

TI - 99 USERS OF PERTH

TIUP TIT BITS
VOLUME 14 No. 6
APRIL 1996

TIUP Celebrates 14 Years of Service,
in PERTH - Western Australia,
To the TEXAS INSTRUMENTS TI-99/4A Computer Owners

'After several non-official, yet very friendly and informative get togethers, we have finally taken the plunge, and on Saturday, 20th March, 1982, at the Mt. Lawley Neighbourhood centre, 715 Beaufort Street, Mt. Lawley, a committee was elected'. Secretary/Editor - Tit BITS: Volume 1, Number 1, Page 1, May 1982.



PLEASE NOTE,
OUR NEW MAILING
ADDRESS IS

* Index Page 14*

SECRETARY TIUP
20 HUDSON ST.
BAYSWATER 6053
WESTERN
AUSTRALIA.

EDITORIAL by F.Graham Secretary/Editor

Vale Tony, it was with great sadness that we of TIUP read, in the February issue of the MUNCH Newsletter, of the death of Anthony N Falco (Tony) aged 56, from brain cancer.

Tony's newsletter program contributions were much appreciated, for the subject matter, methodology adopted and the documentation he provided with them.

The members of TIUP wish to convey their heartfelt condolences to his family, and friends in the MUNCH Organisation.

TIUP's 1996 AGM will be held on the 20th. April. After the election of the Committee, a party will be held to celebrate our 14th. anniversary.

TIUP Tit BITS DISCLAIMER

Tit BITS is the OFFICIAL newsletter of the TI-99 Users of Perth (Inc.), Western Australia (TIUP).

TIUP is a non-profit group of Texas Instruments TI-99/4A computer users not affiliated with Texas Instruments or any other computer manufacturer. The views expressed in articles published in this newsletter are those of the author, and not necessarily those of the Editor, TIUP Committee or Group Members. All articles and programmes published herein are considered to be in the public domain unless notification is received to the contrary.

All care is taken that information published is accurate, but NO RESPONSIBILITY will be accepted by Tit BITS, its publishers or contributors, the TI-99 Users of Perth (Inc.) or its members for any adverse results obtained from the application of any information provided in this publication.

Articles published in Tit BITS may be reproduced by other Groups with similar aims, provided that the source and author are acknowledged. Contributions to Tit BITS are invited from both members and non-members.

Please submit any contributions to the Editor care of:

TIUP (TI-99 Users of Perth Inc).
20 Hudson St.
Bayswater 6053.
WESTERN AUSTRALIA.

OUR COMMITTEE

President:

Merve TROWBRIDGE
90 Williamson Avenue,
Belmont, 6104. W.AUSTRALIA.
Tel. (09) 2771091

Vice President:

Bill La FRENTZ
2 Daly Place,
Padbury, 6025. W.AUSTRALIA.
Tel. (09) 4013430

Treasurer:

Greg BUCK
18 Linden Street,
Dianella, 6062. W.AUSTRALIA.
Tel. (09) 2761291

Secretary:

Frank GRAHAM
20 Hudson Street,
Bayswater, 6053. W.AUSTRALIA.
Tel. (09) 2715972

SUBSCRIPTIONS

FULL MEMBER - A\$20-00 P.A.
NEWSLETTER ONLY - A\$15-00 P.A.

ADVERTISING

Members and newsletter subscribers may advertise their PERSONAL computer - related products in Tit BITS free of charge (subject to space limitations and the Editor's discretion). Commercial rates by negotiation.

BILLS BURST No. 4.

Well here I am again for the fourth time, in this series, with my burst on TIUP/DOM04.

(1) ARROW 20 PROGRAM (XBASIC)
Source: MICROpendium, February 1989.

This is from "Program uses arrow for selection" by Russ Stanton. This program is a simple database whose most unique feature is the use of a moving arrow to make menu selection. The program comes with full documentation and needs no further information except that you might want to change Line 39 to end the program.

(2) CLIFF 17 PROGRAM (XBASIC)
Source: MICROpendium, May 1993.

This is from "Cliff Climber" an Extended BASIC update of the original, the BASIC version was written by Mike Ward and upgraded to XB by Lucie Dorais. Instructions are given before the game. (Caution: leave the Alpha Lock key up). The number one joy stick is needed and one or two players can play with the computer keeping a record of the highest score. You press the space bar to select the level you want to play at and then you press either 1 or 2 for the number of players in the game. The computer takes some time, to build up the screen image, be patient. At the end of each game you can return to the selection screen where you can either change the level of skill, number of players or stop the game.

(3) CONVERT 11 PROGRAM (XBASIC)
Source: MICROpendium August 1993.

This is from "Converting TI and PC text files" by Jim Swedlow. When you run this program you are presented with the choice of DF128 to DV80 and add cr's, DF128 to DV80 without cr's, DV80 to just add cr's, DV80 to DF128 and to end the program. I have checked all of the selections and they all do their job without any problems.

(4) DV/IV-CONV 2 PROGRAM (XBASIC)

- (7) FROG1 39 PROGRAM (XBASIC)
- (8) FROGDATA 14 INT/VAR80 (XBASIC)
- (9) FROGGY 14 DIS/VAR80 (XBASIC)
- (10) HIPROG 9 PROGRAM (XBASIC)
- (11) HISCORES 2 INT/VAR80 (XBASIC)
- (13) KEEPFI 2 INT/VAR80 (XBASIC)
- (14) LOAD 24 PROGRAM (XBASIC)
Source: MICROpendium November 1987.

All of these programs are from "Intruder" by Miles McManus. The first DV/IV-CONV was written just to convert the DV/80 file FROGGY to the IV/80 file FROGDATA but it is not required on the disk to play the game and can be left out. The LOAD program on this disk is strickly for Intruder as it sets up the screen and gives the instructions for playing the game, it also opens the HISCORES file and lists the Hero's hall of fame.

FROG1 is the main program and HIPROG is used to create KEEPFI and HISCORES files. The six program segments needed to play this game are LOAD, HIPROG and FROG1 which are XBASIC programs and FROGDATA, KEEPFI and HISCORES which are I/V format files. The complete program, Intruder, took a lot of time to get it to work satisfactory. It has a number of segments in it as well as other programs which had to be typed in to create the other files used in the game. Another problem was the FROGDATA file, the article said to leave the Alpha Lock key down, but neglected to say "hold the shift key down also, when typing in all the 2's".

The idea of the game is to hop over the obstacles using the W,E,D keys when going from left to right and the P,O,K keys when going from right to left. Use the D and K keys for short hops, the O and E keys for medium hops. You will need to use the W and P keys for the biggest hops to jump over some of the obstacles and to trip the switch in the middle of the belt

(continued on page 4)

above the one you are on. This opens the gate so that you can get to the next belt and so work your way to the top where you have to flick another switch to go to the next room.

There are 13 levels or rooms which are: (1) The Dining Room. (2) The Arrow Room. (3) The Appliances Room. (4) The Clothing Room. (5) The Pyramid Room. (6) The Food Room. (7) The Sphere Room. (8) The Alien Room. (9) The Hurdle Room. (10) The Cube Room. (11) The Bronze Room (Code Blue). (12) The Silver Room (Red Alert). (13) The Vault (Gold Room). I have only been able to get to the Alien Room before running out of lives.

You start with 5 frogs and get another when you score 700 points, you get extra frogs as your score increases but I was so busy concentrating on the game I am unable to say when. After you run out of frogs you are presented with the chance to enter your name in the Hero's hall of fame, that is if your score is better than the 10th highest score. You enter your name using the arrow keys to select the letter and then press ENTER. The broken line at the bottom of the screen is a timer, you must get to the top of the screen and flick the switch before it reaches the other side.

(5) FRIRISLE 30 PROGRAM (BASIC)
Source: MICROpendium October 1990.

This is from "Fairisle pattern design" by REGENA. This is another program mainly for 'Knitters' and is used to design Fairisle patterns, and display them on the screen, allowing you to see what they look like and make changes if you want. Further, you can get a print out of the pattern when you have a design that you like. (Be sure to check the printer listed in Line 2050 matches your system). When you do get a print out, you will find it is made up of X and O characters to define the two colors. The program comes with instructions and is very easy to follow.

(6) FAREWELL 54 INT/VAR254 (BASIC)
Source: MICROpendium November 1993.

This is from "Farewell to friends" by REGENA. This is the final program, written for MICROpendium by REGENA, and brings to an end many years of interesting programs from this programmer who will surely be missed. The program is written in BASIC but will run in XB, all you have to do is type in RUN and it will play the tune "Aloha Oe" with the words of "Farewell to Thee" scrolling up the screen.

(7,8,9,10,11) See Program No.4 above.

(12) INTFLAG 37 PROGRAM (BASIC)
Source: MICROpendium May 1993.

This is from "International code flags" by REGENA. This is another program which is written in BASIC, but will run in XB, and is a great program for learning the international code flags used by seamen around the world. When you run this program you are presented with a choice: (1) The complete Alphabet, where you are shown each flag in order and you have to press that key to proceed. (2) Choose Letters, which will display the flag of any letter that you press. (3) A Quiz, which will display a flag and you have two guesses to get it right, before the computer displays the correct answer and then goes on to another flag. (4) End program, you may return to the main menu at any time by pressing ENTER.

(13,14) See Program No.4 above.

(15) MATRIX 19 PROGRAM (BASIC)
Source: MICROpendium September 1993.

This is from "Matrices" by REGENA. It is another educational program written in BASIC but will also run in XB. It is also another program that I do not know much about as I did not do Matrices at school but the program runs O.K. It may be useful to someone who receives a copy and my kids might use it later in their high school years.

(Continued on page 14)

Fun with numbers

Body Mass Index lets you know if you're overweight (and who isn't?)

By HAROLD C. HOYT JR.

We like to play with numbers. Sometimes our playing results in a little controversy. We note with amusement some disagreement amongst authorities about what people should weigh. They can't even unanimously agree that most of us are too fat (which we probably are).

NASA did a lot of research in the Kennedy days about fitness related to the space program. One idea involved the amount of fat vs the amount of bone and muscle. Physics says that mass is equal to weight divided by volume. Since fat is lighter than muscle or bone, you can make a fitness index out of how dense a person is. No, no, I don't mean *that* kind of dense.

Covert Bailey, a fitness authority and author, has been using Archimedes principle to calculate the density of people. Bailey holds fitness seminars at health centers around the country. These are well attended by health professionals who don't want to be as bad off as their customers.

Archimedes principle, for those unfamiliar with it, is based on the legendary ability of Archimedes to make practical use of mathematics. This is back in a time when most people were unsure of how many toes they had.

Supposedly, the king asked Archimedes to find out if the people who made his crown for him had cheated him, pretty much standard practice in those days, but really bad form to get caught. The divine right of kings pretty much encouraged being boiled in oil, beheaded, torn apart by horses or something else equally unpleasant. Gold is

heavier than most other materials, so the denser the crown the better. Can't you picture the king gently moving the crown up and down in his hands and doing a Captain Queeg imitation, saying, "This is too light"?

Archimedes solved the problem by taking the irregular shaped crown and weighing it, supported by a string, and then weighing it again submerged in water. The crown weight scale reads less submerged by the amount of water displaced. $Weight1 - Weight2$ is the weight of the volume of water displaced. Since he knew the weight of water per unit of volume, Archimedes could calculate the volume of the crown and then its density. As usual, the king was being cheated.

Bailey has his concerned health professionals run around doing some benchmark exercises and then performs what to us irreverent types with a humorous view of the world what might be construed as a baptism into their fitness religion. Bailey

has each of his disciples sit on an oversized grocery-style scale supported by a winch over a swimming pool. Suspended weight is recorded and then the candidate is completely immersed and a second reading is taken sometime before he drowns. A completely accurate measure of the candidates' densities results.

We pilfered information from a chart published by *Life Magazine* in February 1995 discussing new research suggesting that a person's Body Mass Index calculation is an indicator of health. Is this a shortcut on the NASA technique for determining fitness? We entered the information from this chart as DATA in QBASIC which is an improved BASIC over GW-BASIC or BASIC-A which traditionally comes with IBM DOS or Microsoft DOS. We have heard that Microsoft does not intend to make any more improvements or upgrades to newer versions of QBASIC. I guess that's okay. It really runs fine the

(See Page)

Ht"	Body Mass Index Table															
	19	20	21	22	23	24	25	26	27	28	29	30	35	40		
58	91	95	100	105	110	115	119	124	129	134	138	143	167	191		
59	94	99	104	109	114	119	124	128	133	138	143	148	173	198		
60	97	102	107	112	118	123	128	133	138	143	148	153	179	204		
61	100	106	111	116	121	127	132	137	143	148	153	158	185	211		
62	104	109	115	120	125	131	136	142	147	153	158	164	191	218		
63	107	113	118	124	130	135	141	146	152	158	163	169	197	225		
64	110	116	122	128	134	140	145	151	157	163	169	174	203	233		
65	114	120	126	132	138	144	150	156	162	168	174	180	210	240		
66	117	124	130	136	142	148	155	161	167	173	179	185	216	247		
67	121	127	134	140	147	153	159	166	172	178	185	191	223	255		
68	125	131	138	144	151	158	164	171	177	184	190	197	230	263		
69	128	135	142	149	155	162	169	176	182	189	196	203	237	270		
70	132	139	146	153	160	167	174	181	188	195	202	209	243	278		
71	136	143	150	157	165	172	179	186	193	200	207	215	250	286		
72	140	147	155	162	169	177	184	191	199	206	213	221	258	294		
73	144	151	159	166	174	182	189	197	204	212	219	227	265	303		
74	148	155	163	171	179	187	194	202	210	218	225	233	272	311		
75	152	160	168	176	184	192	200	208	216	224	232	240	279	319		
76	156	164	172	180	189	197	205	213	221	230	238	246	287	328		

Use the table by locating your height on the left and finding your weight on the row and reading the body mass index at the top of the chart. Within reason, the denser you are the better.

(Continued on page 6)

BODY MASS INDEX —

(Continued from Page)

way it is, but didn't we hear that the programming is never completely done? A rumor has it that QBASIC may even be deleted from future versions of DOS, and we feel that would be a step in the wrong direction.

We love TI X BASIC, but QBASIC has a lot of nice features, if you aren't programming sprites. X BASIC as well as GW BASIC and most other BASICs are tokenized and require an interpreter to convert DOS text or DV/80 text into programs. Not so with QBASIC which can use text as input. An 80-column format with a clone puts more pressure on us TI people to get our act together with an 80-column system, whether it be a Geneve or something else.

Back to our programming. QBASIC is so nice, we do a lot of our programming in QBASIC first and then write an X BASIC version. Although we found the QBASIC version much easier to write, we won't show it for now, since this is supposed to be about the TI. Suffice it to say that identical results were obtained with both versions of BASIC.

The BMI people calculate your Body Mass Index by taking your weight (mass) and dividing it by the square of your height. We used the program to confirm that is what they are doing. All the measurements are *metric*. Your height in inches is converted to meters by dividing by 39.37 inches per meter. Your weight in pounds is converted to kilograms mass by dividing by 2.2 pounds per kilogram. There seems to be some sloppiness in this calculation. We initially did all the calculations to at least four significant figures by using 2.2054 pounds per kg and still had a one pound discrepancy in 15 of their table entries, most of which were low and one entry was two pounds high. One of us is slightly off in the arithmetic, and it appears to be slightly more than could be accounted for by a different method of roundoff.

A little humor/philosophy here. Why use the height squared in computing density? Shouldn't it be height cubed? Cubed would fit if bodies of different sizes remained of the same proportions in all directions. That is, if one person, twice as

The BMI people
calculate your Body
Mass Index by taking
your weight (mass) and
dividing it by the square
of your height.

tall as another, was twice as thick. Volume goes as the cube of a dimension. For the height squared model to work, you would require that a person's average diameter would increase as the square root of the height. The taller you get, the relatively skinnier you would have to be, a really Puritan, non-Tom Jones way of looking at people. Since a basic questionable assumption was made at the very beginning, should we relabel the thing as a pseudo body mass index, or something?

Now looking at the program. Still using the Swartz Emulator and redirecting the print to a file for further manipulation by a word processor, we can edit the results to suit our newsletter. After the table was printed to file, it was edited to present the numbers that were different from the *Life Magazine* numbers in italics.

In doing all this editing, flipping back and forth from running the program, editing it and printing the results, it would be easy to mix up which version of the program was running currently from a pile of papers (Computer myth: Computers are leading us to the paperless society?). So, I added the time of day the program was current to the program ID in line 100 — 8 minutes after 5 p.m.

One really big crack in TI X BASIC is the way that it uses the IMAGE statement. I gave up after a while and customized how the table was to be printed using TAB statements. Lots of unnecessary work. No problem using IMAGE in QBASIC. In order not to print an empty line between table lines, one must force the printer not to add linefeeds with carriage returns, hence the 140 OPEN #1 : "PIO.LF". From the DIMension statement in line 150 the

independent variables HT() in inches and BMI() Body Mass Index (units of kilograms per square meter *not* per cubic meter) are used to calculate the weight that will fit in line 220. The conversion factors of 39.37 inches = 1 meter and 2.2 pounds (the more accurate 2.205 didn't help) per kilogram generate table entries of WT().

You use the table by locating your height on the left and finding your weight on the row and reading the body mass index at the top of the chart. Within reason, the denser you are the better.

BMI3

```

1 !SAVE DISK1.BMI3 !103
100 !Prog by H.C. Hoyt Jr. 7
695 17:08 1048
110 !Calculates Body Mass In
dex (Latest Health Measureme
nt Idea) !139
130 RT$=CHR$(10)&CHR$(13) !03
1
140 OPEN #1:"PIO.LF" !192
150 DIM HT(19),BMI(14),WT(14
,19) !207
155 DATA 19,20,21,22,23,24,2
5,26,27,28,29,30,35,40 !080
160 FOR R=1 TO 19 :: HT(R)=5
7+R :: NEXT R !201
170 FOR C=1 TO 14 :: READ BM
I(C) :: NEXT C !099
180 PRINT #1:" Ht"
-----Body Mass Index Table
-----"&RT$ !183
190 FOR C=1 TO 14 :: PRINT #
1:TAB(4*C+2);STR$(BMI(C));:
NEXT C :: PRINT #1:RT$ !128
210 FOR R=1 TO 19 :: PRINT #
1:HT(R);: FOR C=1 TO 14 !06
1
220 WT(C,R)=INT(2.2*BMI(C)*(
HT(R)/39.37)*(HT(R)/39.37)+.
5):: WT$=STR$(WT(C,R)) !231
230 PRINT #1:TAB(4*C+4-LEN(W
T$));WT$;!124
240 !CALL KEY(0,K,S):: IF S=
0 THEN 240 !039
250 NEXT C :: PRINT #1:RT$ !
109
260 NEXT R !232
270 CLOSE #1 !151
280 END !139

```

CHILD'S FIRST LOOK AT MULTIPLICATION

by Tony Falco

"Dad, today I heard someone talk about three times four. What does that mean?"

"Get that jar of pennies in your room and I'll try to show you. Three times four means three fours. Make a row of four pennies. Now two more rows like that one. That's four plus four plus four or twelve. You can see that it is also four rows of three each. Four times three is the same as three times four. Hey, we could do this on the computer."

The program listed below grew out of a conversation much like the one above. I strongly believe that understanding numerical concepts at all levels is aided by, or maybe even only possible with, a physical or mental picture that gives the learner a comfortable and familiar feeling. With this in mind, I try to help my child build a conceptual basis for ideas that tend to become rote with time. With a program that is conceptually sound, the computer's infinite patience provides an ideal means for acquiring insight into basic numerical concepts.

To use the program you pick the highest factor the child is to work with. For example, for products up to 6×6 or 36 pick 6. The program randomly picks problems and displays the problem with an array of asterisks to show the factors to be multiplied. The user types an answer. If wrong he tries again until correct. The format in which the array appears on the screen encourages the child to try repeated additions or, as a last resort, counting. To end the program enter <Q> instead of an answer. The interaction between computer and learner is reinforced with speech. I feel this makes a program more friendly and familiar. The TI-99/4A has been, and continues to be, the only machine for inexpensive, high quality and easy to use speech.

```
10 FOR S=1 TO 14 :: CALL COL
OR(S,2,12):: NEXT S
20 CALL SCREEN(12):: CALL CO
LOR(2,2,16):: CALL CLEAR
30 INPUT " HIGHEST FACTOR=>
":H
40 RANDOMIZE :: CALL CLEAR
50 A=INT(H*RND)+1 :: B=INT(H
*RND)+1
60 IF A*B>H*H THEN 50
70 P$=STR$(A)&"x"&STR$(B)&"=
" :: DISPLAY AT(5,12):P$
80 S$=RPT$("x",B)
90 FOR T=1 TO A
100 DISPLAY AT(T+7,14-LEN(S$
)/2):S$ :: NEXT T
110 K=12-LEN(S$)/2+(A>9)
120 DISPLAY AT(8+INT(A/2),K)
SIZE(-2):STR$(A)
130 DISPLAY AT(9+A,13):B
140 ACCEPT AT(5,12+LEN(P$)):B
EEP:ANS$
150 IF ANS$="Q" OR ANS$="q"
THEN 180
160 IF VAL(ANS$)=A*B THEN R=
R+1 :: CALL SAY("GOOD"):: GO
TO 40
170 CALL SAY("SORRY TRY AGAI
N"):: W=W+1 :: GOTO 140
180 CALL CLEAR :: PRINT ::;
::; PCT=INT(100*R/(R+W)+.5)
190 PRINT " ";W+R;"TRIED"
200 PRINT " ";R;"CORRECT"
210 PRINT " ";PCT;"%"
220 CALL SAY("GOOD BYE")
230 END
```

BASIC stores your program in two sections. In the top of memory it stores each line of the program, not necessarily in the correct order. As a matter of fact each time you edit a line, it becomes the last line in this area, with all other lines packed together above it. Each statement is made up of three parts. The first byte is the length of the rest of the statement in memory. The last byte is zero, and in between are bytes that represent the particular BASIC statement you have written. BASIC keywords are translated into a single byte each (known as a token) while strings and numeric constants are represented as a leading token (199 or 200) followed by a length byte, followed by the ASCII character values of the string. By running this program you can determine how other elements of a BASIC program are stored.

Underneath the statements (that is, lower in memory) is a list of statement numbers and pointers to the first token in each statement. Each statement in your program has a four-byte entry in this list. The bottom two bytes store the statement number. The top two bytes are a pointer to the first token in the statement (the byte following the length byte). This program goes through this list and prints out each token in the statements of your program.

Pointers to the top byte in the statement pointer list and the bottom byte in the list are stored in the scratchpad RAM and read by lines 150 to 180. The loop that starts in line 190 examines each statement in the program. If you have gotten this far in the article, you will understand how the rest of the lines in the program print out each token of each line.

This is one of a series of articles I will be writing for 99/4A newsletters. I hope you found it interesting. Permission to copy this article for publication is given on two conditions: first, that these conditions on publication be published with the article and second, that copies of the next three issues of the publishing newsletter be sent to me at the following address:

Tim MacEachern
P.O. Box 1105
Dartmouth, N.S.
Canada B2Y-4B8

```

100 DEF HEX1(X$)=POS("123456789ABCDEF",X$,1)
110 DEF HEX2(X$)=HEX1(SEG$(X$,1,1))+HEX1(SEG$(X$,2,1))
120 DEF HEX4(X$)=HEX2(SEG$(X$,1,2))+HEX2(SEG$(X$,3,2))
130 DEF HEX(X$)=HEX4(SEG$("0000"X$,LEN(X$)+1,4))
140 DEF MA(X)=X+65536*(X>32767)
150 CALL PEEK(MA(HEX("8332")),A,B)
160 TOSL=MA(A6+B)
170 CALL PEEK(MA(HEX("8330")),A,B)
180 BOSL=MA(A6+B)
190 FOR PTR=TOSL-3 TO BOSL STEP -4
200 CALL PEEK(PTR,A,B,C,D)
210 PRINT "STATEMENT #";A6+B
220 PRINT "TOKENS:"
230 SPTR=MA(C6+D)
240 CALL PEEK(SPTR-1,L)
250 FOR I=0 TO L-1
260 CALL PEEK(SPTR+I,X)
270 PRINT X;
280 NEXT I
290 PRINT :
300 NEXT PTR
    
```


NOTES

At the last PUNN meeting we took up the 'INTERNET'. For your information I have checked into this and have written the following report

INTERNET

The Internet was set up to connect schools around the US, so that their students, teachers and the different schools could interchange information. When first started, only the above could use it.

A few years ago they started letting others join and now more than 20,000,000 computers are connected world wide.

On the Internet the following can be called up.

1. Message base called e-mail
2. Chat line. These are places that a group can talk with other people on a central ideas (TI, Ham Radio, Food, Girls, etc).
3. Web Sites. These are both small and large files on information. Can be catalogs for you to use to place orders, school topics, general information, weather, pictures, airline flight scheduls, etc.
4. Private information that some services have only for their own members.

On Internet there are different ways you can join. The price can vary from \$.20 an hour to \$2.00 an hour. You have to check on the different ways and see which one is the best for you.

The following are some of the different ways you can join.

1. The Retail end. This is the most expensive way. The cost for the first five hours is \$9.99 and \$2.50 to \$2.95 an hour thereafter each month. These are companies like Compuserve, America-On-Line, etc. They do have a few files that other people cannot use but about 99% of everything on Internet is open for all. Also, with these people you need a credit card as they will bill you the first of each month.
2. Independent Connections. They are set up for schools, businesses, and other people that do not want to pay the high price that the above services charge.

There are about 20,000 independent companys in the world that does this. These companies charge from \$.20 a hour up to \$.50 a hour with minimums from 30 to 120 hours a month.

The following is a listing of two of them. TRANSPORT LOGIC charges \$.50 an hour with a minimum of 30 hours a month (\$15.00) or \$.20 an hour for a minimum of 120 hours a month (\$25.00). This covers every thing.

AGORA charges \$.20 an hour for text only a month (\$6.00) or \$.30 an hour (\$9.00) for everything. This is for 30 hours a month.

They are both a lot cheaper then the other services, BUT there are a lot more companies in Portland that do this. These are the only 2 that I have checked on.

If anyone would like to contact PUNN via the Internet you will be able to do so after the 1st of March 1996. The complete address is not available yet but will be posted on the PUNN BBS as soon as known. This information will also be shown in next month's Wordplay.

Messages addressed to PUNN which are received on the Internet will then be entered on the PUNN BBS.

If you need the address or phone number of any of the above services call up Ron, Ted or look on the PUNN BBS.

-Ted-

WHERE'D THEY GO?

Over the past year or so the TI community kept hearing of things to come such as the SCSI card which was to open many new things that the TI could do. Sure would be nice if it still did happen even though the number of possible users is getting very small.

Would also be nice to hear from anyone who is using a TI-994A in accessing the Internet. What programs are you using? What is your equipment configuration? What problems have you run into? Let us know so we can pass the word along.

-Ron-Ted-



Maze Maker
by Steve Karasek



This program will print mazes for you to solve. It asks for the number of mazes to print, then for the level of difficulty, from 0 to 9. Level 0 is a VERY trivial maze (a child's first maze, perhaps), while level 9 is fairly challenging. The level number is printed at the top of the maze.

No matter what level you select, the maze will be printed to fill as much of the page as possible, so the lower-level mazes will have wider pathways which are easier for young children. There will always be exactly one path from Start to Finish.

The higher-level mazes take a while to compute. In particular, level 9 mazes take over 20 minutes each. You can always start up the program and come back a few hours later. The program keeps track of how far it has gone in computing each maze by displaying a line of the form M / N on the screen, where N is the number of squares in the maze and M is the number of squares the program has computed a path to. When M equals N, the maze is done and is sent to the printer.

If your printer is not named "PIO", change the name in line 110. The last part of this line sets the printer line spacing to 7/72 inch. If you do not have an EPSON-compatible printer, you will have to change this to the codes needed by your printer to set the line spacing. If you can't set it to 7/72 inch, set it to 8 or (preferably) 10 lines per inch.

The !'s and numbers at the end of each line are the checksums for Tom Freeman's CHECKSUM program, and are not needed by the maze program.

```

*****
* MAZE - THE PROGRAM *
*****

100 RANDOMIZE :: OPTION BASE
1 :: DIM M(39,39):: INPUT "
HOW MANY MAZES? ":Z :: PRINT
1223
110 INPUT "LEVEL OF DIFFICUL
TY(0-9)? ":L :: IF L(0 OR L)
9 THEN 110 ELSE OPEN #1:"PIO
",OUTPUT :: PRINT #1:CHR$(27
);"A";CHR$(7);1131
120 N=INT(L+1)*4+(L=4 OR L=9
):: X=80/N :: S=INT(X):: S=S
+(X=S)1138
130 PRINT #1:"Start";TAB(30)
;"Level";L :: FOR X=1 TO N :
: FOR Y=1 TO N :: M(X,Y)=0 :
: NEXT Y :: NEXT X :: IF N=3
9 THEN 150 !174
140 FOR X=1 TO N :: M(N+1,X)
,M(X,N+1)=16 :: NEXT X 1203
150 C,X,Y=1 :: DISPLAY ERASE
ALL AT(12,12):"1 /";N*N ::

ON ERROR 290 !059
160 V=INT(RND*4):: DX=X+(V=0
)-(V=1):: DY=Y+(V=2)-(V=3)::
K=M(DX,DY):: IF K THEN
160 !229
170 M(X,Y)=M(X,Y)+2^V :: IF
INT(W/2)*2=V THEN V=W+1 ELSE
V=W-1 !125
180 X=DX :: Y=DY :: M(X,Y)=M
(X,Y)+2^W :: C=C+1 :: DISPLA
Y AT(12,9)SIZE(4):USING "###
#"C :: IF C=N*N THEN 240 10
53
190 IF X<N THEN IF M(X+1,Y)=
0 THEN 160 1198
200 IF Y<N THEN IF M(X,Y+1)=
0 THEN 160 1199:
210 IF Y=1 THEN IF M(X,Y-1)=
0 THEN 160 1117
220 IF X=1 THEN IF M(X-1,Y)=
0 THEN 160 1116
230 X=INT(RND*N)+1 :: Y=INT(
RND*N)+1 :: IF M(X,Y)THEN 19
0 ELSE 230 !248
240 ON ERROR STOP :: PRINT #

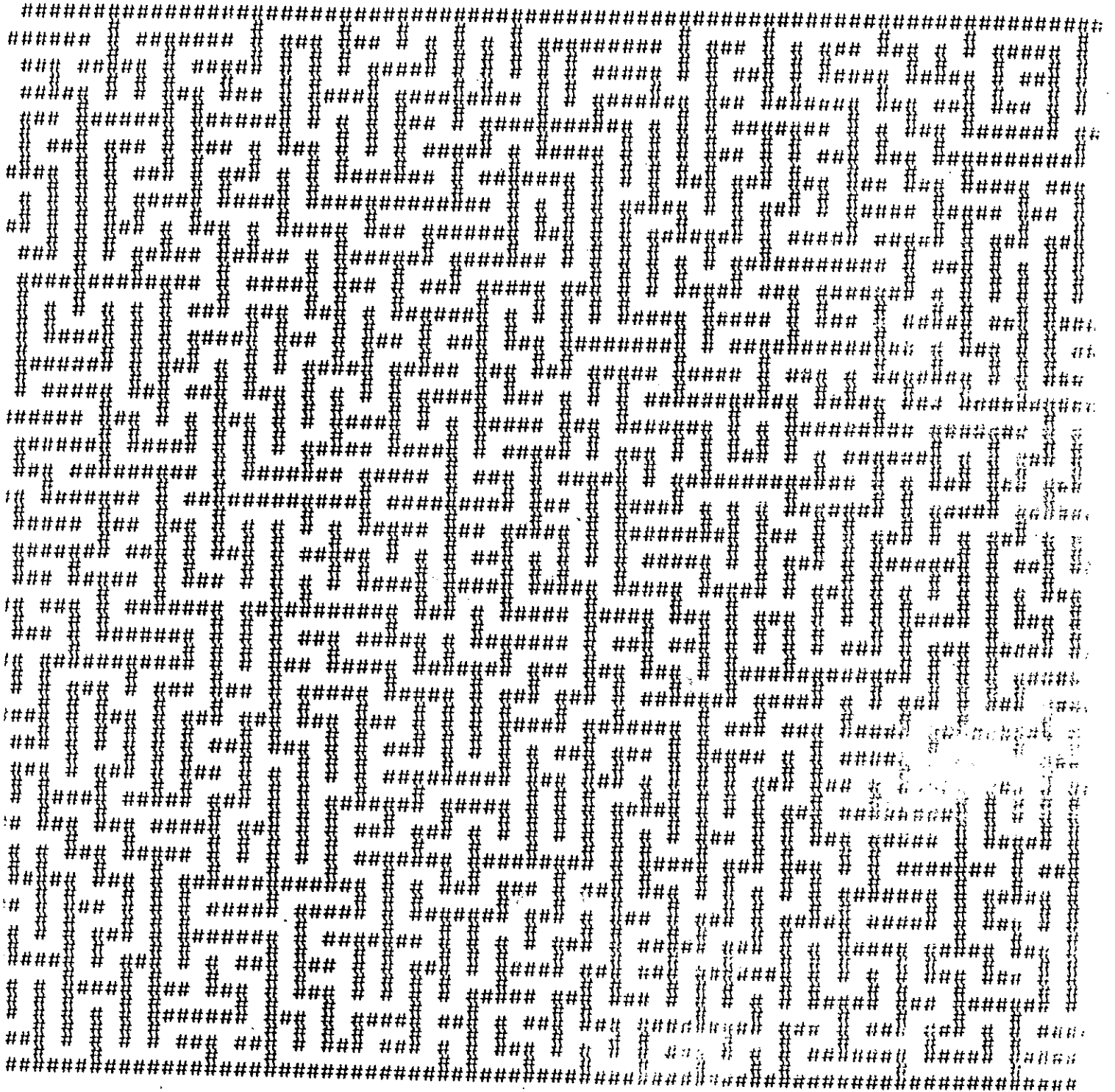
1 :: PRINT #1:"";TAB(S+1);R
PT$("A",S*(N-1)+1):: S=S
-1 :: S$=RPT$(" ",S):: I$=RP
T$("A",S)1069
250 M(N,N)=M(N,N)+8 :: FOR Y
=1 TO N :: FOR W=1 TO S :: P
RINT #1:"";:: FOR X=1 TO N
:: PRINT #1:S$;1076
260 IF M(X,Y)AND 2 THEN PRIN
T #1:"";ELSE PRINT #1:"";1
084
270 NEXT X :: PRINT #1 :: NE
XT W :: PRINT #1:"";:: FOR
X=1 TO N :: IF M(X,Y)AND
8 THEN PRINT #1:S$;ELSE PRI
NT #1:X$;1244
280 PRINT #1:"";:: NEXT X :
: PRINT #1 :: NEXT Y :: S=S+
1 :: PRINT #1 :TAB(S*N-4);"
Finish":CHR$(12);:: Z=Z-1 ::
IF Z)0 THEN 130 ELSE END !0
20
290 ON ERROR 290 :: RETURN 1
60 !159

```

EXAMPLE OF A MAZE FROM "MAZE MAKER"
by Steve Karasek

start

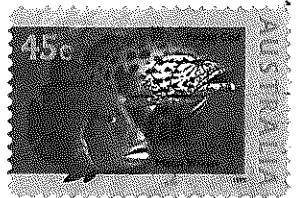
Level 9



Finish

SPIRIT OF 99

 PRINTED MATTER
 ONLY



TO:
 TIBUG

18 ZAMMIT STREET
 DECEPTION BAY 4508, QUEENSLAND.

SENDER:

SECRETARY - TIUP(INC.)

20 HUDSON STREET,
 BAYSWATER 6053.
 WESTERN AUSTRALIA.

(Continued from page 4)

(16) TIME 38 PROGRAM (BASIC) - Source: MICROpendium July 1993.

This is from "Telling Time" by REGENA. This is yet another educational program and one of my kids found it very helpful when learning to tell the time on both analogue and digital clocks. Instructions are given with the program and after selecting one of the two choices, you must get 5 in a row correct before the computer will return to the main menu. If your answer is correct, you are asked to press ENTER to continue. However, if you are wrong the correct answer will be displayed, for you.

(17) TRUMPET 22 PROGRAM (BASIC) - Source: MICROpendium August 1993.

This is from "Playing the trumpet" by REGENA. The program is written in BASIC, but will also run in XBASIC. You are given instructions throughout the program with 4 choices in the main menu: (1) Play scale. (2) Choose notes. (3) Sample tune. (4) End program. This is a good program for those of you who are musically inclined.

For those of you who have a copy of this disk I hope this burst has been helpful, especially if you have been having problems, with any of the programs in it. Well I think that is all there is for now, so goodbye and happy Tling - Bill La Frenz.

TITLE	PAGE	TITLE	PAGE
Bill's Burst #4 (TIUP/DOM04)	3	Advertising	2
Body Mass Index	5	Our Committee	2
Child's First Look at Multiplying	7	Subscriptions	2
Cover Storey	1	TIUP Tit BITS - Disclaimer	2
Editorial	2		
Exploring Basic Programs	9		
Internet	11		
Maze Maker	12		
TI Users Learn to Navigate	8		
