April 84

HUNTER VALLEY 99'ERS NEWS

TI 99/4A

HOME COMPUTER NEWSLETTER

NEWSLETTER No. 9

TEXAS INSTRUMENTS

Newcastle

& The Hunter Region

TI-99/4A

Home Computer

USERS' GROUP

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY NINE USER GROUP. Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result of applying such information.
TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.
HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS.

CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.
Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX 80 or DIS/VAR 80). A suitable Public Domain word processor program will be supplied if required by the club librarian Al Lawrence.
Please include along with your article sufficient information to enable the file to be read by the EDITOR eg. File Name etc.
The preferred format is 35 columns and page length 66 lines, right justified.
All articles printed in HV99 NEWS (unless notified otherwise) are considered to be PUBLIC DOMAIN. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.
Articles for publication can be submitted to.

THE EDITOR
HV99 NEWS
9 THIRLEMERE PDE..
TARRO 2322

General address for ALL other club related correspondence.

THE SECRETARY
HV99 USER GROUP
6 ARCOT CLOSE,
TARRO 2322

YOUR COMMITTEE 1986

PRESIDENT
Allen Wright
77 Andrew Rd.,
VALENTINE 2280
Ph. 468120

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

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

TREASURER
Brian Rutherford
9 Bombala St.,
REDHEAD 2301
Ph. 498184

SOFTWARE LIBRARIAN
Alan Lawrence
35 Bayview St.,
WARNERS BAY 2282
Ph. 466309

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

EDITOR
Brian Woods
9 Thirlmere Pde.,
TARRO 2322
Ph. 662307

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

Garry 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

PRESIDENTS MESSAGE

In the last Issue of this Newsletter I announced on behalf of the H.V. 99'ers that Life Membership had been bestowed on Tony McGovern.

To people outside the H.V.99'ers. Tony is best known for his Funlwriter programme. I have the distinct feeling that if a survey was taken now, Funlwriter would be by far the most used piece of software currently available for the T.I.99/4A.

T.I.99/4A Users in the Newcastle Area first met as a Group on 13th September 1983. Tony and his two children were amongst that original Group. After the Meeting, along with the other 14 people who attended I formed the opinion that amongst us we were fortunate enough to have a very knowledgeable person in the form of one Tony McGovern. Since that time Tony has been on an exponential learning curve with his T.I.99/4A. While the remainder of us are on straight lines at various angles none of which are very large.

From that point on the Group as a whole has been amazed by and also benefited from the fruits of Tonys encounter with the T.I. The first widely used programme from Tony was Co-Lister. I am fairly certain that there would not be a Newsletter Editor who, if he has the programme would not be using it. Then of course are all the uses it would be getting by hackers like myself.

Then came his excellent series of Articles on Extended Basic. What is not widely know is that while producing this series Tony was also giving parallel talks at Users Meetings in Newcastle. It was always an interesting exercise to get the reaction of people leaving these talks. Some of the more common comments were, "learnt more on 30 min. Than the last 6 months", "he sure knows that Computer", "excellent talk", and "he lost me after a couple of minutes".

Having dealt with Extended Basic in the fashion of a true Engineer. That is, take a problem, dissect, it and finally present it in such a way that the layman can then make use of it. Tony was looking for something to really get his teeth into, ENTER Funlwriter.

How fortunate for the H.V.99'ers and the T.I.99/4A User Community at large that this was the way Tony decided to go and not to another machine. The hours which Tony has put into this Programme can only be immagined at by less proficient programmes such as myself. Tony has produced a New version of Funlwriter almost on a weekly basis. Further credit to Tony has been his reaction to the bugs found by people other than himself. He is very reponsive to the comments of others about his programme and enjoys recieving feedback about it.

In regard to Tony and the H.V.9'ers. No Group could ask more from a Member. He is unselfish in his attitude towards the Group and supportive of it's Members. I cannot remember Tony having missed a General or a Committee Meeting. (maybe one General?). He always has something to contribute to the Club. Be it knowledge, some software, literature or some useful general comments, it is always given freely and without reservation.

The Members of the H.V.99'ers. in bestowing Life Membership on Tony have given to him the Highest Honour available to them to show their appreciation of his work within the Group. We sincerely look forward to the future contributions from Tony to the T.I.99/4A Community. We are also aware that Tony has been recieving the justly deserved recognition from other T.I.99/4A Users Groups. We at the H.V.99'ers thank you for this and look forwards to regular contact between our Groups. It is this exchange between Groups that can keep our machines alive for the maximum possible time.

Congratulations Tony on your Life Membership it is justly deserved. I thank you personally for your contribution to the Group and for the information and knowledge I have gained from your works with the T.I.99/4A.

CLUB HAPPENINGS

Your new Committee had its first meeting last Tuesday and judging by the lively discussion, are we in for an exciting year!!

Due to the resignation of Steve Taylor as Secretary because of his move to the Apple Isle (our loss is their gain), Albert Anderson was elected Secretary and Garry Jones was elected to the Committee. Congratulations to both.

Congratulations also to Tony McGovern on his election as our first Life Member for his contributions to the Club. Tony has written a series of XB Tutorials which have appeared in magazines around the world, and also, in partnership with his son, Will, written Funnelwriter, our very own TI-WRITER loader. This utility has proven a boon to those who have the full system and the need for a good word processor without having to pay the exhorbitant price asked for TI-WRITER by IMAGIC.

Tony has consistently practiced what should be the aim of all true user group members - sharing knowledge - and for this we sincerely thank him.

This year should prove to be quite an exciting one for all members. Garry continues his Extended Basic lessons and Allen (Wright) his Basic course. Special Interest Groups will also be arranged through our S.I.G.'s Co-Ordinator Tim Watkins, covering such things as Funnelwriter, Data Bases, etc. and Richard Terry continues on as Forth Co-ordinator. There is a whisper that we will see an 'Introduction to Assembly Language' course within the next few months.

It has been decided that at each General Meeting, a demonstration of the current software tape would be made so that people can see what they are getting and how to 'drive' them.

All members are encouraged to write for their magazine. We would rather fill our mag with original material, written by our own members, than have to resort to reprinting material from other sources. If you would like to see articles about specific subjects please see me and I will see what can be arranged.

If you intend writing for the magazine, see the formatting details on the inside front cover. The deadline for material is the Tuesday before the General Meeting. Contributions may also be left with Allen Wright or Brian Rutherford for passing on to me.

If you were not at the Annual Meeting and did not receive a copy of our "Guide to TI-WRITER", see Paul Mulvaney or get in touch with Secretary Albert to obtain one. This 41 page booklet is a compilation of various articles written about TI-WRITER, and is highly recommended for those learning to use TI-WRITER (or should I say, FUNNELWRITER).

Don't forget that the club exchanges and subscribes to various magazines and newsletters from around the world. Copies are available for loan at each meeting for a period of one month from Publications Librarian, Paul Mulvaney. (Would those people who have 'forgotten' to return previously borrowed material please do so soon so that others may read them).

Our Module Library is steadily growing with about 9 modules available for borrowing for one month at a time. There is no charge for this service, but please remember to return them when they are due. See Rodney Gainsford about these.

In conclusion, I would again encourage all members to make some contribution to the club to ensure that we all gain maximum benefit from belonging to the club.

Brian Woods

P.C. 86

LATE ON THURSDAY EVENING 13TH MARCH 1986 , I STARTED TO COME DOWN WITH SOME AILMENT WHICH COULD ONLY BE BEST DESCRIBED AS COPUTERITIS . LATER THAT NIGHT , WHILST HAVING A BIT OF A NATTER WITH PETE (SMITTY) SMITH , HE SEEMED TO THINK THAT HE WAS COMING DOWN WITH THE SAME COMPLAINT , AND THE ONLY CURE WAS FOR US TO TAKE A SICKIE AND GO TO SYNDEY TO SEE DOCTOR PC86. IF WE DIDN'T GO NOW IT WAS GOING TO BE ANOTHER TWO YEARS BEFORE WE COULD MAKE AN APPOINTMENT FOR PC88.

WE SPENT FIVE HOURS WALKING UP AND DOWN AISLES , THROUGH EXHIBITORS STANDS , LOOKING WITH AWE AT SOME OF THE EXHIBITS AND GETTING CAUGHT IN DISCUSSION WITH OVER EMPHATIC SALESPERSONS.

WE MADE OUR WAY TO THE TOP FLOOR DURING THE FIRST HOUR AND A HALF , HAVING A QUICK LOOK HERE AND THERE ON THE WAY , AND STOPPED FOR A CUPPA AT 11.30 AM . ONLY ONE ONLOOKER AT THE TISHUG STAND BUT PLENTY OF ONLOOKERS AT THE OTHER USER GROUP STANDS (IT MAY HAVE DIED BUT WON'T LIE DOWN). WE THOUGHT WE MIGHT DO THE RIGHT THING AND HAVE A BIT OF A TALK WITH THEM ??, SHOULD WE TELL THEM WE WERE HV99ERS ?? (DO YOU KNOW WHAT A GRAMKRACKER IS !! , THEY MUSN'T HAVE READ OUR HV99ER ARTICLE). NOT MUCH GOING ON HERE SO WE MADE OUR WAY , EXHIBIT BY EXHIBIT , BACK TO THE GROUND FLOOR.

WE WERN'T REALY FEELING VERY HUNGRY LOOKING AROUND THE APPLES AND APRICOTS (I WONDER HOW MANY LEMONS WERE THERE). MOST STANDS WERE WELL ATTENDED WITH SWARMS OF ONLOOKERS EVERYWHERE LIKE LITTLE MICROBEES AROUND HONEYPOTS.

THE TWO MOTOR VEHICLE MANUFACTURERS , COMMODORE AND IBMW WITH VARIOUS CLONES SPREAD AROUND THE STANDS WERE WELL REPRESENTED , WITH MOST SHOWING IBMW/COMPATIBLE SOFTWARE OFF TO THE FULLEST. I JUST WONDER HOW MANY FUTURE IBMW CLONES/COMPATIBLES WILL COME TO FLOOD THE MARKET , EACH ONE BETTER THAN THE LAST ?? , WHICH MUST BY NOW BE VERY COMPETITIVE , AND WHICH MACHINES WILL END UP THE WAY OF OUR OWN BELOVED TI99/4A.

ONE ITEM THAT TOOK MY INTEREST WAS A FISCHERTECHNIC PLOTTER/SCANNER KIT FROM MODERN TEACHING AIDS , AND THE VARIOUS ROBOTICS/KITS.

VARIOUS SOFTWARE HOUSES DEMONSTRATED THEIR WARES , ONE BEING "WINDOWS" . IT WORKED VERY NICELY , AND WITH SO MANY WINDOWS OPEN ON YOUR SCREEN AT ONCE IT LOOKED LIKE A PENNSYLVANIAN AMISH QUILT MAKING SESSION .

I HAVE NEVER SEEN SO MANY BOOKS AND ACCESSORIES IN ONE LOCATION BEFORE TO CATER FOR ALMOST EVERYBODY - SORRY - NOT MUCH AVAILABLE FOR THE TI.

ALL THAT I WAS INITIALLY GOING FOR WAS TO LOOK AT A FEW "PRINTERS" !!!

PETE WAS BUSY TRYING TO BEAT THE PRICES DOWN ON A FEW COMPUTERS FOR THE SCHOOL (AS WELL AS HAVING A BIT OF FUN WITH THE VARIOUS MACHINES - IF YOU COULD GET ONTO ONE THAT IS). HE WAS LAST SEEN BUZZING BETWEEN THE BEE AND BARSON STANDS. GUESS WHOSE SCHOOL ENDED UP ON A BUS EXCURSION TO PC86 , AND GUESS WHICH SCHOOLS HEADMASTERS SON CAME ACROSS AND SAID HI TO SMITTY.

AND BY THE WAY , DID YOU KNOW THAT TISHUG TECHNICAL GROUP HAS DESIGNED MEMORY EXPANSION UNITS WHICH CAN NOW BE FITTED INSIDE THE EXISTING CONSOLE , AND ALL THIS TIME I THOUGHT IT WAS THE WORK OF WESTRALIAN INSTRUMENTS. (SEE P.45 PC86 SHOW CATALOGUE).

TOLD MY BETTER HALF THAT WE WOULD BE HOME ABOUT SIX , BUT IN STRIFE AGAIN .

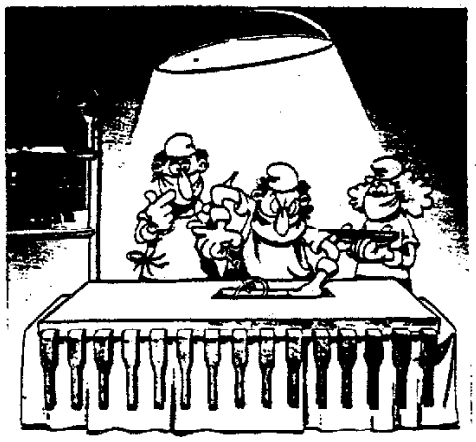
WELL , RUNNING OUT OF MEMOREX OF THE WHOLE SITUATION , AND BESIDES , I HAVE TO GET THIS LOT TO THE EDITOR BEFORE EASTER.

TERRY ROSS
ANOTHER HAPPY HV99ER (BUT NOT COMPLETELY CURED OF COMPUTERITIS).

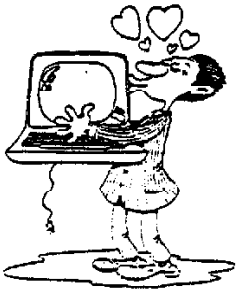
THE GARY JONES STORY.
(OR LOGO LAMENT)

Gary Jones recently imported LOGO II and filled with a sense of joy plugged the module into his recently expanded console (32K matchbox memory) switched on, pressed 2 for LOGO and -nothing-. The console had been working perfectly previously, when the LOGO module was removed it ran OK. Frantic phone calls all around the country to anyone who would listen to a grown man cry revealed that LOGO checks every memory location upon startup.

if there are any problems with memory LOGO will not operate. A program to check every memory location was devised and it revealed several faulty locations which were traced to one of the memory chips.



After investigation the problem was found to be a bad connection between the pins on the chip and the chip socket. Careful adjustment of the pins so they made good contact with socket and now Gary is a happy LOGO LOVER.



EXTENDING

EXTENDED BASIC

by TONY FALCO

Reprinted from the August '85 issue of MUNCH, the Mass Users of the Ninety Nine and Computer Hobbyists of Worchester, Massachusetts.

Last month I suggested that an appropriate activity for user groups like MUNCH might be to create and assemble a set of subprograms to add commands to the Extended Basic language. "Extending Extended Basic", if you will. In keeping with that suggestion, this month I have written a program that supplies several subprograms. These CALLs can be used in programs for young (and young at heart) people. They provide some easy ways to use some of the 99/4A's more spectacular sound, color and graphics effects.

The program consists of a series of subprograms which are in lines numbered 10000 and above. Lines 100 to 140 consist of a brief demonstration on how to CALL the subprograms. If you want to use the routines in your own programs, then simply delete lines 100 to 140 and save the rest to tape or disk. Before you type in a new program put the routines in memory by means of the OLD command. Then write your program CALLing the routines as needed. Just to be sure that none of your line numbers exceed 9999.

Here's a brief summary of how they work:

CALL BORDER(C1,C2,C3) - Creates a 3 color border. C1, C2 &C3 are constant or variables with values between 1 & 16 (for TI's color codes).

CALL FLASHBORDER(C1,C2,C3) - Makes an already created border flash the sequence of 3 colors C1,C2 &C3.

CALL SHOW(<String Variable or String Expression>) - Centres a string at the screen's dead centre.

CALL EXPLORE - noises and flashing screen.

CALL SIREN, CALL POLICE, CALL EMERGENCY - These are siren-like sound effects of different types.

CALL CRAWL(ROW,<STRING VARIABLE or STRING EXPRESSION>) - Causes a message to crawl across the screen along the specified row.

```
100 CALL CLEAR :: CALL SHOW("A COLD
RED BORDER") :: CALL BORDER(7,9,10)
:: CALL SHOW("LISTEN TO THE SIREN")
:: CALL SIREN
110 CALL SHOW("HERE ARE THE POLICE")
:: CALL POLICE
120 CALL SHOW("IT FLASHES TOO!") ::
CALL FLASHBORDER(7,16,6) :: CALL
EXPLODE :: CALL FLASHBORDER(11,15,3)
130 CALL SHOW("IT'S AN EMERGENCY")
:: CALL EMERGENCY
140 CALL CRAWL(24,"THIS IS A TEST OF
THE CRAWL ROUTINE. TO END IT AND
THE PROGRAM PRESS ANY KEY.") :: CALL
CLEAR :: END
```

```
9995 !----------------------------
10000 SUB BORDER(A,B,C) :: CALL
COLOR(13,A,B,14,C,C) :: CALL CHAR(
134,"0",135,RPT$("F",16), 136,"0")
10010 FOR T=1 TO 3 :: CALL HCHAR(T,
1,133+T,32) :: CALL HCHAR(25-T,1,
133+T, 32) :: NEXT T
10020 FOR T=1 TO 3 :: CALL VCHAR(T,
T,133+T,26-2*T) :: CALL VCHAR(T,33-
T,133+T, 26-2*T) :: NEXT T :: SUBEND
10030 SUB FLASHBORDER(A,B,C) :: FOR
I=1 TO 50 :: T=A :: A=B :: B=C :: C
=T::CALL COLOR(13,A,B,14,C,C) :: FOR
J=1 TO 30 :: NEXT J :: NEXT I
10035 SUBEND
10040 SUB EXPLODE :: CALL SHOW("KA
BOOM") :: FOR SC=2 TO 16 :: CALL
SCREEN(SC) :: CALL SOUND(-2000,INT(
RND* 4-8),0)
10050 CALL SOUND(-2000,INT(RND*4-8)
,0) :: NEXT SC :: SUBEND
10060 SUB SIREN :: FOR N=1 TO 4 ::
FOR F=700 TO 900 STEP 5 :: CALL
SOUND (-99,F,0) :: NEXT F
10070 FOR F=900 TO 700 STEP -8 ::
CALL SOUND (-99,F,0) :: NEXT F ::
NEXT N :: SUBEND
10080 SUB POLICE :: FOR N=1 TO 20 ::
FOR F=1200 TO 900 STEP -30 :: CALL
SOUND (-99,F,0) :: NEXT F :: NEXT N
:: SUBEND
10090 SUB EMERGENCY :: FOR T=1 TO 30
:: CALL SOUND(150,1320,0,1324,5) ::
CALL SOUND(150,880,0,888,5) :: NEXT
T :: SUBEND
10100 SUB SHOW(M$) :: DISPLAY AT(12
,4)SIZE(24):RPT$(" ",24) :: DISPLAY
AT(12;15-INT(LEN(M$)/2))
SIZE(LEN(M$)):M$ ::SUBEND
10110 SUB CRAWL(ROW,M$) ::
M$=RPT$(" ",28)&M$
10120 FOR T=1 TO LEN(M$) :: DISPLAY
AT(ROW,1): SEG$(M$,T,28)
10130 FOR D=1 TO 9 :: NEXT D :: CALL
KEY(0,K,S) :: IF S<>0 THEN DISPLAY
AT(ROW,1) :RPT$(" ",28) :: SUBEXIT
10140 NEXT T :: GOTO 10120 :: SUBEND
```

★ "I'M citing a home computer."

HI 99'ers,

"A Tribute"

****
Our One
****

We are heartened to know that one of our most valued members is still with us inspite of Someone Else's passing creating a vacancy in the club."Our One"has been with us since the club was formed,even before then he was one who did more than a normal persons work.
Whenever leadership was mentioned, this Super Programer was looked to for his Inspiration and Tutiorials. It was often said "Our One will be able to do that easy enough"
Whenever there was a job to do or help needed,one name was on everybodys list.."Our One's" sure to know how".
"Our One"has Extented the Basic to all our advantage! Sometimes even appearing Superhuman.Though a person can only do so much."Our One" were the truth be known,Will stick with the 99'and we Will all sing 4A's a jolly good person for a long long TI*ME(Free plug for U.K mag. TI*MES which we get in Exchange. Make sure you read IT)
Now"Our one"has shown a wonderfull example to follow,but who Will EXPLORE further ? Who can do the things that "Our One" has? Lets all hope Someone "Will" !!!

"Well Done".

I am sure we dont have to Spell everything out as there is no doubt we all know there are no REDBACKS on this FARM !!!

****

Library Access

Any clubs or indivuals interested in obtaining any PUBLIC DOMAIN software in volume disks have 2 choices

(a) Send blank initialised disks to us with return postage or send us disks with programs on it and we will send at our cost an equal number of disks filled with programs requested or volume disks.

(b) We can supply programs, or volume disks on our disks for the cost of disk and PP($4.00)

Happy Computing.
Al Lawrence.

PUBLIC DOMAIN DISK

#	TITLE	SOURCE	COMMENT
001	FLWTR V3.3	HV99'ers	McGovern
002	COLLIST	HV99'ers	McGovern
003	XB.Tutorls	HV99'ers	McGovern
004	JW-DISKCAT	HV99'ers	J.Wright
005	HV99-PDL01	HV99'ers	Various
006	TIWR+MPLAN	TEXAS-INST.	UGPD'83
007	SSPD1	S.Shaw(U.K)	UGPD'85
008	DM-1000	OTTAWA 99.UG	B.Caron
009	NO-FRILLS	PD85(TIshug)	B.Cabot
010	NEATLISTER	FREEWARE	Michael
011	FREEDUMP	FREEWARE	Michael
012	ALROUTINE1	NEW-HORIZONS	Clulow
013	ALROUTINE2	NEW-HORIZONS	Clulow
014	PROGRAMS	CHANNEL 99	Various
015	4th.		XB-load
016	4th.		MM-load
017	4th.Scode	TEXAS-INST	2 Disks
018	4th.	TEXAS-INST	UGPD'83
019	4th.Calndr	San Diego	Winkler
020	4th.Acount	San Diego	Winkler
021	4th.Demo	AMNION Free Access	
022	A*38	AMNION "	Games
023	A*39	AMNION "	Games
024	E* 8	AMNION "	Bus/Fin
025	F* 7	AMNION "	TecMath
026	G* 6	AMHION "	PerHome
027	UK-PD3/85	TI*MES (U.K)	Various
028	UK-PD4/85	TI*MES (U.K)	Various
029	XBDataBase	SLAVE(UTAH)	V.Parker
030	99-"c"Comp		C.Pulley
031	99V3-BBS.	PD-Use.ViaHoogendoorn	
032	TE3C-BBS.	T.I.&Mod.by J.Freeman	
033	MASS COPY	FREEWARE	Lawless
034	128-WRITER	"	Lawless
035	CAT&EnvDsn	" Trio	Software
036			
037			

****
* THE ORIGINAL BUG *
****

To new programmers, bugs often seem to take on animate characteristics, such as hiding from the programmer and deliberately undermining all his efforts to find them.

However the first bug (at least the one from which the term is derived) really was animate. In trying to eliminate an error from a program she was developing on the Harvard Mk. II in 1945, Captain Grace Hooper discovered that a large moth had got caught up in the electromechanical working of the computer and was causing the fault. As a result of that incident, the term 'de-bugging' was coined.

MINI WORD PROCESSOR PLUS MINI FORMATTER

BY BRIAN RUTHERFORD, HYSSERS

For the members of the club who have put in the extra 32k. of memory, but only have a cassette, here are the necessary changes to Mini wordprocessor and mini Formatter to join the two programmes together.

First load in Mini Wordprocessor and add line 155 CALL INIT::CALL LOAD(-31806,16), that will turn off the FUNCTION QUIT, which makes life easier. Next at line 160 change DIM L$(66) to DIM L$(88), to increase the number of text lines that can be entered. Also in line 160 change L$(0)="8000" to L$(0)="11500" and the same at line 280. Then in line 480 change IF L=66 to IF L=88. After that just add the lines of code printed below.

```
880 SUB PF(L$(),L):: CALL CU
TUP(L$(),L,K):: IF K=48 THEN
 900 ELSE IF K<>50 THEN
CLOSE #1 :: IF K=49 THEN SUB
EXIT
890 CLOSE #2 :: SUBEXIT
900 DISPLAY ERASE ALL :: FOR
 I=1 TO L :: PRINT STR$(I);"
:-":L$(I):: IF I/4=INT(I/4)T
HEN CALL KC
910 NEXT I :: CALL KC :: SUB
END
```

```
1000 SUB CUTUP(L$(),L,K):: C
ALL LO(K):: IF K=48 THEN SUB
EXIT ELSE P$="" :: I=0 :: IF
 K=50 THEN CALL PLINE(P$,I,1
)ELSE CALL PLINE(P$,I,0)
1010 PL=I :: P$="" :: FOR I=
1 TO L :: LL$=P$&L$(I):: P$=
"" :: IF LL$="" THEN CALL PL
INE(LL$,K,0):: GOTO 1110
1020 LL=LEN(LL$):: IF SEG$(L
L$,1,3)="/C/" OR SEG$(LL$,1,
3)="/c/" THEN LL$=RPT$(" ",I
NT(PL-LL)/2)&SEG$(LL$,4,LL):
: CALL PLINE(LL$,K,0):: GOTO
 1110
```

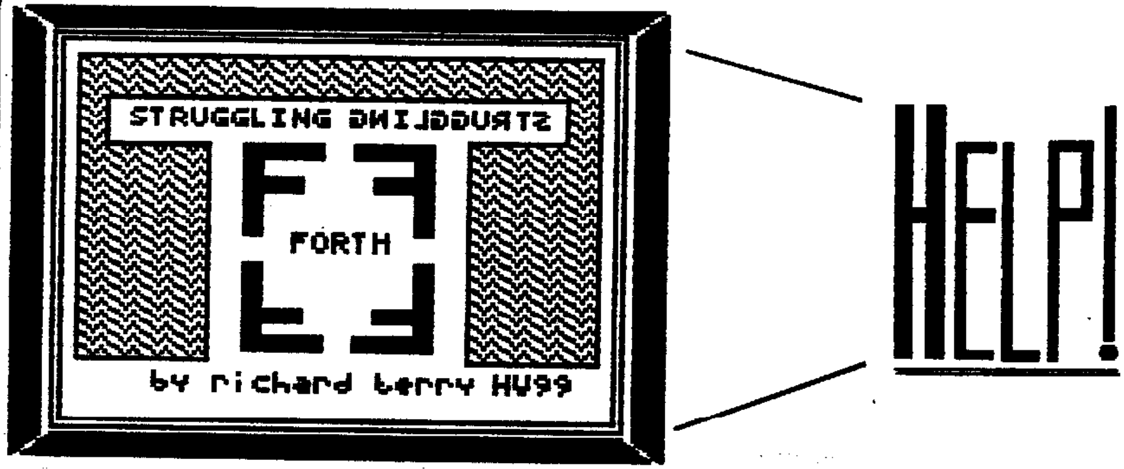
CONT

```
1030 IF SEG$(LL$,1,2)="//" T
HEN P=POS(LL$,"/",3):: LL$=R
PT$(" ",VAL(SEG$(LL$,3,P-3))
)&SEG$(LL$,P+1,LL):: LL=LEN(
LL$)
1040 IF LL<=PL THEN CALL PLI
NE(LL$,K,0):: GOTO 1110
1050 FOR J=PL+1 TO 1 STEP -1
 :: P$=SEG$(LL$,J,1):: IF P$
<>"" AND P$<>" " THEN 1100 E
LSE P$=SEG$(LL$,1,J-1):: LL$
=SEG$(LL$,J+1,LL):: LL=LEN(L
L$)
1060 IF LL$<>"" THEN CALL PL
INE(P$,K,1):: GOTO 1050
1070 IF I=L THEN CALL PLINE(
P$,K,0):: SUBEXIT
1080 IF SEG$(L$(I+1),1,3)<>"
/C/" AND SEG$(L$(I+1),1,3)<>
"/c/" AND SEG$(L$(I+1),1,2)<
>"//" AND SEG$(L$(I+1),1,4)<
>"    " AND L$(I+1)<>"" THEN
 1110
1090 CALL PLINE(P$,K,0):: P$
="" :: GOTO 1110
1100 NEXT J
1110 NEXT I :: SUBEND

1120 SUB LO(K):: DISPLAY AT(
5,1)ERASE ALL:"Output to":"_
_________" :: DISPLAY AT(8,3
):"1 Screen": :"  2 Printer"
: :"  3 Disk": :"  4 Both"
1130 CALL CH(4,K):: K=K-1 ::
 IF K=48 THEN SUBEXIT ELSE I
F K=50 THEN 1150
1140 DISPLAY AT(10,13)BEEP:"
PIO" :: ACCEPT AT(10,13)SIZE
(-13):P$ :: IF P$="" THEN 11
40
1145 CALL SCREEN(14):: DISPL
AY AT(22,1)BEEP:"Is your pri
nter turned on?":"" :: CALL
KC :: CALL SCREEN(5):: OPEN
#1:P$ :: IF K=49 THEN SUBEXI
T
1150 DISPLAY AT(12,13)BEEP:"
DSK1.FILE" :: ACCEPT AT(12,1
3)SIZE(-15):P$ :: IF P$="" T
HEN 1150 ELSE OPEN #2:P$,OUT
PUT,DISPLAY ,VARIABLE 80
1160 SUBEND

1170 SUB PLINE(P$,K,S):: IF
K THEN 1250
1180 DISPLAY AT(4,4)ERASE AL
L BEEP:"Line length 10-80":
:"   Left margin"
1190 ACCEPT AT(4,24)VALIDATE
(DIGIT):P$ :: IF P$="" THEN
1190 ELSE LL=VAL(P$):: IF LL
<10 OR LL>80 THEN 1190 ELSE
IF S THEN 1240
1200 ACCEPT AT(6,24)VALIDATE
(DIGIT):P$ :: IF P$="" THEN
1200 ELSE LM=VAL(P$)
1210 IF LL+LM>80 THEN DISPLA
Y AT(11,1)ERASE ALL BEEP:"LI
NE LENGTH PLUS LEFT MARGIN I
S LONGER THAN EIGHTY..." ::
CALL KC :: GOTO 1180
1220 DISPLAY AT(8,8):"Paper
size" :: DISPLAY AT(10,4):"1
 A4": :"   2 Quarto" ::
CALL CH(2,X):: IF X=49 THEN
X=11 ELSE X=9
1230 DISPLAY AT(23,2):"Right
 justified Y/N" :: CALL KEY(
3,J,S):: IF J<>78 AND J<>89
THEN 1230
1240 DISPLAY AT(12,10)ERASE
ALL:"Working on it" :: L=0 :
: K=LL :: SUBEXIT
1250 IF K<>49 THEN PRINT #2:
P$ :: IF K=50 THEN SUBEXIT
1260 IF J=89 AND S=1 THEN CA
LL RJ(P$,LL)
1270 IF L=0 THEN PRINT #1:""
 :: PRINT #1:""
1280 PRINT #1:TAB(LM);P$ ::
L=L+1 :: IF L<56 THEN SUBEXI
T
1290 FOR I=1 TO X :: PRINT #
1:"" :: NEXT I :: L=0 :: SUB
END

1300 SUB RJ(P$,LL):: P=0 ::
IF LEN(P$)=LL THEN SUBEXIT
1310 A=2 :: FOR J=1 TO LL-LE
N(P$)
1320 P=POS(P$," ",P+A):: IF
P=0 THEN A=A+1 :: GOTO 1320
1330 P$=SEG$(P$,1,P)&CHR$(32
)&SEG$(P$,P+1,LEN(P$)):: NEX
T J :: SUBEND
```



3 hours to hand-in deadline and only just starting. I hope my afternoons work is not too busy or I'm in trouble. This week I promised you some work with array's, vdp reading/writing. As per usual I'll avoid the issue by starting with something else.

BEGIN/UNTIL LOOPS.

These are incredibly useful and adaptable once mastered, but annoying until the following basic concept is grasped:

ONE CAN ONLY LEAVE THE LOOP ONCE IT IS ENTERED BY HAVING THE LAST VALUE ON THE STACK BEFORE UNTIL AS A NON-ZERO OR TRUE FLAG. AS A COROLLARY IF WE WANT TO STAY IN THE LOOP UNTIL A CERTAIN CONDITION IS MET WE MUST LEAVE A FALSE FLAG OR A ZERO (0).

Why are they useful?. Well they are elegant. For instance they allow us to pause the flow of the program, scanning the keyboard looking for a particular keypress before moving on.

```
SCR #70

: CONTINUE                  ( Expects nothing                )
        BEGIN               ( enter loop                     )
             ?KEY           ( leave ascii of key or 0 if none )
             32 =           ( was space bar pressed leave 1  )
                            ( if was or 0 if not             )
        UNTIL               ( leave loop if non-zero         )
```

Using ?KEY command is far more elegant than the KEY as an untidy cursor appears on the screen. Compare the examples on the next screen.

```
SCR #71
( EXAMPLES -detect key press 11Mar86)
: ECHOICE BEGIN ?KEY DUP DUP          ( check keypress, 3 copies  )
            65 < SWAP 69 > MAX ( (65 =A,)69=E,leave flag  )
               IF                ( if it is in out of rang   )
                 DROP 0          ( drop ascii code,false flag)
               ELSE 1            ( else put true flag to exit)
               THEN              ( now have true/false flag  )
        UNTIL                    ( leave if flag=1 with ascii)
CASE 65 OF ." A PRESSED"  ENDOF
     66 OF ." B PRESSED"  ENDOF
     67 OF ." C PRESSED"  ENDOF
     68 OF ." D PRESSED"  ENDOF
     69 OF ." E PRESSED"  ENDOF ENDCASE ;
```

Here is an example you could use in conjuction with a menu stating things like:

<A>... ADD <C>...CHANGE etc.

Instead of the message one could have:

CASE 65 OF PROGRAM1 ENDOF

etc

```
SCR #72
( BIGREQUES - display records   24Mar86)    ( CHECKED CLEAN)

: ANOTHER                        ( ? display another record      )
        BEGIN ?KEY               ( check for any key press 0 =none)
              DUP 2DUP           ( make 4 copies                 )
              10 =      SWAP     ( was it function X -next rec-  )
              11 = MAX SWAP      ( was it function E -last rec-  )
              14 = MAX           ( was it function 5 -exit key-  )
              0= IF              ( if result flag=0 was neither  )
              DROP 0             ( drop key code, leave false flag)
              ELSE 1             ( if X/E/5 pressed leave true flg)
              THEN
        UNTIL ;                  ( test flag leave if flag=true=1 )
```

I will analyse this one in more detail to show you exactly how it works.

?KEY

Scans the whole keyboard checking for a key press. If none were pressed it leaves a zero (0) on the stack.

DUP 2DUP

Four copies of the ASCII key code are made, 3 for future testing and the last will either be dropped if BEGIN will finds a 0 flag or kept and passed on via the stack to the next word in the program.

(An aside here. If ever you want to find out the ASCII value of a key just type at the terminal KEY and press enter then press the desired key and then . (print) to list it to the terminal)

10 =
----
was the key pressed function X, leave a flag. Similarly compare to the key values of function E, and function 5 which I use to exit menu's.

MAX

This when used leaves the larger of the two values on the stack, using it as we accumulate flags cleans up the stack as we go. Here it ensures that if only 1 of the three options are true a true flag will be kept.(1).

0=
--
The resultant flag is compared to zero. In effect it reverses the above flag for use by IF. Here it dosn't matter as one could have used 1 = and changed the next two lines around. At other times, especially when using MYSELF within definitions it is useful to change the direction of the flag provided you the programmer keeps track of which one you want it to be.

IF tests the flag, if the key pressed was not one of the three we wanted it drops the excess value to keep our stack clean and leaves a 0 or false flag for use by BEGIN which will loop again and again until we press one of the three keys. If the key we pressed was a valid one we deposit a true flag or 1 on the stack to enable us to exit the loop. In other words the UNTIL is like a sentry at a gate. If you give him nothing - 0 - he sends you back to try again. If you give him somthing - 1 or greater - he accepts this as being satisfactory and lets you out of the loop and back into the rest of the program.

KEEP YOUR STACK CLEAN.

Speaking of stacks always know what should be on there and what shouldn't. As each FORTH word can be run as a mini-program unto itself one can debug the word before releasing it into the general program . This article is beginning to have penal overtones!
Run each word as you define it by, after compiling it, providing on the stack the numbers/addresses it needs to work or by sustituting messages after IF or ELSE statements to let you know where it's at. First clear the stack by typing XX and pressing enter. Forth will reply by typing XX?. If you subsequently examine the stack using . then you will find its base layer ie 0 ,digit,11776. Run your word and see what garbage is left on the stack, then go back to your definition to see what is leaving it, clean it up and re-test.

With one exception which I will show below I have resisted using ROLL or PICK in my definitions and try and keep my stack never deeper than I can access with standard stack operators. I feel it use of PICK etc may encourage sloppy thinking, but am open to suggestions/arguments to the contrary.

GRAPHICS.

One thing you won't get out of me is any detailed descriptions of using graphics/sprites as they do not interest me. My interest lie more in the realm of business programming as you will see in months to come. I will touch on graphics only as necessary to visually enhance my programs.

CHARACTER REDEFINITIONS

```
SCR #34
  0 ( B/CHEQUE - Character Redefine 14Mar86)      ( CLEAN)   HEX
  1 FF00 FF00 0000 0000 84 CHAR ( 132 TOP  LINE )
  2 FF80 BFA0 A0A0 A0A0 7E CHAR ( 127 UPPER RIGHT CORNER)
  3 A0A0 A0A0 A0BF 80FF 7C CHAR ( 124 BOT LEFT  CORNER )
  4 FF04 F414 1414 1414 7D CHAR ( 125 SUPPER LEFT CORNER)
  5 1414 1414 14F4 04FF 7E CHAF ( 126 BOT RIGHT CORNER )
  6 0000 0000 00FF 00FF 80 CHAR ( 128 ERSE LINE )
  7 A0A0 A0A0 A0A0 A0A0 81 CHAR ( 129 LEFT  VERTICAL )
  8 1414 1414 1414 1414 82 CHAR ( 130 RIGHT VERTICAL )
  9 00FF 0000 0000 0000 83 CHAR ( 131 UNDERLINE )
 10 9090 9090 9090 9090 85 CHAR ( 133 SINGLE LINE RIGHT VERTICAL)
 11 0404 0404 0404 0404 86 CHAR ( 134 SINGLE LINE LEFT  VERTICAL)
 12 0000 0000 0000 00FF 87 CHAR ( 135 SINGLE LINE BASELINE)
 13 8080 8080 8080 80FF 88 CHAR ( 136 SINGLE LINE LOWER LEFT ANGLE)
 14 0404 0404 0404 04FF 89 CHAR ( 137 SNGLE RGHT LEFT ANGLE)
 15 8000 8000 8000 8000 92 CHAR ( 146 BOT VERTICAL      ) DECIMAL
```

As perusal of this screen will see it is far easier to write in HEX. If like me HEX boggles you brain, use the terminal to determine your values eg by typing in 65 HEX . you will get back 41. Rechange back to DECIMAL. To determine the pattern it is exactly the same as in BASIC. Use the pattern identifier conversion table in the BASIC USERS REFERENCE GUIDE. Allocate this pattern as per format to an unused character, and comment next to it what it is supposed to do or you wil become all mixed up.

ROUTINE FOR DRAWING BOXES.

```
SCR #73
  0 ( B/CHEQUE - BOX AROUND TITLES  14Mar86)     ( CHECKED CLEAN)
  1 : F   2 + SF6 + @ ;  ( put copy of nth number to top of stack)
  2 : DBOX                                        ( lc,ur,rc,lr    )
  3       4 P  4 P                 1 123 HCHAR   ( upper left  car )
  4       4 P  OVER                1 124 HCHAR   ( lower left  car )
  5       OVER 4 P                 1 125 HCHAR   ( upper right car )
  6       OVER OVER                1 126 HCHAR   ( lower right car )
  7       4 P 1+ OVER 4 P 1- 7 P - 128 HCHAR   ( lower horizontal)
  8       4 P  4 P 1+ 3 P 1- 6 P - 129 VCHAR   ( right vertical  )
  9       OVER 4 P 1+ 3 P 1- 6 P - 130 VCHAR   ( left  vertical  )
 10       4 P 1+ 4 P  4 P 1- 7 P - 132 HCHAR   ( upper horizontal)
 11                              3DROP 2DROP ;
 12
 13
 14
 15
```

Yes, after blasting use of PICK/ROLL as being sloppy I have used them here, otherwise the routine would be sloppy and intricate as ordinary stack operators don't dig deep enough. This routine is simple enough if you analyse it, it just uses the appropriate values and the HCHAR and VCHAR commands to draw a double lined box, convenient for around messages,titles etc to make you effort look a little prettier. You the user must ensure you put in valid row/column numbers.

lc=left hand column
ur=upper row
rc=right hand column
lr=lower row

Enter these within validity limits and type DBOX and hey presto!

ARRAYS

What is an array?

Rather than confuse you by an abstract definition lets take a practical approach to a problem. Say we decide we are sick of our cluttered desk and our inability to be able to find our paper copies of our recent cheque statements or bankcard statements, and , because of our inegible handwriting on our cheque buts we ca...ot read what we wrote the cheques for anyway. Using our newfound ability to program in FORTH we decide, rather ambishiously to write a program to keep a record of all this. We decide on a basic format. Each entry must consist of the following minimum data:

1 complete record= entry number
entry date
paid to whom
amount paid
reason
category

Lets extract 1 element from this - the amount paid, and allocate a memory space to deposit the data:

```
0 VARIABLE AMOUNT 7 ALLOT  ( allot a 9 byte buffer)
```

This will allow us to accept the number as a string of up to 8 characters (amounts up to $99999.99) and store this with a preceding character count at the address indicated by AMOUNT. If we enter data from the terminal for record 1 fine. If we enter data for record 2 then it will displace the previous data from the AMOUNT buffer. So we have to have somewhere to store successive inputted amounts, but in a way that we keep track of which one belongs to each record.

Lets allocate space for say 10 record amounts, each of 9 bytes:

```
0 VARIABLE AMOUNT 88 ALLOT ( allots 90 bytes of space in
dictionary)
```

Now lets draw a picture:

```
 9bytes        entry#     start Adress
[0---------8]  amount 1   AMOUNT
[9---------17] amount 2   AMOUNT + 9
[18--------26] amount 3   AMOUNT + 18
[27--------35] amount 4   AMOUNT + 27
[36--------44] amount 5   AMOUNT + 36
[45--------53] amount 6   AMOUNT + 45
[54--------62] amount 7   AMOUNT + 54
[63--------71] amount 8   AMOUNT + 63
[72--------80] amount 9   AMOUNT + 72
 [81--------89] amount 10  AMOUNT + 81
```

We can now calculate a formulae for finding the starting address of any particular record amount we may want to access

ENTRY-ADDRESS = ENTRY# -1 * 9 + ARRAY START ADDRESS.

Eg amount1 = 1-1 * 9 + START ADDRESS OF AMOUNT = AMOUNT + 0
Eg amount5 = 5-1 * 9 + START ADDRESS OF AMOUNT = AMOUNT + 36

Lets translate this into FORTH:

```
: ENTRY-START           ( expects record no you wish to access )
              AMOUNT    ( leave start address of array on stack)
              SWAP      ( put record no choice on top of stack )
              1-        ( subtract one                         )
              9 *       ( multiply by 9 to give offset to array)
              +         ( add this offset to start address     )
              ;         ( end definition                       )
```

This concept can be expanded to any situation where you want to keep sequential amounts of similar data. For example an array of balances which could be updated and re-stored within a program, or arrays of whole records.

Short listing today due to lack of time. Next month I will expand on using arrays and moving blocks of data around memory.

ADDRESS FOR CORRESPONDENCE:
R.TERRY
141 DUDLEY RD,
WHITEBRIDGE 2290.
049(436311/22430)

Comment

CONGRATS ON THE HOMEMADE P.E. UNIT.

I HAD A TELEPHONE CALL FROM PETER GLEED OF THE TI MELBOURNE USERS GROUP THE OTHER NIGHT WITH CONGRATULATIONS TO RON KLEINSCHAFER FOR HIS FINE WORK ON THE HOMEMADE PERIPHERAL EXPANSION UNIT . HE WOULD LIKE TO SEE ONE ON THE STAND AT THE TI FAIR IN MELBOURNE ON 14TH JUNE'86 IF WE HAVE ONE READY TO TAKE DOWN . (HE TOLD ME HE HAS ALREADY SET ASIDE A SPACE). HE WANTS TO SHOW OTHERS WHAT A DEDICATED MEMBER OF A USER GROUP CAN DO WHEN FACED WITH A SHORTAGE OF HARDWARE AND A WILL TO SURVIVE. MY CONGRATULATIONS ALSO TO RON FOR A JOB WELL DONE.

TERRY ROSS
HV99ERS.

--------HELP--------

The module library is indesperate REPEAT desperate need of any modules at all. They can be lent or they can be donated. At the moment we have 8 modules and one set of joysticks the modules are:

1.Extended Basic.
2.The Attack.
3.Tombstone City.
4.Mind Challengers.
5 Star Trek.
6.Dragon Mix.
7.Car Wars.
8.Alligator Mix
and the Joysticks.

I would like to thank the people who have lent modules, they are D.Winton for Mind Challengers and J.Grimmond for Dragon Mix and Star Trek. If you are in the market for any modules eg Touch Typing, Extended Basic, Parsec then please contact me so when it becomes available we shall know you are looking for one and will be able to supply you with a copy for more information on any of the above please contact Rodney Gainsford on 583515 or see the Module Librarian at the General Meetings.

Rodney G.

SOME UNUSUAL COMPUTER LANGUAGES

This Article first appeared in the APL SIG newsletter THE SPECIAL CHARACTER SET (D.BOHRER, EDITOR) and has gained steam very since. It then appeared in the Northwest Ohio 99er News 1/86, then the Cleveland Area newsletter 2/86. It then appeared in the Rocky Mountains 99ers "TIC TALK" 2/86. This Group is based in Denver Colorado. It is from this newsletter that the H.V.99'ers. obtained this copy of the Article. Finally thanks to Tony McGovern who recieved that Newsletter and brought the Article to our attention. I hope you find it as entertaining as I did!.

****

Lauguages NOT included in the Commercial Language SIG or the Language Tools SIG.
by Doug Bohrer, Bohrer and company, near Chicago.
and
Ted. A. Bear, NCA Corporation, In the heart of Silicon Valley
and
A Usually Reliable Source, Digital Equipment Corporation, Somewhere in New England.

APL, BASIC, COBOL, FORTRAN, PASCAL, RPG....these programming languages are well known and (more or less) loved throughout the Computer industry. There are numerous other languages, however, that are less well known yet still have ardent devotees. In fact, these little known languages, generally have the most fanatic admirers. For those who wish to know more about these obscure languages -- and why they are obscure -- we present the following catalogue.

C-
This language is named for the grade recieved by its creator when he submitted it as a class project in a graduate programming class. C- is best described as a "low level" programming language. In general, the language requires more C-

statements than machine code instructions to execute a given task. In this respect it is very similar to Cobol.

DOGO.
Developed at MIOT (Massachusetts Institute of Obedience Training). DOGO heralds a new era of computer literate pets. DOGO commands include SIT, HEEL, STAY, PLAY_DEAD, and ROLL_OVER. An innovative feature of DOGO is "puppy graphics", a small cocker spanial that occasionally leaves deposits as it travels across the screen.

FIFTH.
FIFTH is a precise mathematical language in which the Data types refer to quantities. The data types range from CC, OUNCE, SHOT, and JIGGER to FIFTH (hence the name of the language), LITER, MAGNUM, and BLOTTO. Commands refer to ingredients such as CHABLIS, CABERNET, GIN, VERMOUTH, VODKA, SCOTCH, BOURBON, CANADIAN, COORS, BUD, EVER_CLEAR and WHAT_EVERS_AROUND.
The many versions of the FIFTH language reflect the sophistication and financial status of the User. Commands in the ELITE dialect include VSOP, LAFITE and WAITERS_RECOMMENDATION. The GUTTER dialect is a particular favorite of frustrated FORTH programmers who end up using this language.

LAIDBACK.
This language was developed at the Marin County Center for T'ai Chi. Mellowness and computer programing (now defunct), as an alternative to the more intense atmosphere in nearby Silicon Valley.
The Center was ideal for programmers who liked to soak in hot tubs while they worked. Unfortunately few programmers could survive there because the center outlawed Pizza and Coca-Cola in favor of Tofu and Perrier.
Many mourn the demise of LAIDBACK because of its reputation as a gentle and non-threatening language since all error messages are in lower case. For example, LAIDBACK responded to syntax errors with the message:

"I hate to bother you, but I just can't relate to that. Can you find the time to try again?"

LITHP.
This otherwise unremarkable language is distinguished by the absence of an "S" in its character set. Programmers and users must substitute "th". Lithp is said to be useful prothething litht. This language was developed in San Francisco.

REAGAN.
This language was also developed in California, but is now widely used in Washington, D.C. It is the current subset of the international bureaucratic language know as DOUBLESPEAK. Commands include REVENUE_ENHANCEMENT, STOCKMAN, CAP_WEINBERGER, MALCOMB_BALDRIDGE, CABINET, CHIP_WOOD, LAXALT and SCENARIO. WATT and BURFORD have been removed from the commands while there is a current effort to add MEESE.
The operating systems used is NEW RIGHT and the designated memory is THE_RANCH. The compile SCENARIO is a compile with NANCY followed by a link with BONZO resulting in a SNOOZE. COMMIES (programme bugs) are removed with the GRANADA command.
A REAGAN program commences with LANDSLIDE and terminates with SENILITY.

RENE.
Named after the famous French philosopher and mathematician Rene DesCaters, RENE is language used for artificial intelligence. The language is being developed at the Chicago Center of Machine Politics and Programming under a grant from the Jane Byrne Victory Fund. A spokesman described the language as "Just as great as dis (sic) great city of ours".
The center is very pleased with progress to date. They say they have almost succeeded in getting a VAX to think. However, sources inside the organisation say that each time the machine fails to think it ceases to exist.

SATRE.
Named after the late existenial philospher, SATRE is an extremely unstructured language. Statements in SATRE have no purpose they just are. Thus SATRE programmes are left to define their own functions. SATRE programmers tend to be boring and depressing and are no fun at parties.

SIMPLE.
SIMPLE is an acronym for Sheer Idiot's Monopurpose Programming Linguistic Environment. This language, developed at Hanover College for Technological Misfits, was designed to make it impossible to write code with errors in it. The statements are, therefore, confined to BEGIN, END and STOP. No matter how you arrange the statements, you can't make a syntax error.

SLOBOL
SLOBOL is best known for the speed or lack of it, of the compiler. Although many compilers allow you to take a coffee break while they compile, the SLOBOL compiler allows you to travel to Columbia to pick the coffee. Forty three programmers are known to have died of boredom sitting at their terminals while waiting for a SLOBOL programmme to compile.

VALGOL.
From its modest beginnings in Southern California's San Fernando Valley, VALGOL is enjoying a dramatic surge of popularity across the industry. VALGOL commands include REALLY, LIKE, WELL, and Y*KNOW. Variables are assigned with the *LIKE and *TOTALLY operators. Other operators include the California Booleans, A1 and NOWAY. Repetitions of code are handled in FOR - SURE Loops.

Here is a sample programme:

```
     LIKE, Y*KNOW (I MEAN) START
  IF PIZZA          =LIKE BITCHEN AND
  GUY               =LIKE TUBULAR AND
  VALLEY GIRL       =LIKE
                GRODY**MAX(FERSURE)**2
  THEN
     FOR I=LIKE 1 TO OH*MAYBE 100
           DO*MAH - (DITTY**2)
     BARF(1)= TOTALLY GROSS(OUT)
     SURE

     LIKE BAG THIS PROGRAMME
     REALLY
     LIKE TOTALLY (Y*KNOW)
     IM*SURE
     GOTO THE MALL
```

VALGOL is characterised by its unfriendly error messages. For example , when the user makes a sytnax error, the interpreter displays the message:

GAG ME WITH A SPOON!

ADVENTURES INFO CORNER.

with Rodney G.

Hi this month's adventure column will be longer than usual because I have solved an adventure, also I beg, if you Have any hints or if you need any then phone the Adventure HOTLINE.

ADVENTURELAND
****
*Dragons hate bees.
*Bears love honey.
*If lava is a problem then dam it up.
*No swimming in the lake but what about a quicksand bog.
*Rub the lamp.
*A brick wall a problem then blow it up with gas.
*Firestone hot then wet it.
*Mirror breaks use the rug.

PIRATE ADVENTURE.
****
*The key is under the mat.
*Make sure you weigh that anchor once you build the boat.
*If fish dry up then use a bottle of water.
*Pirates don't like books.
*If you fall then say yoho.
*Use the plans.

VOODOO CASTLE.
****
*Window closed keep a good luck charm on you.
*Wave a ring near a stone door.
*Stuck in lower levels a crystal ball reveals all.
*Dust the idol.
*To open safe dial something then dial something.
*Someone stuck then pull them out.
*Remove soot from chimney or it could get dark.

H.H.G.T.T.G.
****
*Junk mail keeps robots busy.
*Listening in the dark can reveal everything if you keep it up.
*Computers lie - look in the engine room.
*Store fluff in bags, boxes and pockets.

ADVENTURE HOTLINE.
--583515--

WE HAVE MOVED

AS FROM 20th MAY, OUR GENERAL MEETINGS WILL BE HELD AT THE WARNERS BAY HIGH SCHOOL.

WANTED

(A) ONE PIECE D/SIDED COPPER CLAD BOARD APPROX SIZE 100 MM X 20 MM FOR USE AS A 30+30 CONNECTOR.

(B) DETAILS AND AVAILABILITY OR INFORMATION ON A TI MOUSE FOR THE TI99/4A ???

CONTACT TERRY ROSS

COMING EVENTS

BASIC CLASSES
April 15 - Input/output & Simple Applications
22 - Decisions, Branching & Applications
29 - Arrays

EXTENDED BASIC CLASSES
April 15,22,29 - Graphics, leading up to Sprites.

GENERAL MEETING MAY
Due to the school holidays, the next General Meeting will be held at our new venue, on Tuesday, 20th May. The demonstration will be Menu Maker with Al Wright.