 1987. area to

the and to

the UR

in AGM. tend o be how

that 1st ship

by his main

tive r is with yday ition

acts

tics

Although the Ti-99/4a has some are in numeric functions, the combinations. TI40. The TI-99/4a would be great they still form just one 'hand'. for math calculations with an added TI40 keyboard and TO-FROM control.

A function not present in the TI is factorial! which is essential for some probability calculations. For those who haven't encountered n/r=n!/r!(n-r)! factorials (I have only a slight acquaintance), 3 factorial i = written 3! and is 3x2x1 (3*2*1) computerese) which equals Similarly 4! = 24. Whilst all this the combinations of 3 out of 26 seems simple, it's not long before 26!/3!(26-3)! or 2600. factorials result in BIG and then REALLY HUGE numbers. 8! is 40320. 10! 3428800, and 20! i s 2.4329E18. ТIЗØ Му and calculate to 69! (1.7112E98) throwing in the towel. And my 99/4a larger numbers, break into keeps trying up to 85! when it goes multiply into minor TILT. It really runs out exponents. of room for the exponent at 80! by showing the result as 7.15695EXX.

whose interested probability? I have used it to work 30 X=1 ou t odds and corresponding 40 C=1 'dividends' for a race game I was 50 X=X*C fiddling. with. There are some 60 C=C+1 simple probability events which are easy to evaluate - the chance of 80 PRINT "N!= ";X rolling a 6 with a single die is in 6 and two 6's with a pair of dice is 1 in 36 (6*6). a 7 (6+1,1+6,5+2,2+5,4+3 or 3+4) is cards well ABC, ACB, BAC, BCA, CAB and CBA. There 649740 letter which each choices for the second and one for the events occur. the third. Hence the 3*2*1=6=3!

Wading how arrangements of 3 letters out 26? First up there will be 26 to it? choose from and this will leave 25 choices for the second each time and for the third. And 26*25*24-15600.

Although three letters can be deeper. made up into six (we'll arrangements call permutations from now on) the letters remain the same each time. They form just one combination. whilst there are 15600 permutations of three letters out of 26. there

only 15600/6 or Likewise for five range is limited when compared to a cards dealt from a deck of cards simple calculator such as a TI30 or in whatever order they are received

> The formula for permutations of n objects r at a time is: nPr=n!/(n-r)!

> and for combinations Ωf n objects r at a time is:

So permutations of 3 out of 26 in are 26!/(26-3)!=15600. If 26*25*24 6. seems easier try 15 out of 55.

Of course it all becomes easy is only with the ! facility. TI40 knocked up a short (sub)program to before provide factorials up to 84!. two. mantissas and add

> 5 REM FACTORIALS 10 CALL CLEAR in 20 INPUT N (looking for N!) 7Ø IF CK=N THEN 5Ø

Now you can easily work out And the chance of that the number of combinations of 5 from of 52 a deck 6 in 36 or 1 in 6. So why ! you may 52!/5!*(52-5)! or 2598960. And with ask? Well - with 3 letters four Royal Flushes about, there's a (A,B,C) there can be 6 arrangements probability of one appearing every hands. are three choices for the first probabilities, the actual begins to time leaves two match the theoretical the more often

> You can also calculate many number of combinations of 6 numbers of in Lotto. Somewhat daunting, isn't

This article doesn't scratch the surface of probabilities and similar subjects, complex maths are needed to go much It's also surprising to different discover how random events can be them calculated to exact precision when three the subject is investigated further.

RANDOM BYTES

A potpourri of tips for BASIC, EXTENDED BASIC and FORTH users.

By Bob Carmany

I promised you some for this interesting material edition of the article so, without further delay, let's get started.

of the terminal program MASSTRANSFER rather simple example of a program term "freeware") and this version is could be quite easily added. been some obvious (and not made to obvious) changes program. been increased and some internal "glitches" have been fixed. If you have ever struggled with them there!

Let's qet down to programming and other "goodies". mine in '84). At that time, TI Have you ever looked at a program Forth was just a rudimentary idea! all character of statements defined. see what they look like.

100 CALL CLEAR :: INPUT "TYPE IN CHAR CODE: ":A\$ 110 CALL CLEAR :: CALL CHAR (35, A#) :: CALL HCHAR (12, 14, 35) 120 FOR DELAY=1 TO 2000 :: NEXT DELAY

125 DISPLAY AT(14,7):A\$ 13Ø DISPLAY AT(22,1): "ENTER ANOTHER CODE (Y/N)?" :: ACCEPT AT (22,26) VALIDATE ("YN"): B\$ 14Ø IF B\$="Y" THEN 100 ELSE 15Ø CALL CLEAR :: FOR DELAY=1 TO 1000 :: NEXT DELAY :: END

This short program will allow you you to enter the codes, one at a time, change character #35 into the proper character definition, and display both the character and the code you typed in on the screen. Displaying the code was to enable you to verify that you hadn't made any mistakes when you typed it. Of course, there is a prompt to allow you to enter a subsequent code First, there is another version before you quit. This is just a It's "fairware" (I hate the and there are enhancements denoted Version 4.1. There have that comes to mind is to merge the so data statements into the program and the have them read, one at a time, with The input buffer size has the standard READ C\$ statement.

Now, it's Forth time!! First of the all, you must realize that I am multiple key presses of Paul rather partial to Wycove Forth. It Charlton's FASTERM, you will really is much easier to use, has more appreciate the ease and convenience utilities built in, and runs most of this little beauty! If fact, benchmark programs much faster than several of the club members here the "Beast from Lubbock" -- TI have put their copies of fast term forth. It was the only non-TI on a shelf somewhere and just left produced language as far back as Versions 2.0 1983 (Version 1.0). and 2.1 (addenda and "fixes") were some issued shortly thereafter (I got and wondered what shape of a Now, we have Version 3.8 which is a those DATA truly "slick" bit of programming. It may be There are some words that I would easier for the programmer to use like to see defined in Wycove that them to define his characters but it just aren't there. GOTOXY is one sure makes life miserable for the that exists in TI but not in Wycove. rest of us. Where there a will, It's XB equivalent is DISPLAY AT. there is a way! This next program! There are three ways that the word does just that -- it allows you to can be defined for Versions 2.0 and 2.1.

> : GOTOXY SCREEN-WIDTH @ * + CURSOR-POS ! :

: GOTOXY SCREEN-WIDTH @ * + >0908 ! ;

: GOTOXY SCREEN-WIDTH @ * + [CURSOR-POS] LITERAL ! ;

with the 1148 1655 thou defi

defi Vers vali

memo For dρ Res 100 t o env

Wyc

sam The Sta the pla Ric

> HEX , D

por it doe

501 let the f = 0

the tai We

SA OL Si an a l

pr

Yσ

All three of these will work with Versions 2.0 and 2.1. However, the last example is the best one to $ROU(X) = INT(((10^2)*X) + 0.5)/(10^2)$ There is a with Version 3.0. lesson to be learned in all of this, though. The second Ωf these definitions uses a HEX value in the definition. As a result, when the Version changed, the code was not valid!

So, the lesson to be learned is try to avoid using absolute memory locations when programming in Forth. If the Version changes, so do some of the memory locations. Besides, i f you use absolute locations, the words are not going to be portable from one Forth environment to another (ie. TI to Wycove), for example.

llow

it a

the

and

the

en.

ble

iade

04

Low

ode а

iram

hat

One

the

and

iith

٥f

äM

Ιt

iore

iost

han

TI

-T1

as

2.ø

ere

got

TI

ea!

s a

nq.

uld

hat

one

ve.

AT.

ord

and

Here is another example of the same thing -- this time in TI-Forth. The first example is from Mariusz Stanczak (THE SMART PROGRAMMER) and the second has appeared a number φf '86 places including HV99 (OCT Richard Terry).

CREATE PICK (item# -item HEX C019 , 0A10 , A009 , C450 , 045F :: CALL SCREEN(2) . DECIMAL

: PICK 2 * SP@ + ;

Once again, the second is more | SGN(95-B) *A) :: NEXT A portable than the first. In fact, it works as easily in Wycove as it does in TI-Forth.

Now, we that have gone through the more complex issues, some of let's see what we can do for some of the BASIC programmers out there. In there may even be some out there who are still using an (ugh!) recorder to save programs. Well, this one is for you!

Have you ever meant to type in SAVE CS1 but accidentally typed in OLD CS1 instead? What do you do now? Simply type in SHIFT E and (ENTER) and you will get an I/O error -- and also a second chance to save the program!

Here is another way to round numbers in your programs. The actual "workhorse" is in line 110. can change the "2" in that line (each occurrence) to conform to the number of places you want the number to be rounded off to.

1ØØ X≃ 5.6883 DEF 11Ø 12Ø PRINT X,ROU(X)

have one more before I go. This little "gem" was released Vaughn Software some time ago (they are now out of business) for usage by Users' Groups here in the USA. I have found it to be relaxing and almost hypnotic screen I have thoroughly enjoyed display. it "dance" across my screen letting for the several years that I have an "oldie but it. Truly had goodie"! The name of the program, incidentally , is SNAKE DANCE. you run it, I think that you will see why they gave it such a name. Well, here it it!

100 REM ************* 110 REM * SNAKE DANCE * 12Ø REM #VAUGHN SOFTWARE# 13Ø REM ********** 14Ø REM 15Ø REM 16Ø REM EXTENDED BASIC 17Ø REM 18Ø RANDOMIZE :: CALL CLEAR 19Ø B=RNDØ :: CALL MAGNIFY(1) :: CALL CHAR(96, "B") :: FOR A=1 TO 28 :: CALL SPRITE(#A,96,8,95,10 ,10, 200 D=RND :: FOR A=1 TO 28 :: CALL MOTION(#A,D,A*SGN(1Ø-D)) :: NEXT A :: C=C+1 :: IF C<25 **THEN 200** 21Ø FOR A=1 TO 28 :: CALL COLOR(#A, RND+2) :: NEXT A :: C=Ø :: CALL MAGNIFY(2) 22Ø FOR A=1 TO 28 :: B=RND+2 :: CALL PATTERN(#A,96) :: CALL POSITION(#A,U,V) :: CALL MOTION (#A,SGN*(96-U)*9,SGN(10-V)*9) :: NEXT A 23Ø FOR A=1 TO 28 :: CALL PATTERN (#A,96) :: NEXT A :: CALL DELSPRITE (ALL) :: GOTO 190

The next time, we will some more Forth, a BASIC program that will calculate those simple interest loans for you. Of course, there will be some other tips, "goodies". tricks and other Once 'til again, next time from The States.

ASSEMBLY ZANGUAGE

FOR THE LAYMAN

WITH ALLAN WRIGHT, HV99ERS

Winter has just shot it's first warning shots across the bow. Temperatures are generally getting cooler and we have just had our first good rains so far this year on the Eastern seaboard. Where does this lead us? PERFECT weather for computing, too wet to work in the yard, too cold to go to the beach!

The Assembly Language classes are progressing along nicely, no major problems having thus far been encountered. As mentioned in the last article, chapters 1 through 4 of Molesworth's book, "Introduction to Assembly Language", were set for the group as "homework". The programme from chapter 5 was also typed in and assembled. At the last get together using Superbug discussed and it's capabilities shown. The procedure for modifying object code using the EDITOR was also discussed and demonstrated. These have all been described in earlier Assembly for the Layman articles. August to November 1986.

It came as something of a revelation to me that some of the people who attended the group asked questions that had been covered in those earlier article. Finally it was admitted that most of them had not articles. the So the questions, particularly on Superbug and on modifying object code were answered by reading sections Ω£ those articles.

The question of modification of object code (uncompressed variety) using the E/A Editor was given a fair amount of time. The problem that had been encountered was that after alteration, the code would not load for running - it was stopping on "CHECKSUM ERROR". The answer to the problem was that the CHECK SUM INDICATOR tag was not being changed from a 7 to 8. Indicator tag 8 tells the loader to ignore the check

sum for that particular line of code. The alternative to this is to modify the check sum. Information on doing this appears in the November 1986 Newsletter. One other word on this matter. After modifying the OBJECT CODE and then selecting the SAVE option answer NO! to the prompt. "VARIABLE 80 FORMAT Y/N".

Answering NO resoves the file in FIXED $8\emptyset$.

Now onto this month's article.

NON CONTEXT SWITCH BRANCH.

Three instructions can be used to branch to subroutines without causing a CONTEXT SWITCH.

BRANCH and LINK. BL

BRANCH. B

EXECUTE. E

BL and B will now be discussed.

BRANCH and LINK.

The instruction BL performs a non-CONTEXT SWITCH BRANCH. That is, the BRANCH is performed but the same WORKSPACE REGISTERS are used by the subroutine as those used by the CALLING programme.

When BL is performed the address of the instruction immediately following the BL is saved by being loaded into R11.

GETTING DATA INTO THE SUBROUTINE.

Using a similar technique to that used with the BLWP instruction, DATA can be taken into the subroutine by using the following code. In this case when the instruction BL is used, R11 is autoincremented whereas with BLWP, R14 is autoincremented.

Apar techi

SBRT

In tis SBRT subrat the (the BL). address the to the to the to the to the to the to the the transfer the transfer the transfer the transfer the transfer transfer the transfer transfer

add the DAT by sub to

inte

firs

ins >20 Usi int pro

sub sub app alt

BRA bra wit It ins

und you han Bed LII

pr

difference Apart from that technique is similar. The following code demonstrates this method.

> @SBRT BL DATA >0104,>2000

PROGRAMME

SBRT MOV #R11+,RO MOV *R11+,R1

RT

In this example the instruction BL is used to branch to the subroutine into Data is taken SBRT. subroutine by the MOV instructions at the beginning of the subroutine. When the BRANCH and LINK is executed This code causes an unconditional the RETURN address in placed in R11 BRANCH to the address contained in (the address immediatly following R11. That is, RT simply BRANCHES to BL). address placed in R11 points to the psuedo instruction RT saves you first piece of DATA >Ø1Ø4. By using typing B *R11 to RETURN from a the MOV instruction the DATA pointed to by the address in Rii is loaded This takes place on the DON'T DISAPPEAR. into RØ. first line of the subroutine. The address in R11 AUTOINCREMENTS to Without fail every book, series of then point to the second piece of articles or talks written on or DATA >2000. subroutine. Ril then AUTOINCREMENTS control of the machine. Your code the next instruction. to point to That i s the next programme DATA instruction following the

a subroutine allows write programmer to That is, the subroutines. application to tasks can change by address to be saved into Ril. The altering the DATA carried in to it.

BRANCH.

>2000.

BRANCH an causes branch to anywhere in your programme subroutine that was called using BL without causing a CONTEXT SWITCH. to similar the JMP It is BRANCH instruction. advantage Ωf causing an unconditional BRANCH to anywhere in back into Rii. That is unless YOU your programme. JMP on the other intend to do another BL before the hand has a limited range of control. Because BRANCH does not cause any YOU, the LINKING to occur, programmer, will be required to

include in your code to save any RETURN address you might have in since conserving mind. However memory is upper most in our minds it would be bordering on madness to write your own BL routine when one already exists for you to use.

RT --- A PSUEDO INSTRUCTION.

The instruction RT is used to RETURN from a subroutine that has been entered by the instruction BL. RT instruction causes programme flow RETURN to the address saved previously in R11. When you include RT in your assembly programme, the assembler supplies the same OBJECT CODE as though the SOURCE CODE had contained this code;

B *R11

In the above example that the address held in Ril. Using the subroutine.

This is MOV'd into R1 about ASSEMBLY language will mention by the second MOV instruction in the one inscrutable fact. You have total must be precise and exact. If not, your computer will seem to go off into a new dimension, where ever it is it, it definitely is not under Using this technique to get data your control. This can occur easily the if you are not careful when using BL universal from within a subroutine that has base already been accessed by BL. The subroutine does not alter but it's second BL causes it's correct return unfortunate part about this is that the original BL return address it lost and cannot be retrieved. So! here is the word of caution. IF you unconditional ARE going to BL from from inside a the contents of R11 MUST be saved before the second BL. After having the RETURNING from the second BL the saved RETURN address can be placed original subroutine is finished.



of s to tion the bther After

then

NO 4

DRMAT in

d to thout

s a is. SAME the the

Ωf itely eing

that DATA e by this ie

reas ted.

Sample code.

SAVRT1 BSS 2 SAVRT2 BSS 2

MAIN PROGRAMME HERE INCLUDING CODE TO SAVE E/A RETURN ADDRESS.

BL @SBRT1 DATA >Ø1Ø4,>2ØØØ

PROGRAMME

SBRT1 MOV *R11+,RØ MOV *R11+,R1

SUBROUTINE 1

MOV @R11,SAVRT2 BL @SBRT2

SUBROUTINE 1

MOV @SAVRTZ,R11 RT

K I

SBRT2 ----

SUBROUTINE2

RT

To save memory, unused registers can be used to save the RETURN address rather than set aside blocks of memory specifically for the purpose. The only constraint being that you do not have needed data already saved in that register and also of course the register is not used in the subroutine. Keeping these in mind BL can be used at will.

FINISH.

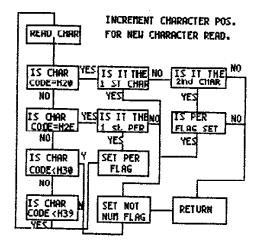
What is the ASSEMBLY group doing for the next couple of weeks???. Well the first project set for them has been to write a subroutine which can be accessed by BL. The subroutine is required to accept from the screen a valid number as defined below;

- 1) Any valid digit is acceptable ie. HEX 30 through to HEX 39.
- A null entry is not accepted. ie if enter is pressed before any digits entered.
- 3) A single period without any accompanying digits is not a valid

number.

- 4) Any input containing 2 periods is A S not a valid number. For
- 5) Any input containing one period and valid digits is acceptable.
- 6) Leading blanks are stripped from the input characters.
- 7) The number terminates at the first space found, except for (6).
- 8) Place the valid number in a BUFFER in CPU RAM.
- 9) If a non number is found leave the subroutine and leave a flag set to indicate that the entry was not a valid number.

A flow diagram of the routine excluding the section to strip leading blanks (do this later) is shown below:



This is fairly advanced for the stage that the group is at. However I do believe in jumping in at the deep end. The intention is to write useful subroutines that can be used in the future, the next logical set to follow this routine is a subroutine to evaluate the number placed in the BUFFER.

The code for the number validation routine will be included in the next article. I must point out here to anybody who has an attempt at this that the aim is to get a routine that works. The code you write may not look anything like the routine which appears in the next article. Something that works is the criteria at the present.



is much totas of Bri

vol

\$01

rea

rou

exe

be

iŧ

it

be

dat

⊏lu

spa

bla

A T

Our

Ωf

ext enj eve bed you

WO

US

Fo

d i

าต

MC

אמ

Ou

i s

 $\subset C$

ot

a

th

GF

αí

ods is

riod

from

(6).

ave 9 set not a

putine strip) is A SECOND FINISH.

For anybody who feels they are not for the number validation routine then here is another exercise. Write a routine which can be called by BL. This time however it is to clear the screen, but write it so that the area of the screen to be cleared can be varied by using data after the BL. A couple of clues, use VSBW and write >20 the space character where you want blank.

A THIRD FINISH.

Our Editor Brian is the unsung hero of this group. I feel sure that it is not generally appreciated how much time and effort he puts into the NEWSLETTER. Doing "things" for other people, which could they easily do themselves consumes a lot of his time. All I ask is that Brian be given a fair shake of the After all it is all done on a you want voluntary basis. Ιf something done and still want an excellent Newsletter then AT LEAST try to do it yourself first. enjoys working on the NEWSLETTER but even the most enjoyable things can become a pain in the ring gear if you are being used!

THE TI GOES BUSH

BU BEV WARREN

I was asked by Brian if I would do an article on some of the uses that my machine if being used for, as I believe it is a little different from the uses most people have.

teach course called а "Computer Awareness" which i 😅 by TAFE through a section called Outreach. So what is Outreach? It is a part of the college which take courses to the people who would otherwise not have ready access to a course on campus. In my case this. i⊊ cucal paopla living anything from a few miles to 100's of miles from a college campus.

Computer awareness is a very basic course teaching the participants (adults) about what a computer is, using mainly hands on teaching. It is presented in 2 Wez work with weekend workshops. Apple machines (yes the ones your machine ate for lunch today). what has this to do with TI? When the course was originally requested by the rural people in my area there was no course for computers. wished to study by unless they Correspondence. Most just wanted to actually work with a computer so the CA course was born.

I and two other teachers wrote the syllabus and took our course into the bush. Well, the first one was great. Lots of questions but we couldn't show them on the Apple as we didn't have programs. This is where my II comes in - I use it as a demonstration tool to show people a lot of basic uses for a computer in a small business, in the home etc. I also take along my TE2 for a little talking session by Brian the TI itself, always a success as it really breaks the ice when the student starts to respond to its questions!!!!!

I find the Apple a great teaching tool as it is a very simple machine with very clearly defined instructions, but even then you would be surprised how hard it is for some, mainly I think because they are still a bit wary of this thing called a Computer!!!

We also log on to Viatel to show some of its many uses.

Thy II really is good as we are lucky to have so many programs written for basic jobs eg cheque book balances, mailing lists etc.

So for we have not been able find many suitable programs for the Apple and this means we have to buy packaged programs which mean \$\$\$\$ and this restricts us. aven though packages are great, if they get too complex the student will not COME to accept the Computer for what it i < machine! They remain wary and this is what the course is about, to let them become confident and realize how it can help them in their every day lives.

the wever t the write used set

is a

umber

ation

next

e to

this

utine

may

itine

icle.

teria

The only thing my poor old II DOT like and that is. travelling around on the back seat of the car in all that dust!!!! No. our roads are not sealed like the city folks', only the highway, but is For isn't tourists. it!!!!!! The people we teach all live on one of the tracks, only experiencing the piensure driving on that blue stuff on the rare visit to town!! I guess Ron K from Walgett will know what I mean, or is he lucky enough to have not only a græat phone system. (microwave like our new one) but bitumen too??????

I also use my machine for making a lot of printouts of programs I have for basic business applications. I put these together so that the student can take these home, along with other printouts done in class, to see and study different applications ΟF the computer and to be able to then consider whether or not one would help in their everyday life, be it business or just in the home.

I would like to thank all the help they have given me (and will continue to give me I hope!) in the collection of simple programs that I can use. Wait till they see my next order, they may not accept my membership at renewal time due to a Great need of some sleep instead of answering that Cobar Indy's manuscripts full of 77777. Yes, they got another one today. Wait until I get my own Viotel Number!!

I also use the II to help the teaching of my children who the Air and do attend School of Correspondence. (That school ONLY 350 miles away!). It is used for maths drills, spelling, writing and sending messages usina ΤI Artist and Graphx to their teachers. Why not send the kids to school you say? Oh, the nearest is only 100 miles away, not far really you go 250 miles to get proceries is it!! My husband, (yes he is there, ever so potient, how I don't know!) uses it to keep his work records etc. He gets a turn when I am away teaching and the pirls are in bed. Quite good too since he didn't know anything about them until I got mine.

ΙF anyone has any programs which they think may be of help to me in any of the mentioned fields I am sure Albert or Brian Forward them on to me in the outback south of Cobar. (Approx 1 day further west than Walgett).

We don't even have a river out here unless it rains!!!! What is rain???Parhaps someone can tell me !!!!

DOUBLE DENSITY WITH YOUR

II DISK CONTROLLER CARD

This information comes to us from Bits Bytes and Pixels, the Newsletter of the LIMA 99/4a Users Group.

You can purchase a Double Density Myarc or Cor-comp Disk Controller Card for the PE Box for about \$US150, or you can have Ryte Data modify your existing TI card to Double or Quad density for much less. According to the November 86 issue of the Ryte Data Newsletter, Committee at HV99'ers for all the they will sell you the necessary chips for \$US45, or you can send them your disk controller and they |will install the chips for \$US45 + \$US10 labor and postage. You ship your TI card to them via parcel post, UPS or Purolator with your payment, and they will get it back in the mail to you within 48 hours of receipt. This should work with your existing TI or other single sided or double sided drives and allow you to store twice or four times the data on a disk side, compared with single sided format. To use Quad density, you need high quality disks. This looks like a good deal to me.

> Information can be obtained from: Ryte Data 210 Mountain Street HALIBURTON ONTARIO KOM 1SØ CANADA

> 99ers please note. our secretary, Albert, has written to Ryte Data for further information on this modification. Any reply will appear in the Newsletter, so stay tuned!

This expl I've stat RKEY

vers inte Dohr Cent gene proc accd rev Brid memo whic exte soul etc evei CODY

prof froi DaT P.Ø PAL CAL

COM

for **\$24** onl mУ

Ano ove Syd and For fro boa the

fas

For

the

bgrams lp to rids I would the rox 1 er out nt is

teil

80 o us the

Users

Double Disk ox for Ryte ard to much ber 86 letter, essary send d they 9845 + u ship parcel h your back hours with single s and r four side, ormat. high like a

btained

our en to ion on will o stay SCR #35 0 (APRIL-87/01) 1 STRUGGLING DMIJDDURTS STRUGGLING FORTH 4 5 HV99ERS APRIL ARTICLE 1987 EXPLORING VDP-RAM / PARTI 8 -STORING PROGRAM DATA IN VDPRAM 9 -SAMPLE PROGRAM -ASCII TABLE 10 -A LITTLE BIT ON BUFFERS 11 STRUGGLING BNIJBBURTZ 12 13 14

This month I thought I'd exploring the VDP chip.

I've had some feedback from the bulletins versions. interesting letter from Dohmann, from the Johnson Space Centre group. Ed Users generously has sent me a copy of his Edgar has SUPERFORTH, and accompanying manual. reviewing this in a later article. decompiler, which is easy to make, and is a nice specific permission was copy because it will not be Forth able to return your kindness. coming. (pardon the pun!). This product is copywrite and available Ok, now lets get down to business. from:

DaTaBioTics. Inc P.Ø. Box 1194 PALOS VERDES ESTATES CALIFORNIA 90274

only 15.00 and a couple of hours of interesting my time.

Another interesting snippet over here. A friend of Forth, has been downloading from same of the US Forth Bulletin boards on the east coast. printouts alone the USA. Not only that, Joe's jokes|the display to be impressive AND

start about FIFTH which we incurred at the last months meeting just happen to have some truth in them, as the smattered are states with help for us with the commentary about this extension of RKEY routine, from Robert Carmany, FORTH! My friend tells me he has who thoughtfully sent me several seen some directory references to I also received an the 994a and will download anything Edgar for us he sees.

very Incidentally , and I almost forgot, given me the permission to publish in the HV99ERS I will be the coding of his excellent whi⊏h Briefly, it needs an extra 8K ram of copywrited, providing I mention its memory in the cartridge port space origin from SUPERFORTH, and that granted. extension of TI-FORTH, including Thank you Edgar. I wish we had sound routines, hard disk support something of value we could give you etc. Before any of you try, don't in return. Perhaps some day when we even think about asking me for a become more proficient we will be

Why VDP RAM, I hear you all chorus? Well, since I've been playing with IBM XT's over the last couple of months, although I don't think a lot of the actual machine, and certainly for the very reasonable price of US|wouldn't buy one of the current \$24.95. Making the module cost me crop, I've noticed there are some programs about containing lots of windows (hence last months article) and a multitude for us of things that "pop up" onto your mine in screen in a window if you need them Sydney who runs a computer company! - you know the type: calculators, and just happens to be nuts about ascii charts, calenders, programming data aids, note pads etc.

Perusing So I sat down and gave it a few makes moments thought. In our case, one fascinating reading, and shows that cannot afford much dictionary space Forth is more than alive and well in for any associated code. One wants

fast.

any of you have experimented with I've deliberately made it heavily them, is the quickest way. But to store the data? In a previous article I've discussed using the buffer area, but that's usable RAM where one could put other perhaps even extra dictionary definitions. So I thought why not the VDP chip.

Perhaps in a later article we will take a visual journey through the contents of the chip and what all the different areas are used for. For today we'll just use a bit of its space and not wonder where or whv.

First I'd like you to use your imagination and miniaturise yourself, until you are so small you can walk along the etched tracks inside your computer from chip to chip.

this as a large room. Walk from user is awaited. this through a small area I'll call control the scrolling up and down by the the TRANSIT area, also belonging to lots of sixteen. This is entirely CPU RAM, down the tracks and into arbitrary, you can scroll them one pass the VDP RAM chip. Wander back if at a time properly within the box if you want to familiarise yourself you so desire. IE, function X will with the topography.

So why do we need a TRANSIT hall?

As I explain ad-nauseum, I'm not program. grounded in electronics, but my windowing routine of last month, you understanding is that one can't could then produce windows of data, directly address VDP RAM as one does scroll through it, and jump back ordinary instructions, such as C!, ! etc. Also, once in VDP RAM, to move data about one has to move it out of the MAKING THE ASCII TABLE IN VDP. area, into the TRANSIT area, and then back to the correct area in VDP One can: RAM. That's just the way its set up electronically.

Today in our program the CPU RAM area that corresponds to our TRANSIT area is the variable BUFR BUFR1. We move data from CPU RAM to the BUFR and then onto the VDPCHIP , or from DISK to the CPU RAM disk buffer areas and onto the VDP RAM chip and visa versa.

SAMPLE PROGRAM - ASCII TABLE.

I know one could type out a table, but taking the time to do

exercise one learns a lot about off programming. I've made no attempt repr Using the VDP reads and writes, if to crunch the code, to the contrary be m places As w commented and in many redundant, unnecessary down re-using commands. rather confuse amon than using beginners ЬУ stack|firs manipulators. The expanded code it t takes about 800 odd bytes of memory, crunched probably a lot less. When REFE you move onto a new generation computer this will be peanuts!

OVERVIEW OF THE PROGRAM.

Forgetting for the time being how it 65 4 gets there, the ASCII table from characters 32 thru 126 is kept in then VDP RAM from HEX 1400/DEC 5120 on, into occupying a total of 950 bytes, ten BUFR bytes for each character. This is to not within the dictionary space fair hence saves space. This area of the to VDP chip is not used by the system.

When the routine runs, the first the are flashed to othe sixteen characters We'll start from CPU RAM, imagine the screen and an input from the over The arrow keys we drop you down, function E up, and Function 5 will exit the routine, in we m our case back to the terminal, but last within a program, back to your Incorporated with CPU RAM Forth into your program where you left off.

1.Laboriously do it by hand on a Forth screen.

2.Use the computer to place it on Forth screen and load into RAM as needed.

3.After letting your TI994a do the work for you, incorporate the data into a definition within dictionary - Ok if you've got plenty of spare space, wasteful if you haven't.

MAKING YOUR COMPUTER WORK.

the While the masochists amongst you are

What

the on t

deci

depo

Incr prac take muli 100 don Ιf sim

won Man sta hap hav wha

ind unl can wri oft We

rat up!

about attempt ontrary heavily places cessary confuse | stack d code memory, When eration

how it e from kept in 20 on, es, ten his is space of the ystem.

e first hed to rom the w keys down by ntirely hem one box if X will p, and ine, in l, but your the th, you data, mp back u left

nd on a

t on a RAM as

do the he data the plenty you

you are

off scribing and calculating gaps to reproduce what's on SCREEN #49, lets be more cerebral.

As we discovered whilst wandering Ídown the copper plated highways amongst the chips earlier on, we first have to create the data, pass it through RAM and into the chip:

REFER TO SCR #36.

What this screen does is to generate the pattern we visually want to see on the screen ie eg:

65 41 A

into our TRANSIT area, in this case, BUFRI, and then passes this through fairly self explanatory. Just refer to lines 12 and 13. Each time the system do it automatically. decimal/hex and character is read from the screen it must be put in What RAM, in VDP the next address over the top of one another. the address:

pass 1- 512Ø 32 32 - 1Ø * + ie 512Ø 2- 512Ø 33 32 - 1Ø * + ie 513Ø etc.

We multiply by 10 to jump over the re-written to disk. address or data previously deposited.

Incrementing addresses using the loop index is a common and easy practice in Forth. BUFR to show you what's happening. when Many a time you'll just see a few alternativly if space permitted have to spent hours decoding just area. The effect is the same. what's being incremented by the loop index. Personally, I feel that space speed is unless ar al consideration, is better off one Anyway, very THE GRAPHICS. writing in longhand. often in Forth, as in this example, we have to slow the action down, rather than find ways of speeding it article, up!! (see DELAY)

SAVING FOR FUTURE USE.

Having got it there, one can either use that method everytime you boot the program, which wouldn't be a real hastle, (I hope you didn't blink as it was working or you will have missed it flashing past), or a much more elegant and intellectually satisfying solution - to save it to disk.

REFER TO SCREENS #47,48,49.

You don't have to enter the EDITOR made to get disk screens from or to disk. The word BLOCK does this for you. You can tell the computer to then it reads this from the screen load data into whichever buffer area you want, by 1st installing start address of the buffer in the to the correct VDP address. It's variable USE, but in our case as it doesn't matter, we can let

SAVE-CHART doing transferring the entire ASCII table otherwise they would all be written from the VDP RAM address, into one Hence of the buffer areas, the address of we use the loop index to increment which BLOCK has left on the stack. Naturally don't BLOCK load a screen on which you have written code, or you'll overwrite the code on it. UPDATE simply tells Forth that that something has changed in will it be buffer, 50 that If a buffer is not UPDATEd it will not re-write to disk, when later FLUSHED, or when the system wants to use the buffer space it occupies.

In this case we LOAD-ASCII is simply the reverse. take 65 off to produce an initial Conceptually this time our TRANSIT multiplication by zero(Ø). In other area is an actual disk buffer. The loops using a N Ø loop index we second definition of LQAD-ASCII will don't need this sort of technique. hold the data internally in the If you refer to SCR #44 you'll see a dictionary in the form of a string similar thing done. Forth code without a count byte. The !" won't always be as clear as this (string store) command loads all the because I've deliberately re-written data to the start of the buffer area this word is executed. stack operators swapping or dupping you so desired, you could move it happily to themselves, and you'll into an array within the dictionary

> Lets examine the rest of the program.

This is identical to last months using the single routine to draw lines around titles and the area of the screen we will write to.

THE INFRASTRUCTURE.

writing messages, validations etc. Of interest, is -! If I haven't being mentioned it before, the +! it decrements a opposite Ωf number from that already which exists in a variable.

The VDP addresses of TABLE-START and TABLE-END and SCREEN-VDP are constants, because their values must not change during execution

ADJ-UP ADJ-DOWN change the start! address in VDP RAM we will display the ascii data from, in increments programs and inadvertently forget to of 16 characters ie 160 bytes. They verify we haven't gone to far one way or the other and correct it we Hope have. Refer to screen 44. The word does all the co-ordinating that indefinate loop. Useful things those BEGIN...UNTIL loops! It simply wacks up at 16 lines at lightning and awaits your further instructions via ?ACTION. Like all such loops you can only exit it when the top of the stack is zero, hence the flag is left by ?ACTION. #43)

AND FINALLY....

Refer to SCR #50. Just to prove it into the VDPCHIP, did go included a couple ٥f SCREENS 39, 40, 41, 42, and 43, do words. EMPTYVDP simply wipes clear the back work of selecting keys, the area the data will go like a duster on a teachers blackboard. You can then use MAKE-TABLE to load your data, and then look at any VDP address you want to by entering the address and typing VIEW!

To run the program Type:

46 LOAD MAKE-TABLE DRAW-TABLE

ВУ way, please excuse the apparent descrepancies between the code and the comments, because I often grab routines out of other change minor details.

learn you beginners all something from this, because I sure have!

ADDRESS FOR CORRESPONDENCE :

RICHARD TERRY 141 DUDLEY RD WHITEBRIDGE 229Ø Ø49 436511/2245Ø.

FORTH Learners Group Report

The Forth group under Richard Terrys expert tutelage is going ahead in leaps and bounds (bounding forth), with his little band of converts battling on to past 11pm. most class nights.

At the last meetings we have continued our work on writing a word to accept a number input, and printing it back to the screen from the stack. Using Richards earlier defined word GET\$ to accept the input, the class then defined a word called INRANGE to check if the input was a number. This was the first introduction to the RETURN STACK. The word inrange was further refined and used in conjuction with another word (?DIGIT) to be able to check any range of digit input required, by passing the lower and upper ASCII values of the range of input wanted. From this point the class put all this together with the word ?DIGITS, which used (?DIGIT), ?DIGITS stepped through the inputed string byte by byte checking each byte to see if it is within range, and leaving a true flag if all bytes where digits. We ended up finally with a word called !NUMBVAL which used all the other words, and if it did not end up with a true flag on the stack, which was changed to a false flag by the word Ø= it did a MYSELF for the user to have another go. For the next lesson we have all been told to go forth and finish this off by writing a word to leave the inputed number string on the stack as a number, so the programme can manipulate it as a number. So all you budding forth programmers come along and have some fun learning. Brian R.

```
bve it
  I've
'ifying
               SCR #36
clear
                 O ( ASCII-TABLE - place in VDPRAM 18Feb87)
                 1 O VARIABLE BUF1 8 ALLOT
                                                         ( allot 10 byte buffer
like a
                 2 ( creates table of ascii 32->126 at hex 1400 in VDPRAM
kboard.
 load
                                                         ( expects nothing on stack)
                 3 : MAKE-TABLE
any VDP
                                  5120 1100 32 VFILL
                                                          ( blank vdpram area
                                                         ( loop 32-Blank to 126 ~ )
ng the
                                  127 32
                 5
                                  CLS 0 0 AT
                                                         ( so we can see it working)
                             DO
                 6
                                  I S->D 3 D.R 2 SPACES ( format decimal number
                 7
                              HEX I S->D 2 D.R DECIMAL ( format hexdecimal number)
                 8
                                                         ( print character to scrn )
                                  2 SPACES I EMIT
                 9
                                                         ( read from screen and
                                  O BUF1 10 VMBR
                10
                                                         ( transfer to transit bufr)
                                  BUF 1
                11
                                                         ( start of table in vdpram)
  any
                                  5120
                12
  the
                                                         ( increment approp# of byt)
                                  I 32 - 10 * +
                13
ause I
                                  10 VMBW
                                                         ( place each line in table)
                14
 other
                                                         ( leaves nothing on stack )
                             LOOP :
                15
get to
               SCR #37
                                                    10Feb87)
                 O ( REDEFINITIONS Characters
 learn
                 2 ( all characters needed to draw a single edged box
Isure
                 3 HEX
                 5 OOFF 0000 0000 0000 83 CHAR ( 131 SINGLE UNDERLINE
                  6 8080 8080 8080 8080 85 CHAR ( 133 SINGLE LINE RIGHT VERTICAL
                 7 0404 0404 0404 0404 86 CHAR ( 134 SINGLE LINE LEFT VERTICAL
                 8 0000 0000 0000 00FF 87 CHAR ( 135 SINGLE LINE BASELINE
                 9 8080 8080 80FF 88 CHAR ( 136 SINGLE LEFT BOTTOM CORNER
                 10 0404 0404 0404 04FF 89 CHAR ( 137 SINGLE RIGHT BOTTOM CORNER
                 11 FF00 0000 0000 0000 80 CHAR ( 128 SINGLE LINE TOP
                 12 FF80 8080 8080 8080 81 CHAR ( 129 SINGLE TOP LEFT CORNER
                 13 FF04 0404 0404 0404 82 CHAR ( 130 SINGLE TOP RIGHT CORNER
                 14
                 15 DECIMAL
               SCR #38
                 O ( GRAPHICS - Single frame hox 11Feb87)
                                                                ( CHECKED CLEAN)
                 1 ( same as PICK ie place copy of the nth number on top of stack)
word
the
                                                                 ( expects n
                          2 * SP@ + @ ;
ut,
                 3
was
                   ( routine to draw a box/rectangle -do not exceed screen size
                 4
he
                                       ( expects Leftcol/up row/Rghtcol/lower row )
                 5 : SBOX
                                                                ( upper left cnr )
er
                                                  1 129 HCHAR
                          4 P 4 P
                 6
d, by
                                                                 ( lower left cnr )
                 7
                          4 P OVER
                                                  1 136 HCHAR
ted.
                                                                 ( upper right cnr )
                                                  1 130 HCHAR
                 8
                          OVER 4 P
ITS,
                                                                 ( lower right cnr )
                                                  1 137 HCHAR
                 9
                          OVER OVER
te by
                          4 P 1+ GVER 4 P 1- 7 P - 135 HCHAR
                                                                 ( lower horizontal)
                10
а
                          4 P 4 P 1+ 3 P 1- 6 P - 133 VCHAR
                                                                 ( right vertical )
                11
ord
                          OVER 4 P 1+ 3 P 1- 6 P - 134 VCHAR
                                                                ( left vertical
                12
end
                          4 P 1+ 4 P 4 P 1- 7 P - 128 HCHAR
                                                                ( upper horizontal)
                13
9 Бу
                                                   3DROP DROP : ( leaves nothing )
                14
the
                15
Ьγ
```

a

```
SCR #39
                                                ( CHECKED CLEAN)
 O ( KEYBOARD - Key choice words 13FEB87)
  2 ( validates input, leaves true flag for valid key, 0 if invalid)
  3 : VALIDATE
                                       ( Expects 0 or ascil code
                  DUP 2DUP
                                       ( now have 4 copies of ascii)
  5
                  10 =
                           SWAP
                                       ( was it funct X:next
                                                                    )
                  11 = MAX SWAP
                                       ( was it funct E:previous
                                                                    3
  6
                                       ( was it funct 5: last screen)
  7
                  14 = MAX
                                       ( stack:flag/ascii code
  8
  9
 10 ( decrement contents at address by n
                                       ( exp n, adress with number
                                      ( subtract n from this
                  DUP & ROT - SWAP !
 12
                                       ( leaves nothing on stack
 13
 14
 13
SCR #40
  O ( KEYBOARD - Key choice words 13FEB87)
                                                 ( CHECKED CLEAN)
  1 ( selects user inputted choice, here up, down, or exit routine )
                                   ( expects nothing on stack
  2 : UP/DOWN/EXIT
              BEGIN
                                   ( start indefinite loop
  マ
                                   ( check for any key press 0 =none)
  5
                                   ( O=non valid/i=X,E,5,0,C pressed)
                    VALIDATE
                    IF i
                                   ( if valid, true flag to leave
                                                                    )
  6
                                   ( invalid, drop key copy, false flg)
  7
                    ELSE DROP 0
                                   ( continue rest of definition
  8
                    THEN
                                   ( until tests flag, do again if 0 )
  9
              UNTIL
 10
                                   ( leaves ascii code of valid key )
 11
 12 ( inserts a delay lengh according to number on stack
                                   ( expects nothing on the stack
 13 : DELAY
 14
              1500 0
 15
              DO NOP LOOP :
SCR #41
  O ( ASCII-TABLE -Variables/window 18Feb87)
                                       ( start position on screen
  2 241 CONSTANT SCREEN-VDP
                                       ( start of table in vdpram
  3 5120 CONSTANT TABLE-START
                                       ( end minus 160 bytes table )
  4 6060 CONSTANT TABLE-END
                                       ( current table adr pointer )
       O VARIABLE TABLE-ADR
                                       ( buffer to pass vdp->cpuram)
       O VARIABLE BUFR 10 ALLOT
  6
  8 ( puts up headings and draws a box around them and ascii chart)
                                       ( expects nothing on stack )
  9 : TABLE-BOX
 10
                  CLS
                   0 0 13 2 SBOX
 11
                   0 3 13 22 SBOX
 12
                   4 I AT . ASCII"
 13
 14
                   1 4 AT
                   .* DEC HEX CHAR* ; ( leaves nothing on stack
 15
```

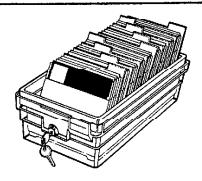
```
SCR #42
  O ( ASCII-TABLE - limits of table 18Feb87)
  2 ( checks have not overrun bottom limit /resets pointer if have)
  3 : ADJ-DOWN
                                       ( expects nothing on stack )
               140 TABLE-ADR +! ;
                                       ( increment table pointer
               TABLE-ADR @
                                       ( current table address
  5
                                       ( are we over end of table
               TABLE-END >
  6
                                       ( if so re-set back to end
               IF TABLE-END
  7
                  TABLE-ADR ! THEN ; ( and re-store to pointer
  8
   ( checks have not overrun top limit of table/re-sets if have
 10 : ADJ-UP
                                       ( expects nothing on stack
                                       ( decrement address pointer )
 11
               160 TABLE-ADR -!
 12
               TABLE-ADR
                                       ( current table address
 13
               TABLE-START <
                                       ( are we before the start
                                       ( if we are, re-set to start )
 14
               IF TABLE-START
 15
                  TABLE-ADR ! THEN ; ( and re-store pointer
SCR #43
  O ( SCROLLING - action up/down
                                     13F+697)
  1
  2 ( do we want to scroll up/down table or exit from routine
   ( the flag left is tested by UNTIL in the word .ASCII
  4
  5 : ?ACTION
                                       ( expects nothing on stack
                                       { get ascii code of choice
                  UP/DOWN/EXIT
  6
          CASE 10 OF ADJ-DOWN O ENDOF ( fn'x pressed =down
  8
               11 OF ADJ-UP
                              O ENDOF ( fn'e pressed =up
  9
               14 OF
                               1 ENDOF ( fn'5 pressed =exit
                                       ( leaves i or O flag
 10
          ENDCASE ;
 11
 12
 13
 14
 15
SCR #44
  O ( ASCII-TABLE - print to screen 18Feb87)
  1 ( prints the ascii characters 16 lines/loop to window area
                                       ( expects nothing on stack
  2 : .ASCII
                                       ( start of vdp ascii table
            5120 TABLE-ADR !
  3
                                       { start inedefinate loop
            BEGIN
                                        ( will display 16 at a time )
              16 0
                                        ( last point in vdp readfrom)
               DO TABLE-ADR @
                  I 10 * +
                                        { increment it to next line }
  7
                                        ( transit area to deposit it)
  8
                  BUFR
  9
                  10 VMBR
                                        ( read one line into bufr
 10
                  BUFR
                                        ( address of transit area
                  SCREEN-VDP
                                        ( vdpadr of top left row/col)
  1 1
                                      . ( start of next screen line )
                  I 40 * +
 12
                                        ( write one row of the table)
                  10 VMBW
 13
                                       { want to up/down/exit?
 14
                LOOP PACTION DELAY
 15
             UNTIL (
                                        ( O = exit stack empty
```

```
18Feb87)
 O ( ASCII-TABLE
                    Draw Table
 2 ( this expects a table to be present in VDPRAM at hex 1400
 3 ( either using make-table or loaded as saved from disk
                                       ( expects nothing on stack
 5 : DRAW-TABLE
                                       ( draws box/headings
                                                                    )
                  TABLE-BOX
                                       ( puts contents into box
                  .ASCII
 7
                                       ( leaves nothing on stack
 8
 9
 10
 11
 12
 13
 14
 15
SCR #46
 O ( ASCII-TABLE - Load screen
                                     19Feb87)
  1
  3 : LOAD-ALL 46 36 DO I LOAD LOOP ;
  5
      LOAD-ALL
  6
   ( NOTE: To load ascii table from disk, include the definition
  8
            LOAD-ASCII, currently on scr# 48, at the start of
  9
 10
            any program using this chart, or the LOAD-ASCII
            definition on scr 47 to block load without a
 11
            definition )
 12
 1.3
 14
 15
SCR #47
  O ( ASCII-CHART - saving to disk 18Feb87)
  2 ( transfers ascii table from vdpram to buffer then onto disk
                                        ( expects nothing on stack
  3 : SAVE-CHART
                   49 BLOCK
                                        { disk->memory,leave addr
                   UPDATE
                                        ( mark as updated
  5
                   5120
                                        ( start of asc table in vdp )
  6
  7
                   SWAP
                                        ( stack now vadr, adr scr 49 )
                                        ( transfer table to scr 49 )
  8
                   950 VMBR
                                        ( write to disk, stack empty )
  9
                   FLUSH I
 10 ( loads ascii table from disk into vdp-ram
                                        ( expects nothing on stack
 11 : LOAD-ASCII
                                                                     )
                   5120 1100 32 VFILL
                                        ( blank vdp area
 12
                   49 BLOCK
                                        ( load chart.leave block adr)
 13
 14
                                        ( start address to put it
                   5120
                                        ( transfer, stack now empty )
 15
                   950 VMBW
```

SCR #45

```
O ( ASCII-TABLE - load from disk 18Feb87)
  1 ( loading data to vdparea via a definition
  2 : LOAD-ASCII
                                  ( expects nothing on stack
                  USE @ DUP
                                  ( next available disk buffer)
              ( store the following data into this )
  5
    65
        41 A 66 42 B 67
                         43 C 68 44 D 69 45 E 70 46 F 71
    47 G 72 48 H 73 49 I 74 4A J 75 4B K 76 4C L 77 4D
    M 78 4E N 79 4F 0 80 50 P 81 51 Q 82 52 R 83 53 S 8
  8 4 54 T 85 55 U 86 56 V 87 57 W 88 58 X 87 59 Y 70 5
  9 A Z 91 5B [ 92 5C \ 93 5D ] 94 5E ^ 95 5F
                                                   _ 96 60 '
 10
    97 61 a 98 62 b 99 63 c100 64 d101 65 e102 66 f103
 11
    67 g104 68 h105 69 i106 6A j107 6B k108 6C i109 6D
 12
    miio 6E niii 6F oli2 70 pii3 71 qii4 72 rii5 73 sii
 13 6 74 til7 75 ull8 76 vll9 77 wl20 78 x121 79 y122 7
 14 A z123 79 {124 7C {125 7D }126 7E ~ *
 15
                  5120 d30 VMBW ( write this data to vdp area)
SCR #49
  0 32 20
             33 21 ! 34 22 * 35 23 # 36 24 $ 37 25 % 38
  1 26 & 39 27 ' 40 28 ( 41 29 ) 42 2A * 43 2B + 44 2C
    , 45 2D ~ 46 2E . 47 2F / 48 30 0 49 31 1 50 32 2 5
  3 1 33 3 52 34 4 53 35 5 54 36 6 55 37 7 56 38 8 57
  4 7 7 38 3A : 37 3B ; 60 3C < 61 3D = 62 3E > 63 3F
    64 40 @ 65 41 A 66 42 B 67 43 C 68 44 D 69 45 E 70
    46 F 71 47 G 72 48 H 73 49 I 74 4A J 75 48 K 76 4C
         4D M 78 4E N 79 4F 0 80 50 P 81 51 9 82 52 R 8
  8 3 53 S 84 54 T 85 55 U 86 56 V 87 57 W 88 58 X 89 5
      Y 90 5A Z 91 5B [ 92 5C \ 93 5D ] 94 5E ^ 95 5F
           ' 97 61 a 98 62 b 99 63 c100 64 d101 65 e102
 10
    96
        f103 67 g104 68 h105 69 1106 6A j107 6B k108 6C
    1109 6D m110 6E n111 6F 0112 70 p113 71 q114 72 r11
 13 5 73 si16 74 t117 75 u118 76 v119 77 w120 78 x121 7
 14 9 y122 7A z123 7B {124 7C | 125 7D }126 7E ~
 15
SCR #50
 O ( STRUGGLING FORTH-APRIL)
 1 ( fill the designated vdp area with blanks
 2 : EMPTYVDP
                                 ( expects nothing on stack
               5120
 3
                                 ( designated start addr vdp )
 4
               950
                                 { lengh of ascii chart
 5
               32 VFILL
                                 ( fill area with blanks
 6
                                 ( stack empty
 8 : VIEW
                                 ( expects a vdp address
 9
               BUFR
                                 ( transit buffer to put data)
 10
                10 VMBR
                                 ( place ten bytes there
                                 ( printout buffer contents
 11
               BUFR 10 TYPE
 12
                                 ( leaves stack empty
 13
 14
 15
```

SCR #48



SOFTWARE LIBRARIANS NEWS

BY AL LAWRENCE

99'ers, HI

*********************** ??? × **HAPPINESS** IS

Happiness is a D.I.Y Horizon Ramdisk in the PE Box, all the LATEST furry Funnelweb creatures little residence, in a microsecond ready to work quietly with no whirling disk Life here in the Dungeon is easier now that I (meself that is) took the 'tiny' soldering iron in hand and tackled the task with some But I need not have been worried, as the manual supplied with the board is so easy to follow. Everything clicked into place as described, the most difficult part picking out the red and green LEDs, as to my untrained eyes our supplier had given me 2 White ones! Thanks Albert for the efforts in getting everything for all of us at the best possible price - much appreciated by all. Not to worry, just go ahead and if you believe in in the luck of First test showed a tiny the Irish! GREEN (Well what else would you expect?) light just as the good book lo and behold, when all said, and the chips were down a big RED followed. Wow! Fantastic. Viola... Happiness.

THE MORE YOU PAY... THE MORE YOU GET

DONATION SOFTWARE.

A new software concept from Steve Risner of Tustin, CA. II.S.A. also a new name for TI'ers instead misused much the of FREEWARE/FAIRWARE

RECORDS/PLUS.

A lot of thought and work (7 MONTHS) makes this an extremely versatile menu driven data base on a SS/SD disk in XBasic. It will handle 10 books and 200 pages of 9 fields Can be expanded to 50 books (for a few dollars more).

Full screen editing, help screens, set screen colour at title screen, flexible sorting and printing to Sun allow you customised printed report Disk lists, labels, etc.

The disk can be duplicated and the DRAG data is stored on the program disk from thus there is no need for 2 drives SLEE or swapping of disks.

If write protect tab left on ADDR disk you will not get past title prin screen as it will not be able to HEX. access the files on the disk.

There are no instructions, but screen at various stages informs you of d what is available as optional extras for the sums of \$5, \$10, \$15 or \$20 Full In fact disk donations to the author. the Demo book is made up of 13 records which you can print out FUNN (singly or in 3 rows) the whole comments. Why message and requests for If donor response is good a Utility so m Spanish, some versions disk and in Mail Merge, Move keer French, German. page/book,disk/disk may appear have Printing Option of books gives you on choice of Entire Book, Partial Book cust or Single Page, Dated. Lets you view sample printout on screen 3 across whil and 3 lines down, with or without you fields.Fully sorted.

After printing Menu gives 4AII you choices. feat 1.Back to print option menu. SL. 2.Print again (duplicate) 15 3. Back to master menu. DV/8 4.Call it quits. COM

Ther can

Duri but the all

You Some Inco Car

pres

on t

lot may pay

Vide

DS/S incl

Prin

loca

FORM the prin

of i

file incl Steve and stead

sused

NTHS) atile SS/SD le 1Ø ields books

eens, reen, ig to eport

the n disk drives

eft on title ble to

t the ms you extras or \$20 fact of 13 t out whole ments. tility anish, , Move appear ves you Book bu view across without

you 4

There are 7 sample books and 3 you can play around with.

During usage Function Quit disabled but on leaving Records+ message of the Donation scheme on screen while all open files are closed then the press Function Quit To Exit appears on the screen.

You may be able to make use of it. Some suggested ideas: Income Tax Records. Car Maintenance. Video Tape labler.

All in all a good effort amongst lot of competition in this field and may suit some, but you gets what you pay for as the man said.

** New Disks **

Sun City 99/4a H.G.

programs and files Disk of on a or 2 SS/SD disks. Disk includes an XB Freeware game called DRAGON STORM - Knight rescues Maiden from castle game. SLEEVE3.

Prints disk sleeves with contents or catalog only.

ADDRTABLES.

Prints reference listing table of HEX, DECIMAL, BINARY Memory address locations.

print in 2 or 3 col format. Heaps of demo inst. files are included.

Full instructions included on the disk for all programs .

FUNNELWEB.

Why not update yours now. so much more in it. If you are like some people who said its TOO much to keep CUSTOMISING the LOAD then you have not been reading the FWDOC/REPT on MERGING the LCAD you have customised already. READ IT NOW!

you know you are 5 spaces from end of line.

All KNOWN Bugs eradicated and a new tickets with you. feature includes: SL. Batch File LOADER allows up 15 object files to be specified in DV/80 script, possible to load combination of RE Locatable object files that E/A will load.Auto Start included.

SAVIT.A simple Demo file is also included on the disk.

All the Club Software is available on demand - \$4.00 P/P each disk.

*********** ¥ * Special deals on BULK orders

BASICly if you have EXTENDED the BASIC then go FORTH and ASSEMBLE and to PILOT how your way ADVENTURING and start HACKING up the old TI 99/4a.

> Keep on Computing. Al Lawrence.

WIN! WIN! WIN!

ın THE HU 99ER5

RAPPLE

The three monthly raffle FORMAT.use with TI WRITER files to drawn last month and was won by RON PRATT, who took away a pack of 10 disks. The next draw will be held at the Annual General Meeting on 9th June.

Entering the raffle is easy. For each \$1 you spend at the club diskthrough purchases, monthly There is meetings etc, you receive 1 ticket. These all go into the barrel and 1 prize of either 10 blank disks or 10 blank cassettes is drawn each three months. Non-local members purchase software from Al da nat miss out - their name is written the stub so that if their number While in edit mode a BEEP now lets comes out they win! Local members must be in attendance to claim their prize, so make sure you come to the meetings and bring your

ELECTRIC CIRCUITS part 2 by Paul Mulvaney

This program is to be added to part 1 published in the February issue of HV99er. Where part 1 had one switch controlling a light part 2 now has two switches controlling the light. This circuit is used to allow the light to be switched on at one side of a room and off at the other side, particularly useful in hallways or rooms that are passed through. Operating either switch will cause the light to change state, ie if ON operating either switch will turn it OFF and vice versa. Remember add it to part 1 as it is not a stand alone program.

43Ø CALL CLEAR :: REM TWO WA Y LIGHT CIRCUIT 64Ø CALL CHAR(113, "CØEØ7Ø381 CØEØ7Ø3"):: CALL CHAR(114, "F FFFFF"):: CALL CHAR(129, "CØE Ø7Ø381CØEØ7Ø3"):: CALL CHAR(13Ø, "Ø3Ø7ØE1C387ØEØCØ")

65Ø ! 66Ø DISPLAY AT(3,4): "TWO WAY LIGHTING CIRCUIT* :: RESTOR E 71Ø

67Ø DISPLAY AT(8,5): "SW1 SW2" :: DISPLAY AT(20,5) : "PRESS 1 FOR SWITCH 1" :: D ISPLAY AT(22,5): "PRESS 2 FOR SWITCH 2'

68Ø DISPLAY AT(24,5):"PRESS 3 TO END"

690 FOR L=1 TO 4 :: READ R,C ,K,T :: CALL VCHAR(R,C,K,T): : NEXT L

700 FOR L=1 TO 3 :: READ R,C ,K,T :: CALL HCHAR(R,C,K,T): NEXT L

710 DATA 6,4,97,12,6,29,120, 12,6,6,65,1,6,27,78,1,12,5,9 8,3,12,25,122,1,12,26,121,3 72Ø FOR L=1 TO 14 :: READ R, C,K,T :: CALL HCHAR(R,C,K,T)

:: NEXT L 73Ø CALL KEY(Ø,N,S):: IF S(1

OR N<49 OR N>51 THEN 73Ø 74Ø CALL GCHAR(12,9,SW1):: C ALL GCHAR(12,18,SW2):: N=N-4 8 :: ON N GOTO 750,800,740 75Ø IF SW1=32 THEN 78Ø

76Ø IF SW2=32 THEN RESTORE 9 20 ELSE RESTORE 860

770 GOTO *720*

78Ø IF SW2=32 THEN RESTORE 9 ØØ ELSE RESTORE 88Ø 79Ø GOTO 72Ø

800 IF SW2=32 THEN 830

81Ø IF SW1=32 THEN RESTORE 9 2Ø ELSE RESTORE 9ØØ 82Ø GOTO 72Ø 83Ø IF SW1=32 THEN RESTORE 8 *40 ELSE RESTORE 880*

84Ø GOTO 72Ø 85Ø PRINT "HELP"

84Ø DATA 12,8,32,2,11,8,13Ø, 1,10,9,130,1,10,10,98,8,12,1

0,104,8,12,18,114,2,10,18,32 1,11,19,32,1 87Ø DATA 12,2Ø,1Ø4,4,12,24,3

7,1,11,24,105,1,10,24,107,1, 10,25,108,1,11,25,106,1 88Ø DATA 11,8,32,1,1Ø,9,32,1

,10,10,104,8,12,8,128,2,12,1 Ø,98,8,12,18,128,2,1Ø,18,32, 1,11,19,32,1

89Ø DATA 12,2Ø,98,4,12,24,96

,1,11,24,136,1,10,24,138,1,1 Ø, 25, 139, 1, 11, 25, 137, 1

700 DATA 11,8,32,1,10,7,32,1 ,12,8,128,2,12,10,98,8,12,18 ,32,2,11,19,113,1,10,18,113,

1,10,10,104,8

91Ø DATA 12,2Ø,1Ø4,4,12,24,3 7,1,11,24,105,1,10,24,107,1,

10,25,108,1,11,25,106,1 920 DATA 12,18,32,2,11,8,130

,1,10,9,130,1,10,10,98,8,10, 18,129,1,11,19,129,1,12,20,9

8,4,12,24,96,1

93Ø DATA 11,24,136,1,10,24,1 38,1,10,25,139,1,11,25,137,1

,12,8,32,2,12,10,104,8 94Ø PRINT "MORE NEXT MONTH" :: FOR DELAY=1 TO 500 :: NEX

T DELAY :: PRINT "BYE"

INIR

real Resi

Tuto this Fort 'oth The

prog

appl

thos

Data

intr info

I with the need poss ₩₽₩ αf play not was coul

lanc

COMM

mana

Forth users out really Forth then Besides Richard's usual *Tutorial, we are happy to include* packages, both commercial Forth Data Base, written by our requirements. should I Data Bases in general ...



FORTH DATA BASE

A PROGRAM BY KEITH BRUCE

INTRODUCTION

ht

ţ.h

it

This article will give introduction to organized information storage and retrieval.

When I first purchased my TI99 I very quickly realized with the withdrawal of Texas Instruments from the Home computer market that I needed to obtain as much software as composed of fields e.g. address. possible to run on the TI99. There was at that time a plentiful supply of games available. was needed was software that you persons name would be the key field. could do something with e.g. languages, editors, spreadsheets, communications packages and database records is small then any sorting or

th users out there are two readily available languages, but looked after this month. after that there wasn't much. Since there has been this month the first of a 2 part | fairware, released to fill the above However *'nther' Forth expert, Keith Bruce.* | managers have been pretty scarce for The documentation accompanying the the TI99 up until recently. Now I say had always promised myself that I application?) will be of interest to would have an attempt at writing my those wishing to learn more about own database manager. But first I had to learn enough about FORTH to be proficient enough to do a task like that. Well after about two years of playing around with FORTH I decided to have a go. What follows is a discussion on the technique I have used in managing the data and a discussion of the FORTH code to do

> Because of the large number of screens of code, about 50, I will present half this month and the other half next month.

DATA_ORGANISATION

To be able to store data efficiently some retrieve organising scheme has to be used. Now there are many of these around and there have been many books and papers written on these techniques relative merits and performances.

First a definition of few terms. A database is a collection of related items or information. It may be a list of telephone numbers with the person's name and address. You may want to retrieve the data sorted in alphabetical order on which is usually not the way in which you entered the information in. So the database must be arranged, or sorted, so that the items, or records are retrieved in alphabetical order.

Each item of information termed a record e.g. name, address and phone number. Each record is

The field on which the database However once is arranged or sorted on is referred played several times most games did to as the key field. In the above not hold the users interest. What example the field containing the

Of course if the number of managers. Now TI basic and XB were searching of the data is just as fast as doing it on all the data.

Once the amount of data that is with what you are looking for. stored in the database exceeds the it is then stop. Otherwise divide memory capacity of the computer then the list in half and pick the upper it becomes inefficient to sort the or lower half depending upon whether whole database. If the number of the desired key is less than or records is large then only a subset greater than the one found. Now of the data can be memory at any check the middle item of that part time and any sorting or searching of of the list and continue repeating the data must involve a lot of disk this procedure until a match is reads and writes.

The method i have implemented to arrange records to overcome this problem is known as Indexed file access.

INDEXED FILE ACCESS

data file. By keeping this list means that for 1000 records sorted and in memory the time taken sequential one disk read. To find a record a binary search of the index list is done, then using the record number a IMPLEMENTATION random access file read is done on the data file.

much smaller than the record itself the index list does not have to be then no advantage will be gained rebuilt by reading every record of using this approach. Problems can the data file. The disadvantage of also arise when deleting records. A doing this is that the index file simple approach to deleting a record may get out of sync with the data is to delete the entry in the index file. This could happen after list, then delete the record in the amending the data file and by the data file and move all the records program aborting or by the power that follow it forward by one being removed from the machine, that follow it forward by However this can result in a lot of accidentally or otherwise, before disk reads and writes. As the data exiting normally. In the first case file isn't sorted anyway a more I have tried to design the program efficient technique is to move the so that it is nearly impossible to last record in the file to the crash it. But someone will probably deleted record location. also presents a problem in that the in the second situation there is no file still contains n records but amount of code that anybody can only n-1 active records. The nth write to overcome that problem. record doesn't mean anything now. Hence some method of keeping track of the active records in the data file is out of sync with the data file is required i.e. a counter file then the index can be rebuilt variable. file is closed and later reopened at file and then sorting the key another session the counter may be fields. incorrect. To overcome this the number of active records in the data <u>RECORDS</u> file must be written into the data file before it is closed. Record read into, and written from BUFR. zero has been chosen to store the This variable is defined at the number of active records.

WHY USE A BINARY SEARCH

In a binary search the middle item of the list is first compared found or the list is only one record long, in which case no match found.

The advantage of using a binary search over sequential access is On average a sequential speed. search will require n/2 comparisons The index list contains the key for a list of n sorted records. field of each record and the binary search requires at most nlog2 corresponding record number in the comparisons. In real terms this requires search on to access an individual record can average 500 comparisons and a binary be done in the time it takes to do|search will require at most 10 comparisons.

To improve the performance the index list is saved to file as well. This substantially reduces the time However if the keyfield isn't involved in reopening a data file as But this manage to prove me wrong.

If it is found that the index However when the data by reading every record of the data

Each record of the data file is start of the program and the length can a di pres pres may thro offsi I ho late memo

> auto data suff suff

EILE

File This file

Data Inde

driv to:

file **FNAM**

the

from

Upon open list inde reco

file DATA

acce indi addi scan for

NUME

dist

reco

nume are and done that wou I the with

recd Howe key

can be altered, if desired, to suit a different record structure to that presented here. However with the present version of the program this may require other modifications throughout the program to suit offsets for different length fields. I hope to overcome this problem in a later edition if there is enough memory to fit it all in.

EILE_NAMES

The name of the index file will automatically be the same as the paragraph on problems below) data file but it will be given a return appropriate error suffix of _DT as shown below:

File Name entered: TEST This will create the following two minimal use files:

Data file name: DSK1.TEST_DT Index file name: DSK1.TEST_IN

For those who wish to access drive other than DSK1 they will have to edit screen 244, line 4.

Once entered the names the files are stored in the variables FNAME and INDNAME.

If the data file exists then index information will be read from the index file into memory. quitting the records will be written to the data file.

DAIABASE_FACILITIES

Currently the οf the database individual record manipulation modifying, deleting and scanning or viewing records - and for printing a report containing all record in index order, i.e. sorted.

NUMBERS

This program does not distinguish between data that is numeric and normal text. Numbers are treated as strings of characters and there is no numeric processing done on them. This does not mean that it cannot be done. In fact it would be relatively easy to amend suit your own record structure and to handle the code numeric data without having to affect the basic record reading and writing routines. However problems may arise if the 14 PASADENA CRES. key field is numeric and situations CARDIFF

such as left or right justification, and whether or not leading zeros are not handled removed. are effectively.

ERROR_HANDLING

Most of the primitive routines that involve disk access have error handling built into them and call either CHEKSTATUS or CHEKFILE to do the error checking. These two words check the status bytes (this will be discussed further detail in suffix of _IN and the data file a They also print an error message if the error code is not expected.

> However at the moment only is made of the error code returned to the higher level This words. could however be altered by creating another word that checks the returned error code and takes action as necessary.

PROBLEMS

The two biggest problems getting this program to work where They to do with the disk system. be peculiar to the CORCOMP may system but I will present them here so as to save others of many wasted The outcome of the first hours. problem was that the file attributes program, or (e.g. INT/FIX 255) are not set when opening another database, the index | a new file is created (on creation list will be written back to the they are DIS/FIX Ø) but when the index file and the number of active first record is written to the file.

> The second problem arose from trying to detect if a file existed previously or not when it is opened. program allows If it did exist then the first for record is read otherwise information is written to the first record. Simply checking the status byte (837C Hex) and the PAB Flag/Status byte on opening the file gave the same status for a new or existing file. The only reliable solution was to use the FORTH word STAT (page 56 of Appendix D, TI FORTH manual) and check the byte returned.

NEXT MONIH

The rest of the program will be presented next month together with some notes on how to modify it to some ideas for improvements.

KEITH BRUCE PH. 547258

ared Ιf ride boer

dle

ther or Now bart

ing i S tord Was

ary is ial

ons Α 092

his a on ary

10

the 11.

time 35 be of of

ile ata ter the

wer ne. ore ase

iram tο bly ver

na can

dex

ata illt ata key

is FR. the ngth

```
SCR #240
    O ( DATABASE
                                 19Feb87 )
         11 WIDTH! ( set length of names in dictionary to 11 )
   3 68 CLOAD STAT ( load FILE options )
4 ! \ IN 2 64 / 1+ 64 + IN ! ; IMMEDIATE
5 \ 230 CLOAD N ( INTACCEPT )
    6
      : 00 ;
   7 200 CONSTANT INDMAX
8 30 CONSTANT INDLEN
                                        \ max no of records index can handle \ length of indexed key field
                                    \ length of data file record
   9 255 CONSTANT BUFFLEN
  10 INDLEM 2 + CONSTANT INDRECL \ length of record in index file
       O VARIABLE BUFR BUFRLEN ALLOT \ buffer records are put in O VARIABLE INDBUF INDLEN ALLOT \ temp buffer into which index O VARIABLE INDIANCE 20 ALLOT \ is deposited O VARIABLE FNAME 20 ALLOT
  13
SCR 1241
   O ( DATABASE
                                18Feb87 )
       O YARTABLE MINI
        O VARIABLE )FNAME
        O VARIABLE RECLEN
      O YARIABLE MAXREC
O YARIABLE KEY OFFSET
O VARIABLE FNAMEFLAG
O YARIABLE LO O YARIABLE HI
O YARIABLE ADI O YARIABLE ADZ
      -2 CONSTANT FIL ERR
-1 CONSTANT NEWFILE
0 CONSTANT NO ERR
1 CONSTANT RECHOTEND
2 CONSTANT NO ROOM
        3 CONSTANT EXISTS
        4 CONSTANT OUTRANG
  15
                                        -->
SCR #242
   Q ( DATABASE
                               18Feb87 )
   5 HEX
         PABS 2 A + BUFR 1700 FILE DATPAB
                                                            \ create PAB for DATA
         PABS 2 28 + INDBUF 1800 FILE INDPAB
                                                            \ create PAB for INDEX
   8 DECIMAL
   Ŧ
 10
 11
 12
 13
 11
 15
SCR #243
  O \ DATABASE
                               18Feb87
  OVER CO O= IF DROP DROP DROP 0 \ n=chars actually entered ELSE >R SWAP R ROT ROT R)
                  CHOVE ENDIF ;
            DNP 96 > \ ( n1 -- n2 ) cor
IF DUP 123 < IF 32 - ENDIF ENDIF }
  B : UPC
                                 \ ( ni -- n2 ) convert ni to upper case
 11 : Y/N BEGIN KEY UPC DUP CASE
           89 OF DROP 1 0 EMDOF \ if Y then true
78 OF DROP 0 0 ENDOF \ if N then false
 12
 13
          ENDCASE O= UNTIL ;
                                            \ repeat until Y or M entered
 15 -->
```

```
SCR $744
  O \ DATABASE
                          18Feb87
  DUP 0= IF DROP 0
ELSE DUP FNAME + 6 +
                                      \ if no chars entered leave false
                                      \ add length of drive name
                1" DT" B +
  8
                                      \ add file type (data) to end
  Ÿ
                DUP DUP FNAME C!
                                      \ save length of filename at start
 10
                FHAME INDHAME ROT
 11
                CHOVE
                                     \ copy filename to index name buff
                2 - INDNAME +
 12
                                     point to start of file type set as index type
 13
                I ERDIF
 14
                                    . \ set fnameflag to true
 15
          FMAMEFLAG ! : -->
SCR #245
  C \ DATABASE
                          19Feb87
    : SETUP PAB
SET-PAB
                            \ initialise pab
          RLTY FXD INTRNL
          RECLEN 2 REC-LEN
          >FRAME @ DUP \ source string adr
C@ DUP ROT 1+ \ get length of string
PAB-ADDR @ 10 + \ destination string adr
  6
  ġ
  9
          ROT VHEW
                            \ move string to PAB
          PAB-ADDR 8 9 + \ point to file name length location
 10
          YSOM
                            \ put length into PAS
 12
 13 : AT GOTOXY ;
 15
SCR #246
  O \ DATABASE
                          19Feb87
  1 BASE-)R HEX
  2 : CHEK IF DROP NO ERR \ = no error & INTERN
3 ELSE 0 14 AT ." File error code " . FIL ERR
                                     \ = no error & INTERNAL file type
          ENDIF :
  6 : CHEKSTATUS
                    \ ( -- Error ) check status of file operation
          STAT DUP 10 =
                                     \ 10h = 16 dec = > bit 3 set
          OVER 11 = OR CHEK :
  9
    : CHEKFILE
          837C C2 0=
 10
                                     \ if no error at status byte
          IF STAT 11 = IF NEWFILE
 11
                                     \ 11h = 17 dec = > bits 3, 7 set
 17
                                     \ then it is a newfile
            ELSE CHEKSTATUS
 13
            ENDIF FIL ERR ENDIF (
 15 DECIMAL R-)BASE -->
SCR #217
  0 \ DATABASE
                          18Feb87
    : SELECTDAT
       BUFRLEN RECLEN! DATPAB
       FNAME SENAME !
                                     \ set fname pointer to data file
       SETUP_PAB
  ğ
   : SELECTIND
       INDRECL RECLEN! INDPAB
INDNAME >FNAME!
 10
                                    \ set fname pointer to index file
       SETUP_PAB
 11
 12
       ,
 13
14 -->
 15
```

```
SCR #248
 0 \ DATABASE
  2 : $ARRAY
                \ ( frows depth -- ) compile time
                \ (index# — index rec addr ) run time
       (BUILDS DUP C, x DUP HERE SMAP BLANKS ALLOT
DOES) DUP C2 ROT & + 1+ }
 7 \ define a string array 'IMDEX' for indexes
 9 INDHAX INDRECL $ARRAY INDEX
10
 1! \ allow two bytes (1 word) for data file record no
12 \ at the end of 'key' in the index
13
14
15 -->
SCR $249
 O \ DATABASE
                      19Feb87
 1 : CHEKRANG
                   \ ( Rnum -- Error ) check Rnum is within limits
         DUP DUP
         OK SWAP
                   \ is it less than zero
         INDMAX ) \ is it greater than max no of records
 5
         OR
 6
         ;
 8 : CLR BUFR
                   \ ( -- ) sets contents of buffer to spaces
         BUFR BUFRLEN BLANKS;
11 : CLR_INDEX
                   \ ( -- ) fills index array with blanks
         17
13
15 : CLR PROMPT 0 20 AT 40 SPACES : -->
SCR #250
 O \ DATABASE
                      19Feb87
              \ ( Rnum -- Error ) writes record Rnum to file
            \ assumes record is in BUFR and file is open
       CHEKRANG
                   \ check Rnum within limits
  5
       lF
         DROP
 6
         CUTRANG
       ELSE
  8
         REC-NO
 9
                    \ put no of record into PAB
 10
          RECLEN 3
                    \ get no of bytes to write to file
 11
                    \ write record to file
          CHEKSTATUS \ get error status
 12
 13
         DROP NO ERR
       ENDIF ;
15 -->
SCR #251
 O \ DATABASE
                      19Feb87
               \ ( Rnum -- Error ) reads record Rnum from file
            \ will leave record in BUFR and assumes file is open
       CHEKRANG
                    \ check Rnum within limits
       ĪF
         Dogo
  6
          CUTRANG
          ŘĒC-XO
                    \ put no of record into PAB
 10
                    \ read record from file
         RD
          DROP 1
 11
                    I no of chars read from file
          CHEKSTATUS \ get error status
 12
         DRCP NO ERR
 13
 14
        ENDIF ; -- )
 15
```

```
SCR #252
  O \ DATABASE -
                          18Feb87
  2: INITDAT
        1 12 AT
." Creating new data file." CR CR
CLR BUFR 0 BUFR! \ first :
                                      \ first record of file contains
        O FORITE DROP
O MAXREC ! ;
                                      \ no of records in use = MAXREC.
                                     \ ( -- n ) n=no of chars entered
  7 : GET KEY
         0 20 AT . " Enter key Field"
 10
                                      \ gets 'key' from operater
\ fill buffer with blanks
         0 4 AT
 11
         CLR_BUFR
BUFR
 12
 13
         INDLEN GETS
 15
SCR #253
  O \ DATABASE
                          28Fe587
  2 : INITSTUFF
                            \ ( -- ) reads rec 0 of data file to get
         O FREAD DROP
                            \ MAXREC. Then read maxrec records from
         BUFR 2 MAXREC! \ index file into index array. Closes index SELECTIND OPN \ file and leaves data file selected
  5
         CHEKSTATUS DROP
  6
         MATREC 9 DUP 0 >
         IF 1+ 1 DO
  8
  Q
               I REC-NO RO DROP
               INDBUF I INDEX INDRECL CHOVE
 10
            LOOP
 11
         ENDIF
 12
         CLSE SELECTDAT
 13
 [ ]
         ŧ
 15 -->
SCR #254
  O \ DATABASE
                          18Feb97
  2 : SELECTFILE
        CLS 1 10 AT GET_FNAME
                                      \ get file name
        FNAMEFLAS & IF
          CLR INDEX
SELECTDAT OPH
                                      \ select and open data file
                                      \ read record zero = maxrec
          CHEKFILE
                                      \ if creating new dat file then
  8
          HEWFILE =
  ě
                                      \ initialize it
           IF INITDAT
 10
          ELSE INITSTUFF
                                      \ get no of records in use=MAXREC
  11
           ENDIF
        ELSE CR . " No filename entered yet"
  17
  13
        ENDIF
 14
        ;
  15
SCR #255
   O \ DATABASE
                           19Feb87
   2 : FCLOSE
        SELECTDAT CLR BUFR
MAXREC & BUFR ! O FWRITE
                                      \ select data file PAB
   3
                                      \ save no of records in rec# 0
         DROP CLSE
                                       \ close data file
         SELECTIND OPN
                                      \ select index file PAB
         CHEKSTATUS DROP
                                      \ chekstatus
        MAXREC 9.1+ 1 DO
I DUP REC-NO
                                       \ copy index to index file -
   Ģ
                                       \ qet addr of index record I
  10
           INDEX
           INDBUF INDRECL CHOVE
                                       \ move index record to buffer
           INDRECL WRT
                                       \ write to file
  12
         LOGP
  13
  14
         CLSE
  15
         O FNAMEFLAG!; ->
```

```
SCR #255
  O 1 DATABASE
                                23Feb87
     : GETCHARS
                          \ ( -- chi ch2 )
         AD1 9 C2
AD2 9 C3
  6 ;
7 -->
   ġ
 10
 11
12
13
 15
SCR #257
   O \ DATABASE
                                23Feb87
           \ ( addr1 addr2 — flag ) returns tree if string at \ addr1 is alphabetically less than string at addr2 ADZ! ADI: 0 \ \ save string pointers
                                               \ save string pointers
\ scan all of key field
   5
            INDLEN
                         0 D0
                                               \ for case when strings are equal
   6
7
              DROP 0
              GETCHARS (
1F DROP 1 LEAVE
   89
                                               \ true
              ELSE GETCHARS >
                 IF LEAVE
ELSE 1 AD1 +!
  10
                                               \ false
                                               \ increment string pointers
  11
                        1 ADZ +!
  12
  13
                 ENDIF
  14
              ENDIF LOOP;
  15 -->
SCR #258
   0 \ DATABASE
                                 19Feb87
   1 : SURTINDEX
          MAXREC 2 1 DO
             I NINI!
             MAXREC 2 1+ 1 1+ DO
I INDEX
MINI 2 INDEX $4
IF I MINI ! ENDIF
   45557
                                              \ $< is string less than</p>
   9
             LOOP
                                INDBUF INDRECL CHOVE
             I INDEX
             MINI & INDEX I INDEX INDRECL CHOVE INDRUF MINI & INDEX INDRECL CHOVE
  10
  11
  12
          LOGP ;
  14 -->
  15
 SCR 1259
   O \ DATABASE
                                  211ar 97
    2 -->
    56789
   10
11
12
13
14
```

```
SCR 1260
  O \ DATABASE
                           28Feb87
                           \ ( -- (LO+HI)/2 )
       10 2 HI 2 + 2 / ;
                   \ { adl ad2 -- flag } true if string adl=ad2
        INDLEH 1- 0 DO
         DUP C2 ROT DHP C2 ROT = IF 1+ SWAP 1+
         ELSE DROP O LEAVE ENDIF
        LOOP
 10
        SWAP DROP ;
 12
                           \ ( -- flag ) true if LO > HI
 13 : L>H
        in a Hi a > ;
SCR #261
                           28Feb97
  0 \ DATABASE
   2: ?SRCH_FIN \ ( ad n1 -- flag ) true if key at addr is OVER OVER INDEX \ the same as element n1 of index CR if
           KEY= DUP HIMI !
                                 1 LO > H1
           L)H DR ;
   6 -->
   789
  11
12
13
14
 SCR #262
   O \ DATABASE
                            28Feb 27
                     \ { ADOR -- N } N=-1 => failed else N=key location
                     \ looks for Key in KList; returns location in KList \ if found otherwise returns -1
      : FINDKEY
         1 LO !
MAXREC 2 HI ! O
                               \ does it using a binary search
         BEGIN DROP
            MID OVER
            OVER INDEX $4
            IF DUP 1 - HI !
ELSE DUP 1 + LO ! ENDIF
    3
    7
            ?SRCH_FIN
   10
         UNTIL
   11
          SWAP DROP
          MINT 2 0= IF DROP -1 ENDIF
   13
   14
          ; -->
  SCR #263
    O \ DATABASE
                             Mar87
    2 0 VARIABLE MAXI
3 0 VARIABLE ITEM
                             \ ( n1 -- n2 ) n1=index rec no
    5 : GET_DATRECNO
                                              n2=data file record no
           INDEX
                             \ point to data file record no
           INDLEN +
    8
    10
   11
   12
13
14
   15
```



The Editor, Hunter Valley 99'ers News, 9 Thirlmere Pde., Tarro 2322

Dear Sir,

I note with interest the Press Release in your Feb.1987 issue concerning the development of an antivenene for the FUNNELWEB (Atrax Robustus) genus T.& W. McG.

It will not come as a surprise to your correspondent that work on such a vital development has been in progress in many research establishments all over the World and I congratulate R.Kleinschafer for being the first to publish. (After all, the first rule in Academia is - PUBLISH OR PERISH -)

However, I feel that I must now submit the results of a preliminary clinical trial carried out at THE INSTITUTE OF MINIMAL RESEARCH into the possibility of utilising other brands of antivenene.

It is not surprising that the Director of Research at the IMR should go down the same track as your correspondent - although I would have thought that he might have first investigated the potential of COOL GOLD, as this is a locally produced brand most likely to be successful against a local organism.

Please note that the author of this letter is NOT responsible for any damage resulting from misuse of the information contained in the Report.

Translated from the Gaelic by Dhonal Dhu (Black Donald).

Yours faithfully.

John L.Grant.

encl. Preliminary Report: CLINICAL TRIAL ON THE POTENTIAL OF SEVERAL BRANDS OF ANTIVENENE CLAIMED TO BE EFFECTIVE AGAINST FUNNELWEB (Atrax Robustus) genus T. & W. McG.

Discussion: Since the organism was deliberately released into the World Community from a Laboratory in KOTARA some time ago, it has spread with a vigor and virulence not unlike the Black Death of the Middle Ages or, more recently, the Influenza epidemic following the First World War. It is now in fact a pandemic.

The symptoms are too well known to require comment here and this Clinical Trial was concerned solely with research into the possibility of the organism being sensitive to a particular class of antivenene which had shown some promise in controlling the debilitating symptoms of the disease.

Protocol: Such was the danger inherent in carrying out this Trial that only ONE patient was willing to volunteer for the series of treatments. The patient was Male, over 40yrs, of Celtic stock - there may be racial characteristics involved - and blood tests indicated relevant antibodies present.

Method: At first sign of an attack the patient was given 375ml. of antivenene at hourly intervals until clinical improvement was evident. The subject's reactions were noted and - of particular importance - the time between attacks.

Material: The following brands of antivenene were used:

COLD GOLD XXXX FOSTVIC MELBIT WESTSWAN

Results:

COLD GOLD: This was not tolerated at all well. The patient complained of a 'nasty' taste in his mouth, feelings of revulsion and general malaise. Such was his reaction to this brand that medication had to be discontinued and the patient revived with a substantial dose of 'uisghe bhath' (Untranslatable - Editor).

XXXX: This was tolerated well and the results looked promising after an unimpressive start. 3 x 375ml, were needed in order to discern any favourable improvement in the patients condition. After that, however, the response was rapid, with the patient becoming totally symptom free following the administration of the eighth dose.

FOSTVIC: This was extremely well tolerated. There were no undesirable side effects, a feeling of bonhomie was rapidly induced and the patient became unconscious after the administration of the tenth dose.

MELBIT: This medication had to be discontinued after the second dose as there was a strike at the suppliers and further batches could not be obtained.

WESTSWAN: Very well tolerated with fairly rapid descent into incoherence. However it was noted that this may have been due to



the fact that this brand is available in glass vials of only 345ml. and the patient's reaction in the early stages may have been due to sub-therapeutic doses. Further trials on this aspect are recommended.

CONCLUSION:

- (a) At this time it would seem that the best results can be obtained with FOSTVIC followed closely by XXXX and WESTSWAN.
- (b) With such a small sample it is not possible to draw any conclusions as to how long subsequent attacks may be deferred. Individual tolerances obviously will play a significant role here.
- (c) With such a wide range of results being obtained from locally manufactured antivenenes it would be imprudent to try to forecast possible results with foreign manufactured products over whose quality control we have no input.

ADDENDUM: Research has disclosed the fact that there are now several different strains of the organism in the Community and this Trial was carried out on the strain known as Vers.3.3

It is recommended that further trials be initiated on patients who are carrying the antibodies to different strains as soon as possible as our preliminary research indicates that clinical results might be very different on strains other than Vers.3.3

Sigmund Bok (Prof.) Director of Research. Institute of Minimal Research.

ADVENTURERS CORNER

WITH RODNEY GAINSFORD

It is a short keep away from him one, but, as always, full of lots of * If the hurricane is too much for exciting Adventure hints, clues and then say YOHO other vital information. I am now *Yes, you do need the bandana other vital information. I am now getting into FORTH, so I do not have quite as much time to devote to my ADVENTURING - but I still find the * Dig in the sand favourite computer hobby time to have a the odd little fiddle.

I obtained a copy of one of Infocom's less popular Adventures known as 'Wishbringer', and I have been working on it recently, and making reasonable progress. As you * Pick lock in workroom. can see opposite, this months map is Drop wooden stake. of Savage Island, so to complement Leave clip outside.

the second step Island and 'The Count'. As yet I completing have been unsuccessful in my efforts to obtain Hitch-Hikers 2 but I'm still trying.

Till next month...Adventure on.

DŔ

SAVAGE ISLAND

Hi, welcome to this month's * If the bear makes you nervous then

the note

* The water in the tidepool is salt water

* At Argh, swim up

* Move the stalactite with the coconut

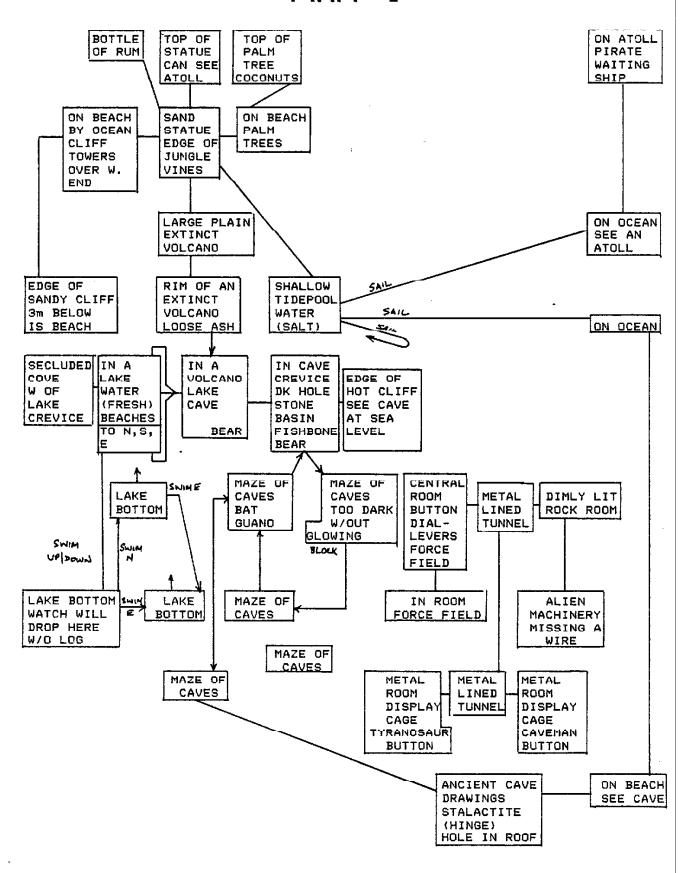
* Fix the alien machinery with the wire

THE COUNT

Get vial. Lock closet. Take mallet that we have some hints for Savage along. (items left in closet won't be stolen

SAVAGE ISLAND

PART 1



to t I rts I'm

on.

hen for

and

alt

the the

et. let n't

al.

HUNTER VALLEY 99'ERS USERS GROUP LIBRARY SOFTWARE GUIDE

The conversation went something like thist

Al to Joe, "Nobody came forward to do the software review for April!" Joe to Al, "Oh well after the crook review the Mob from Maitland did last month I'm not surprised! HEH HEH."

Al to Joe, "Yeah that maybe right but I still have to get the review done for the Newsletter."

Joe to Al in his most magnanimous tone, " She's right Al, give me the disc and I'll do it."

Ha! Caught again, the disc was in my hand before I could blink or BLWP to a change my mind subroutine.

Reviewing software is not an easy offering, should they all be as good type in and then dumped to the as say FUNNELWEB. If it isn't, then printer. criticise it out of hand? Not me, I load data are provided. I would not am a coward of high proportion. On have much use for such a programme the other hand I am not going to be in it's current form. untruthful about any of it either. So that left me with only one and probably the most correct approach. Write what I think about each piece of software.

APRIL SOFTWARE REVIEW.

The first file on the disc is of course the auto boot LOAD programme.

LOAD.

This programme developed into I would even suggest that squares. when Gary Jones gets his grubby F D S A Do the same but move net up hands onto it it well could. A the screen one square. series of subroutines are load into G Moves memory expansion and are vertically. available by using GALL LINK(etc..) SPACE BAR Moves the either in immediate mode or via a square vertically. programme statement. The subroutines are;

SETUP - Installs default printer and one!! disc drive number parameters.

CAT - Catalogues default disc tol screen.

CATP -Catalogues default disc to printer.

CATSP -Cataloques new disc to printer.

More work is needed to complete this utility.

BEETLERUN.

Here is a game in which one has to guide a BEETLE? through a maze and devour certain abjects before advancing to the next screen. Five screens are to be negotiated before completing the game. I think that this is correct, I could only manage to get onto screen two before being obliterated myself.

ESS

Wha Gra It :

dev

Thi

SOM

A s

pro

imp

pro

Му

FLA

Two

Joe

Ιd

on i

mos Mem

req

Cit

dis

ans

MAL

Her

mid

tex At

avai

FUÑ FUN

= 19

Tex

for

can

pro

ed

a bi

have

Per

madi

PIL

Ia

pro

T. I

15: lan

dog smai

I h

any

Doc

it i

fin

lik

PIL

run

The BEETLE is controlled using the E,S,D & X keys. The screen graphics are reasonable. On entertainment it rates on a par with CAR WARS.

BILL/STATE.

This programme in BASIC dumps Statement αf Account How do you approach each printer. Details of the account are No facility to save or

BONKERS.

The introduction to this programmes proclaims that you could be sent BONKERS playing this game. They ain't wrong IRVING! The aim of the game is to catch randomly falling objects in (wait for it) a BONKER net!! Controls are:

B N M , Move the net 1,2,3,4 squares to the right.

H J K L Do the same but move the net could have been one square up screen.

something quite V C X Z Move the net 1,2,3,4 left

the net one souare net down one

I think CAR WARS could well pip this

ELIZA.

As an attempt at interactive computing, I found the whole exercise. useless. A series profound answers are used to answer comments and replies you enter when prompted. In the words of ELIZA "I can help you". mmmmmm.

ESSCHER.

**)

to

and

bre

ive

bre

hat

ge

lng

he ics

it

а

bur

re he

or ot

hne

bes

ent

ley

he

ng

ER

85

let

1,4

up

re

ne

is

lve

le

of

ker.

en

" I

What is ESSCHER?. Thanks to Geoff Gray I am now more well informed. It is the name of the gentleman who developed this type of 3-d drawing. This programme in Ex Basic displays some VERY impressive 3-D graphics. A series of 5 separate screens are produced. Every one is a very 3-D image. impressive This programme is well worth a look at. My rating is GOOD.

FLAGS1.

Two good programmes in a row, has Joe gone BONKERS? No! This programme I did like. The programme displays on screen a National flag, The User, most likely one of the younger Members of your family then is required to type in the Capital City of the country whose flag is displayed. No scoring, no nasty bad answer warnings. My rating is GOOD.

MAIL-RITER.

Here is a programme for the B.B.S. midnight set. Mail-riter prepares text for the TE 11 AUTO Logon file. A type of word processor. Functions available are:

FUNC 5 Return to main menu.

FUNC 6 Turns page.

= Terminate text input

Text can be saved in 28 or 40 column format. Not being a B.B.S. nut I cannot comment on the value of the programme to such people. The text editor does work nicely. I did find a bug in line 500, GOTO 14040 should have been GOTO 140.

Perhaps some more comment could be made by one of our B.B.S. users??

PILOT-PILOT/DOC-PILOT/INST.

I am reliable informed that this programme comes from Stephen Shaw of T.I. Times fame. The first PILOT is a simulation of the programming language PILOT. The second is the documentation and the third is a small programme to print the Docs. I have at no time in the past had any contact at all with PILOT. Documentation has been reprinted it's entirety following this review, firstly as information for those like myself who have never PILOT, and as reference for when you run the PILOT simulator on the disc.

SAVINGS.

This programme is written in Basic. The screen presentation leaves a bit to be desired. none the less the programme can save the intending saver or borrower some time over a calculator. After all isn't that what computers are supposed to do? Four options are available to analyse your savings or indebtedness:

- 1 COMPOUND INTEREST.
- 2 LEVEL PAYMENTS.
- 3 INCREASING PAYMENTS.
- 4 FUTURE VALUE, FIXED PAYMENTS.

That ends this review, time to RTWP back to the Assembly article.

JOE WRIGHT.

PILOT

The Language

Documentation:

This EXTENDED BASIC program simulates the earliest version of the conversational language known as PILOT.

It does NOT simulate the Pilot which - in the end ΤI did not manage to release. compared to the earliest Pilot Extended Basic compares to ANSI BASIC. The general form is however the same. This program will demonstrate the essential structure of PILOT.

When running this program, you must ENTER a line number and then (in a separate entry) ENTER the PILOT program line. Commands such as NEW are given as a program line entry, after a number entry.

Use NEW to start writing a new program.

COMMANDS:

Many commands may be used as program entries (when they will immediately be carried out) although some require that you first enter REQUEST mode by typing as a line entry REQ

LIST. This command will list the LIST will program lines chosen. produce a prompt for the lines required, which are entered as a single entry in the format of two numbers separated by a comma, eg 12,20 or for a single line 5,5

LIST is allowed as a program line entry, and will leave you in request To carry on entering, request mode. any line numbers EDIT with specified. See Edit.

EDIT will produce a prompt asking list. Here are which lines, and two numbers must be language elements, which will be input as a single entry, separated followed by a short example. by a comma.

NB: The lines you choose control T/ only the display. You may then following the / a new line number outside the range requested.

pause while the program switches to the correct format.

If you wish to delete a line, enter the line number, then for the program line enter a null (eg just press ENTER).

To enter EDIT mode when you are in the middle of entering a program, type in as a program line REQ then select EDIT.

first time after entry or after following the / editing it. of the information. Remember to If the list contains NO then any always use RUN after typing in, loading program it tape/disk, or editing it.

line: the program immediately RUNs. Unless there is a previous E/ command in the program, the effect will be the same as entering RUN at ME/ program: it keeps RUNning!

Once you have placed QEP. the program in RUN format, subsequent runs may be commanded with REP, which acts more quickly. RUN may be used as often as you wish without ill effect, but takes longer.

DO NOT use REP after editing: not until you have first used RUN!

after REQUEST? NEW: Use Starts a new program.

STOP: Returns you to TI EXTENDED BASIC.

SAVE: Will save your program.

LANGUAGE ELEMENTS:

Entering a PILOT program, you first ENTER a line number, then as a separate entry a PILOT program line.

Each PILOT program line comprises a command letter(or letters) followed EDIT. As with list, the command by a slash (/) then an optional the available

Will (text) print anything reenter any existing line, or enter NB: The text may not contain a comma (,) Leading spaces are not suppressed.

After typing EDIT there is a short A/ (ask) Requests an input. Provide the prompt with A/ A string variable must follow the / String variables are denoted by the character @ Samples of string variables: @A @NAME @IN and so on To print a variable string subsequently, use the T/ command, thus: T/HELLO @NAME

M/ (match) One of three comparators. the last input variable Checks RUN: Used to RUN a program for the (input with A/) against the list RUN changes the format The command is for a PARTIAL match. a input containing N O together will from return TRUE eg METRONOME is true for NO.

EVERY word in the list must be RUN may be entered as a program terminated with a semi colon (;) even if there is only one word in the list.

(match exactly) Similar to M/ end of an Extended Basic|but requires the input to match one element of the list EXACTLY. Only

NO e l en the cole Samp reti WAS

J/ Rout is | J/*8

R/ subi foli subi

NB: unid The ind line

*SU E/ pro Can prod

con

CON A11 Whe beel I+ COM whi

Sim COM wh f Uns boti

See

NUM Two pro ant and.

To var C/ val CIT pre

> CK/ var CK/ fiv CK/

tha

ht

be

ùŧ

bt

ΕD

st

a

а

ød

a !

le

be

ng

ma

de

he

@A

le

id,

5.

le

st.

ny

11

or

be

11)

in

M/

bne

NO will match with NO. Every element in the list must terminate with a semi colon (;) Sample use: ME/YES;OK;O.K.;Y; will return TRUE if any of these items was last input.

J/ (jump) Jump to named routine.
Routine name must follow the / and
is preceded by an asterisk (*) as:
J/*SUBROUTINE

R/ (return) Used to terminate a subroutine, and pass control to line following J/ which entered the subroutine.

NB: SUBROUTINES: Subroutines are unique in not requiring a / command. The start of a subroutine is indicated by entering as the program line JUST the subroutine name as in *SUBROUTINE.

E/ (exit) Stops program. ALL programs should have this element! Can be used more than once in a program, and can be used conditionally.

CONDITIONAL COMMANDS:

All / commands may be conditional. When M/ ME/ or CK/ (see below) have been used a TRUE/FALSE flag is set. If a match has been made, all / commands suffixed Y are acted on while those suffixed N are ignored. Similarly if no match is made, all commands suffixed N are followed while those suffixed Y are ignored. Unsuffixed commands are followed in both cases.

See sample program at end.

NUMERIC VARIABLES:

Two numeric variables have been provided, I and J. These MUST be set to zero before use using: C/ZJ and C/ZI respectively.

To place a value into a numeric variable use the following form:

C/ (count) C/J+6 sets J to previous value plus six

C/I-2 sets I to two less than its

C/I-2 sets I to two less than its previous value.

CK/ (check) Matches value of numeric variable:
CK/I=5 sets TRUE if I does equal five
CK/J>8 sets TRUE if J is greater than 8

To print a numeric variable use T/with a variable preceded by a hash (#: US term pound) as: T/@NAME HAS A SCORE OF #I POINTS

SUBROUTINES ARE BEST AT THE END OF YOUR PROGRAM...labels are searched for from the end.

At program start up, you are asked PROGRAM NAME? To write a new program just press ENTER. To load from tape just type OLD.

Remember to enter program lines in two entries: first the line number, then the line. Line numbers must fall between 1 and 255 but need not be entered sequentially.

32k RAM will increase the capacity for a greater number of PILOT lines.

TYPICAL PILOT PROGRAM:

1
T/HELLO WHAT IS YOUR NAME?
2
A/@NAME
3
T/HI @NAME
4
T/LIKE TO DO THAT AGAIN?
5
ME/Y;YES;YEP;OK;O.K.;YEAH;

RY/ [no outstanding JUMP so return]
[will RUN program again from line 1]

T/YOU ANSWERED IN THE NEGATIVE!

T/I LIKE @NAME!

T/LETS USE A SUBROUTINE...

J/*GOODBYE

11

T/THATS ALL FOLKS

12 E/ 13

*GOODBYE

14

T/IT HAS BEEN GOOD TALKING WITH YOU 15

T/PLEASE PLAY AGAIN

16

R/

1フ

E/

18

RUN

THE INFORMATION PAGE

IN YOUR NEWSLETTER THIS MONTH

Factorials Member Carmany Random Bytes В. Assembly Language - Branch/Branch & Link Wright The TI Goes Bush Double Density TI Disk Controller Lima U.G. Struggling FORTH - VDP RAM R. Mulvaney Electric Circuits Part 2 - a program to type in Р. FORTH Data Base - Part 1 of a program to type in K. Bruce More feedback on FUNNELWEB antivenene J. Grant Melb U.G. Adventurers Corner incl map of Savage Is. part 1 R. Gainsford

PLUS MUCH MUCH MORE!!!!!

COMING EVENTS

Next Committee Meeting: Tuesday 5th May Next General Meeting: Tuesday 12th May Annual General Meeting: Tuesday 9th June

AGENDA FOR MAY MEETING

ì

Customising FUNNELWEB
Disk of the month demo

CLASSES AVAILABLE FOR MEMBERS

BASIC group conducted by Paul Mulvaney at the Warners Bay High on Tuesday 21st & 28th April.

ASSEMBLY group conducted by Joe Wright will meet on 21st & 28th April. Check venue with Joe prior to each meeting.

FORTH group conducted by Richard Terry each Thursday night. Check with Richard each Thursday re venue.

ANNUAL SUBSCRIPTIONS

Subscriptions to the Group cover the period i July to 30 June following year. Membership enquiries are welcome; please address all enquiries to the Secretary.

The annual subscription is:
Australian Residents...\$20
Overseas Residents....\$40 (airmail)
\$30 (surface)

Back issues of our Newsletter are available for \$1 plus postage

Mr.R. Carmany #97 1504 Larson St. GREENSBORO NC.27407 U.S.A.