YESTERDAY'S NEWS

VOLUME 3 NUMBER 1 Established 2016 JANUARY 2018

VAST TI99ERS

INSIDE INFORMATION

30 Years Ago...

Historical Information taken From Bill Gaskills TIMELINE

JANUARY 1988:

Myarc announces software development plans for the 9640 Geneve computer. Included are a GPL interpreter, MDOS v1.1, the GEME graphical interface, a runtime version of Pascal, Advanced Basic and computer aided design software.

John H. Carver of Bringhurst, Indiana announces the creation of a Forth Clearing house for TI Forth programmers.

Grand RAM begins shipping in limited quantities, five months after its August 1987 announcement.

GRAMulator prototype model is completed by MarK Van Coppenole of CaDD Electronic s.

J. Peter Hoddie releases an enhanced version of Paul Charlton's Fast Term terminal emulator and asKs that contributions be sent to Charlton. The enhancements include:

CATALOG Changes: The disK cataloging routine now handles any drive number from -9.
FILE COMMANDS Added: Three new Key combinations to activate file commands:
FCTN-Shift-K -- allows a file to be deleted.
FCTN-Shift-L -- protects a file.
FCTN-Shift-U -- unprotects a file.
FILE TRANSFER Changes: xmodem file transfer enhancements to fix problems with GEnie and PC Pursuit downloads.
LOG Changes: The Log File is now opened in APPEND mode rather than OUTPUT mode so existing files are not accidently destroyed.
MISCELLANEOUS Changes: DSRLNK routine searches for devices starting at CRU address >1200 through >1F00 and then searches >1000 and >1100 to allow for better performance with certain ram disK configurations.
Inverse video has been disabled.
Parity bit is now set correctly when using the ASCII send file.
Some screen messages were modified to conserve memory.

Barry Boone releases Archiver II v2.4.

Asgard Software releases a bug fix for the Donn Granros authored Legends adventure game program.

MarK BecK releases Creative Filing system v7.0.

Announcement of the FirstBase Data Base Manager by Warren Agee appears in the Boston Computer Society newsletter.

The Johnson Space Center Users Group develops the 99/4A DATA BASE , listing the published location of numerous articles and such in commercial publications, User Group newsletters etc.

The January 1988 issue of the Long Island TI 99er Users Group, on page 4, touts the MBX TESTER from Texaments. *"Dust off your old MBX Expansion System and get it ready for a new line of utility software. Texaments is proud to announce its BX Tester; a program specifically designed to test all the functions of your MBX system. Also coming are drivers to allow software to be written for the MBX system. That's right. You will be able to use the power of your MBX system as you should have years ago. You may looK forward to some pretty original software in the near future, and if you have any ideas please pass them along."* Neither the MBX TESTER nor any of the promised software would ever appear.

Jim Horn is sysop of the TI Forum on CompuServe, Scott Darling is sysop of the TI Roundtable on GEnie, Blaine Crandall is sysop of the TI SIG on The Source, Walt Howe is sysop of the TI SIG on Delphi.

John Johnson releases Menu v7.1 for the Horizon Ram DisK owner.

YESTERDAYS NEWS IS PUBLISHED MONTHLY BY RALPH REES (SPARKDRUMMER). COMMENTS, SUGGESTIONS, IDEAS, ETC ARE ALWAYS WELCOME

ELEMENTS OF BASIC

By Dave Howell

Courtesy of the Erie 99'ers

Part 3

NUMERIC DATA

Most computers use the floating decimal point as the method of representing numeric data. Any number of digits can be placed on either side of the decimal point. Even with numbers having no decimal position, a decimal point is always assumed following the number's last digit.

Floating point numbers are displayed with 10 digits of accuracy. For example, the following entry of a 10 digit number would generate a 10 digit display:

```
PRINT .5666666666
 .5666666666
```

If an 11 digit floating point number was entered, the last digit would not be displayed and the number would be rounded as follows:

```
PRINT .56666666666
 .5666666667
```

Commas may not be included with numeric data. For example, 109700 would be a valid entry, while 109,700 would be treated by the computer as two separate entries - 109 and 700.

Floating point numbers include signed integers as well as numbers with decimal points. Examples:

```
 -.0789 5 7000 0 +.000001 67.98>
```

Negative floating point numbers must be preceded by a minus sign (-). Positive floating point numbers can be preceded by a plus sign (+) but a number is assumed positive if it doesn't have a sign.

TI BASIC uses scientific notation to express integers containing 11 or more digits. The following are examples of floating point numbers expressed in scientific notation:

```
PRINT 12345678901
 1.23457E+10

PRINT -575000000000
-5.75E+11

PRINT .00000066666666
 6.66667E-07
```

In the first and last examples, the number is truncated at the 6th non-zero digit. Any additional digits are rounded off.

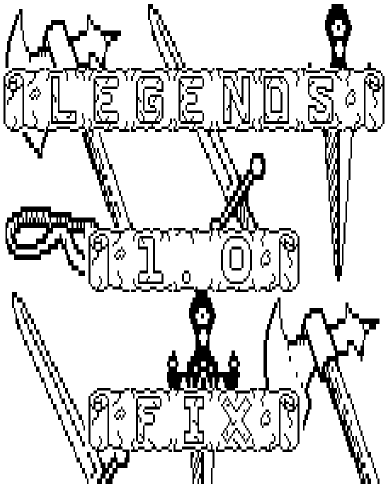
Numbers cannot be displayed with an exponent greater that 99. Numbers that have exponents greater than 99 are displayed with two asterisks in place of the exponent. Example:

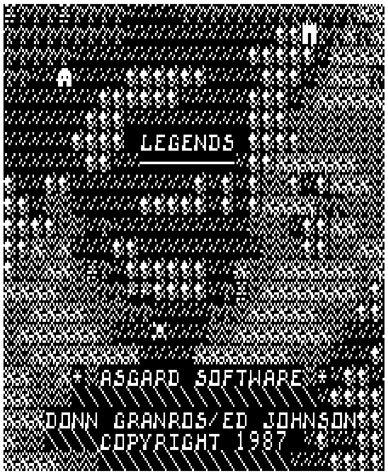
```
2.47685E+**
```

Numbers that require an exponent that is less than -99 are automatically converted to zero. Further the computer cannot manipulate numbers that have an exponent greater than 127. As a result, the following warning will be displayed:

```
* WARNING
NUMBER TOO BIG
```

DH





LEGENDS

ASGARD SOFTWARE
DONN GRANROS/ED JOHNSON
COPYRIGHT 1987

Any large project such as Legends is bound to have some errors. In fact, it is axiomatic that virtually no program is completely bug free. This file is not an apology, it is a recognition of this fact. The plain truth is that we spent 6 months debugging the program, and compared with what we started with Legends 1.0 is virtually bug free. The Keyword here is "virtually". There are a few bugs, and this article will explain how they can be fixed.

Before you make these changes, it is worth noting that Asgard intends to announce details regarding Version 1.1 of Legends, and how to obtain it. You may want to save yourself a lot of trouble as not only does the new version incorporate all these changes, it also has dozens of new features, new spells and other things (wouldn't want to give it away yet, would we?!). We haven't see a price yet, but it will be reasonable.

All of the bugs are located in programs on the DUNGEONS disk. The 2 major programs on this disk are called LGDN/MON and LGDN/TXT. Only those owners with some experience in Extended BASIC programming should attempt these corrections. Before making these changes it is recommended that if you haven't already, make a backup copy of the DUNGEONS disk. Place this backup in drive one and select Extended BASIC from the main menu. You should soon see the Extended BASIC prompt.

First, type OLD DSK1.LGDN/TXT and press ENTER. This program, if you haven't guessed, is the text dungeon section of Legends. Only 2 lines have to be changed in this file. First, bring up line 1470 - it should look like this:

```
1470 CALL DP(22,"AN ICON.","
READ IT@ Y OR N"):: CALL V :
: ON V GOTO 1475,1905
```

The error is on the 2nd screen line of the program line. The program is calling the function V. There is no function V, but there is a variable V. The line should be changed to:

```
1470 CALL DP(22,"AN ICON.","
READ IT@ Y OR N"):: CALL K(V
):: ON V GOTO 1475,1905
```

As you can see, we changed CALL V to CALL K(V).

The next change is only one line down, the line is currently:

```
1475 CALL C(1)...
```

It should be changed to:

```
1475 CALL C(16)...
```

Now, save this program to disk by typing SAVE DSK1.LGDN/TXT and ENTER. Next load in the program LGDN/MON - the monster dungeon. There are a number of errors in this section. The first is an addition that should be made to line 31. Currently, the line reads:

```
31 N,A,AF,P,V,D,DL,=0 :: H$
,I$,C$,ZL$
```

Simply add the variable MX to the first string of variable assignments. In other words, the line should look like this:

```
31 N,A,AF,P,V,D,DL,MX=0 ::
H$,I$,C$L$
```

After making that change, bring up line 1589. The line currently reads:

```
1589 IF D=11...
```

Change it to:

```
1589 IF MX=11...
```

The next change is in line 1935. It now reads:

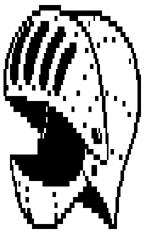
```
1935 IF B<30 THEN...
```

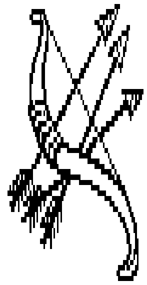
Change it to read:

```
1935 IF B<40 THEN...
```

The final change is in line 2700. This is a very long line, but the change is very small. The line reads:

```
2700 READ @A,@H,@C,@D,@K,@R,
@S,@T,@G,@B,@Q,@E,@F,A,B,MX,
Z$ :: CALL LINK("M","DSK1.W"
&STR$(D))
```





Change it to:

```
2700 READ @A,@H,@C,@D,@K,@R,
@S,@T,@G,@B,@Q,@E,@F,A,B,MX,
Z$ :: CALL LINK("M","DSK1.W"
&STR$(MX))
```

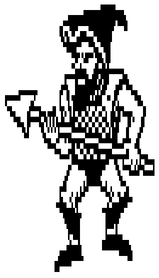
That's basically it. These corrections will fix bugs in reading the icons, saving screens, and moving around within dungeons.

If you don't want to or can't make these changes DON'T send the dungeon disk back just yet. As I said above, we will be announcing a new version of Legends in a matter of a few weeks, which will not only incorporate all these changes, but will also have a lot of new features which, if you like the game, you'll love. They not only make the game easier to play, they also make it a much better game, and more "realistic".

This file was put up as a service to Asgard Software customers.

Thank you.

Chris Bobbitt
Feb 1, 1988



BURGLAR ALARM

"DON'T WORRY FOLKS, OL' TEX WILL STOP THEM CROOKS!"

The program listed allows you to use your spare TI CONSOLE as a burglar alarm with very little investment except for a bit of time.

The actual program is very simple and can be modified to suit your own particular needs. This particular version has a lot of statements that allow you to see what is going on in the program while running a demonstration, however they can be removed quite easily with no effect on the operation of the program. Just a few cautions though. Understand the program first before making any drastic changes. The other precaution is not to use your perimeter loop on the same joystick "direction" as the entry keyswitch. (eg, if you use the UP position for the keyswitch do not use this direction for the perimeter loop even if it is opposite joysticks)

The program is set to use the UP position of joystick 2 for the entry keyswitch and the DOWN position of joystick 1 for the perimeter loop. It is also possible to use the other joystick directions (with appropriate program mods) to have more than one loop. Remember, this program will run as a standalone routine but is intended to be modified or totally rewritten by yourself to suit your particular job. The intention of this program was to be as simple as possible and not require any peripherals or modules. Most of us have a second console so here is a good use for it other than a paper weight.

To set up the alarm you will need the following:

1. TI console

2. Normally open magnetic or pushbutton switches for each door on the perimeter loop. (Radio Shack #49-495 or #49-497) with changes to the program (using the fire buttons and other positions) you may add other protection loops but you must insure that you have one switch per loop when using the normally closed switches. You may use as many switches of the normally open version on the loop as you wish.

3. Entry keyswitch (Radio Shack #49-515) or a hidden SPST toggle switch.

4. An audio amplifier and speaker(s) (your stereo amplifier will work just fine but the alarm will only be sounded in the house)

5. A cable to hook the audio out port from the console to the amp. (If you have a monitor cable these will work fine, Some are available for the TI from Super Valu Stores for $10.95)

6. Joystick connector (Radio Shack #276-1538)

7. Hook-up wire

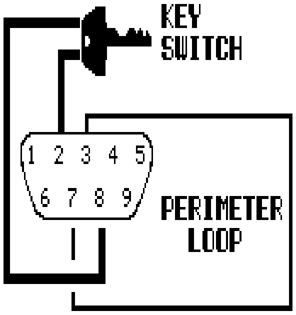
To run a simple demonstration of the program you will need two joysticks and your tv or monitor. First, you may want to set the delay variables in lines 150 and 160. Line 150 is the exit delay variable. This allows you time to leave the house after you turn on the keyswitch. If you mount the keyswitch outdoors, then set this variable to 1. The variable in line 160 sets the entry delay. This one allows you time to enter your home and disarm the system with the keyswitch before the alarm sounds. Remember to set this one on the fast side because it also delays in the event of a break-in.

When you type RUN, the words "PLEASE REMOVE ALPHA LOCK" and "PRESS 'C' TO CONTINUE" appear. Follow the instructions and next comes "PERIMETER CHECK (Y/N)?". If you press "Y" the program jumps to line 700 and checks joystick 1 for any openings in the protection loop. If an opening is found (such as J1 in the center position) the program sounds a warning and tells you too check and remedy the situation. Do this by moving J1 to the down position and holding it there. Now push the "R" key and the program goes back to line 310 and sounds the OK chime.

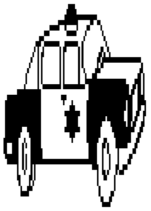
THE WORD "UNARMED" appears and tells you that the system is now ready for input from the keyswitch. When you turn the keyswitch on (by holding J2 in the UP position and J1 in the DOWN position) the program goes to the exit delay loop. This loop allows you to leave your home without triggering the system. Once this times out the program begins looping and checking each of the joysticks for a change in state.

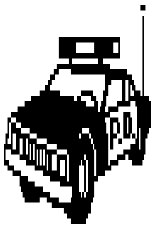
If J1 suddenly becomes open the program moves to the entry delay loop. This delay allows you to enter your home and disarm the system with the keyswitch without setting it off. If the timer times out (eg.break-in) the program now sounds the alarm. You can simulate this by letting J1 return to the center position. Even if you were to close the door now it is too late, the timer is running down and the only way to stop it is too disarm the system.

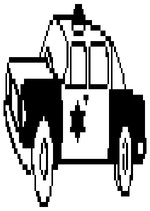
Any number of changes and additions can be made to the program limited only by your imagination and your requirements. The intention of this routine was to give you an idea of what is possible and also to be as simple as possible. There are also heat detectors available that work on the normally open and normally closed switch principals so a fire alarm can also be added.

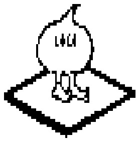


```
100 REM  BURGLAR ALARM PROGR
AM
110 REM  FOR THE TI HOME COM
PUTER
120 REM  A PUBLIC DOMAIN PRO
GRAM
130 REM WRITTEN BY R.A.LUMSD
EN-WINNIPEG,MANITOBA,CANADA
140 REM  85/11 HUG-TIBBS
150 ENDEL=1000
160 EXDEL=1000
170 SKIPD=1
180 CALL CLEAR
190 PRINT "PLEASE REMOVE ALP
HA LOCK"
200 PRINT
210 PRINT
220 PRINT "PRESS  'C'  TO CO
NTINUE"
230 CALL KEY(3,M,N)
240 IF N=0 THEN 230
250 IF M<>67 THEN 230
260 CALL CLEAR
270 PRINT "PERIMETER CHECK(Y
/N)?"
280 CALL KEY(3,L,T)
290 IF T=0 THEN 280
300 IF L=89 THEN 730
310 IF L<>78 THEN 280
320 CALL SOUND(1000,440,0,33
0,5)
330 CALL CLEAR
340 PRINT "UNARMED"
350 CALL JOYST(2,X,Y)
360 IF Y<>4 THEN 350
370 IF SKIPD>1 THEN 400
380 GOSUB 660
390 SKIPD=SKIPD+1
400 CALL JOYST(1,A,B)
410 IF B=-4 THEN 350
420 CALL CLEAR
430 PRINT "ALARM TRIPPED"
440 PRINT
450 PRINT "ENTRY DELAY INITI
ATED"
460 FOR ENTRDEL=1 TO ENDEL
470 NEXT ENTRDEL
480 CALL JOYST(2,X,Y)
490 IF Y=0 THEN 330
500 FOR LOOP=1 TO 5
510 FOR SIREN=700 TO 900 STE
P 10
520 CALL SOUND(-99,SIREN,0)
530 NEXT SIREN
540 FOR SIREN=900 TO 700 STE
P -12
550 CALL SOUND(-99,SIREN,0)
560 NEXT SIREN
570 NEXT LOOP
580 CALL CLEAR
590 PRINT "ALERT !!!!!!"
600 PRINT
610 PRINT
620 PRINT "ALARM TRIPPED"
630 PRINT
640 PRINT "PLEASE RESET"
650 END
660 CALL CLEAR
670 PRINT "EXIT DELAY INITIA
TED"
680 FOR DELAY=1 TO EXDEL
690 NEXT DELAY
700 CALL CLEAR
710 PRINT "ARMED"
720 RETURN
730 CALL CLEAR
740 CALL JOYST(1,A,B)
750 IF B=-4 THEN 320
760 CALL SOUND(1000,-2,0)
770 PRINT "BREAK IN PERIMETE
R CIRCUIT"
780 PRINT
790 PRINT "PLEASE CHECK"
800 PRINT
810 PRINT "PRESS 'R' TO RECH
ECK        PERIMETER"
820 CALL KEY(3,K,S)
830 IF S=0 THEN 820
840 IF K=82 THEN 730
850 IF K<>82 THEN 820
```









Review
by
Charles
LaFara

CUBIT™

Just as the International 99/4 Users-Group had predicted, third-party vendors are now beginning to produce high-quality software for the 99/4 Home Computer. We here at the Users-Group have just recently finished reviewing a copy of a new game from Artios Software. The game is called Cubit, and the game's action is based on a well-known arcade game involving jumping from the top of one cube to another while remaining in the field of play. Cubit is the result of the corroboration of Jack Carrel and Bill Gronos, names which you will recognize as experienced Assembly Language programmers due to their work here at the International Users-Group. The game is written in Assembly Language and as a result there is lots of action and surprises.

Where most video and computer games require the player to move with as much speed and dexterity as possible, this game offers an alternative. When playing Cubit, many times it is wiser to sit where you are and wait for the proper time for your next move. We will let you decide for yourself. The graphics in Cubit are excellent but as you have seen in the past, graphics do not always make the game. Cubit offers to its owners not only action but also pure entertainment.

The main character in the game is called Bit, and game play involves Bit jumping from one cube to another while changing the color of the tops of all of the cubes to a certain color, depending upon the screen level in which you are playing. This is not always as easy as it may seem because while you are trying to accomplish this task, strange and bizarre characters will try to impede or halt your progress. The main enemy, Bit-Buster, is a vicious sort, and if you should jump the wrong way, you will become a victim of the fiery pit, located at the pyramid's base.

The amount of activity on the screen is representative of the quality of a game that can be developed when Assembly Language is used. There are multiple screens provi ed which progressively increase in speed and complexity. For the more experienced player an option will allow you to skip earlier levels and receive bonus points for accepting the challenge of the advanced screens. Cubit also allows for two-player contests like many arcade games where players are allowed to alternate turns to achieve the highest possible point total. One feature unique to Cubit is that several of the program controls have been moved to the joystick for your convenience, such as game resetting and level selection.

We are very pleased with the effort and forethought that has been put into this action-packed game and look forward to additional releases from Artios.

Yesterdays News Information

Yesterdays News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A & Myarc 9640 computers.

TI-99/4A HARDWARE

Black & Silver computer
Modified PEB
WHT SCSI card with SCSI2SD
Myarc DSQD FDC
Myarc 512K Memory Card
Horizon 1.5 meg Ramdisk
TI RS232 card
Corcomp Triple Tech Card
1 360K 5.25 floppy drive
1 360K 3.50 floppy drive
1 720K 5.25 floppy drive
1 720K 3.50 floppy drive
80K Gram Kracker
Samsung Syncmaster 710mp

TI-99/4A SOFTWARE

PagePro 99
PagePro Composer
PagePro FX
PagePro Headline Maker
PagePro Gofer
TI Artist Plus
GIFMania

PC HARDWARE

Compaq Armada 7800 Notebook
Compaq Armadastation
Samsung Syncmaster 710mp

PC SOFTWARE

Dead,Dead,Dead Windows 98se
FileCap
prn2pbns
Irfanview
Adobe Distiller
Adobe Acrobat

Yesterdays News is composed entirely using a TI-99/4A computer system. It consists of 11 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.

Yesterdays News
c/o Sparkdrummer
AtariAge forum
Phoenix, AZ. 85027