

TIPSCUB'S BEST, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, ACTION GAMES, REFLEX AND CONCENTRATION, MANEUVERING GAMES, TWO-PLAYER GAMES, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, KID'S GAMES, MORE GAMES WORD GAMES, VOCABULARY AND READING, ELEMENTARY MATH, MUSICAL EDUCATION, MIDDLE/HIGH SCHOOL MATH, KALEIDOSCOPES AND DISPLAYS

#47

Copyright 1988

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 120 original programs in Basic and Extended Basic, available on cassette or disk, NOW REDUCED TO just \$1.00 EACH!, plus \$1.50 per order for cassette or disk and PP&M. Minimum order of \$10.00. Cassette programs will not be available after my present stock of blanks is exhausted. The Handy Dandy series, and Color Programming Tutor, are no longer available on cassette. Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

TIGERCUB'S BEST, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, ACTION GAMES, REFLEX AND CONCENTRATION, MANEUVERING GAMES, TWO-PLAYER GAMES, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, KID'S GAMES, MORE GAMES WORD GAMES, VOCABULARY AND READING, ELEMENTARY MATH, MUSICAL EDUCATION, MIDDLE/HIGH SCHOOL MATH, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS DISKS

These are full disks of 100 or more utility subprograms in MERGE format, which you can merge into your own programs and use, almost like having another hundred CALLs available in Extended Basic. Each is accompanied by printed documentation giving an example of the use of each.

NUTS & BOLTS No. 1 has 100 subprograms, a tutorial on using them, and 5pp. documentation.

NUTS & BOLTS No. 2 has 108 subprograms and 10 pp. of documentation.

NUTS & BOLTS No. 3 has 140 subprograms and 11pp. of documentation.

NOW JUST \$15 EACH, PPD.

TIPSCUB'S BEST, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, ACTION GAMES, REFLEX AND CONCENTRATION, MANEUVERING GAMES, TWO-PLAYER GAMES, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, KID'S GAMES, MORE GAMES WORD GAMES, VOCABULARY AND READING, ELEMENTARY MATH, MUSICAL EDUCATION, MIDDLE/HIGH SCHOOL MATH, KALEIDOSCOPES AND DISPLAYS

These are full disks which contain the programs and routines from the Tips from the Tigercub newsletters, in ready-to-run program format, plus text files of tips and instructions.

TIPS VOL. 1 contains 50 original programs and files from Tips newsletters Nos. 1 thru 14.

TIPS VOL. 2 contains over 60 programs and files from Nos. 15 thru 24.

TIPS VOL. 3 has ANOTHER 62 from Nos. 25 through 32.

TIPS VOL. 4 has 40 more from issues No. 33 through 41.

NOW, JUST \$10 EACH, PPD.

* NOW READY *
TIPS VOL. 5 - Another
49 programs and files
* from issues No. 42 -*
* 50. Also \$10 ppd *

TIGERCUB CARE DISKS
#1, #2, #3 and #4.

Full disks of text files (printer required).

No. 1 contains the Tips newsletters Nos.42 thru #45, etc.

No. 2 and No. 3 have articles mostly on ExBasic programming.

No. 4 contains Tips newsletters Nos. 46-52.

These were prepared for user group newsletter editors but are available to anyone else for \$5 each postpaid.

If you bought my C11 disk, Kid's Games, please check line 100 of the Butterfly and Flowers program and, if necessary, change it to-1000 CALL CLEAR :: CALL SCREEN(4).

If you bought my C12 disk, More Games, and have trouble loading Lost Plane and Andromedan Invasion, please go to line 1000 of the LOAD program and change *TC-18* to *TC-16 and *TC-23* to *TC-23. Or, return the disks to me and I will fix them.

Thanks to Ollie Hebert for this fix to the Gordian Knot in Tips #36. This will keep it from running off the edge and crashing in the automatic mode.

```
270 GOSUB 480 :: R=R-24*
(R<1)+24*(R>24):: C=C-28
*(C<3)+28*(C>30):: CH=12
8-(D=1)-(D=3):: CALL GCH
AR(R,C,6):: IF G<>32 THE
N IF INT(2*RND+1)<>1 THE
N CH=6
```

The trouble with me is that, before I finish one program I've thought of another that I want to try writing - and so I don't take time to test completed programs as well as I should. The Decompactor in Tips #35 was one that should have been tested more thoroughly. I think this version will work. It will break an XBasic program into single-statement lines to make it easier to modify. Then, John Dow's Compactor or a similar program will put it back together.

```
100 !DECOMPACTER V.1.1 b
y Jim Peterson fixed 12/
87
```

```
110 DISPLAY AT(3,1)ERASE
ALL:"TIGERCUB DECOMPAC
TER V.1.1"::" Program m
ust first be -"::"RESeq
uenced to greater in-"::"
crements than the number
"
```

```
120 DISPLAY AT(9,1):"of
statements in any one"::"
line."::"SAVED by"::" S
AVE DSK(filename),MERGE"
130 DISPLAY AT(16,1):"IN
PUT FILENAME?"::"DSK" ::
ACCEPT AT(17,4):IF#
140 DISPLAY AT(16,1)ERAS
E ALL:"OUTPUT FILENAME?"
:"DSK" :: ACCEPT AT(17,4
):UF#
```

```
150 OPEN #1:"DSK"&IF$,IN
PUT ,VARIABLE 163 :: OPC
N #2:"DSK"&OF$,OUTPUT,VA
RIABLE 163
```

```
160 LINPUT #1:M$ :: LN=A
SC(SEG$(M$,1,1))*256+ASC
(SEG$(M$,2,1)):: IF LN>L
N2 THEN 180
```

```
170 DISPLAY AT(12,1)ERAS
E ALL BEEP:"ERROR! RESEQ
UENCE PROGRAM TO"::"GREAT
ER INCREMENTS AND TRY"::"
AGAIN." :: CLOSE #1 :: C
LOSE #2 :: STOP
```

```
180 LN2=LN
190 P=POS(M$,CHR$(130),3
):: IF P=0 THEN PRINT #2
:M$ :: GOTO 260
```

```
200 A$=SEG$(M$,1,P-1)::
R=POS(A$,CHR$(132),3)::
S=POS(A$,CHR$(201),3)
```

```
210 IF R=0 THEN PRINT #2
:A$&CHR$(0):: GOTO 250
```

```
220 IF S=0 AND R<>0 THEN
PRINT #2:M$ :: GOTO 260
```

```
230 IF S<>0 THEN IF S-R<
3 THEN PRINT #2:A$&CHR$(
0):: GOTO 250
```

```
240 PRINT #2:M$ :: GOTO
260
```

```
250 LN=LN+1 :: LN2=LN ::
GOSUB 270 :: M$=LN$&SEG
$(M$,P+1,255):: GOTO 190
```

```
260 IF EOF(1)<>1 THEN 16
0 ELSE CLOSE #1 :: CLOSE
#2 :: DISPLAY AT(12,1)E
RASE ALL:"Enter NEW"::"
Then Enter"::" MERGE DSK
"&OF$ :: END
```

```
270 LN$=CHR$(INT(LN/256)
)&CHR$(LN-256*INT(LN/256
)):: RETURN
```

If you have my BXB routine from Tips #40 (corrected in Tips #42) or from my TIPS disk Vol. 4 or NUTS & BOLTS #3, or Genial Traveller Vol. 1 No. 6, here is a neat improvement that Barry Traver thought of. Key this in, run it to create a merge file on a disk. Then clear memory with NEW, merge in BXB, MERGE DSK1.LINEZERO, and now save BXB again in merge format and it will CALL itself from line

zero (and do something else that I'm not going to tell you about!

```
100 OPEN #1:"DSK1.LINEZE
RO",VARIABLE 163,OUTPUT
110 M$=CHR$(0)&CHR$(0)&C
HR$(157)&CHR$(200)&CHR$(
3)&"BxB"&CHR$(130)&CHR$(
157)&CHR$(200)&CHR$(4)&"
CHAR"&CHR$(183)&CHR$(200
)&CHR$(2)&"30"
```

```
120 M$=M$&CHR$(177)&CHR#
(199)&CHR$(16)&"B1C37EA5
B199663C"&CHR$(182)&CHR#
(0):: PRINT #1:M$ :: PRI
NT #1:CHR$(255)&CHR$(255
)
```

And if you have merged in BXB, the edge character (ASCII 31) can be reidentified and colored (set 0) to give the screen an ornamental border.

```
100 CALL CHAR(31,"0")::
CALL CLEAR :: FOR J=1 TO
24 :: PRINT :: NEXT J :
: CALL CHAR(31,"18244299
99422418"):: CALL COLOR(
0,5,16)
```

Here is an improved version of the CATWRITER program to create the Tigercub QUICKLOADER, which is intended for disks of programs which you have filled and do not plan to change. It will read the directory, display each filename, and ask you for the complete program name of each one. Then it prepares a program which displays one or more menu screens of complete program names, and auto-loads whichever one you select.

First, key in this part and save it to disk by SAVE DSK1.CAT1, MERGE. If you want, you can change the screen and character colors in line 10. Don't change the line numbers!

```

10 CALL CLEAR :: DIM M$(
127):: CALL SCREEN(5)::
FOR S=0 TO 14 :: CALL CO
LOR(S,16,1):: NEXT S ::
CALL PEEK(8198,A):: IF A
<>170 THEN CALL INIT
11 REM (leave this in!)
12 ON WARNING NEXT :: GO
SUB 21
13 X=X+1 :: READ M$(X)::
IF M$(X)<>"END" THEN 13
14 R=3 :: FOR J=1 TO X-1
:: READ X$ :: DISPLAY A
T(R,1):STR$(J);TAB(4);X$
:: R=R+1 :: IF R<23 THE
N 17
15 DISPLAY AT(24,1):"Cho
ice? or 0 to continue 0"
:: ACCEPT AT(24,26)VALI
DATE(DIGIT)SIZE(-3):N ::
IF N>X-1 THEN 15
16 IF N<>0 THEN 19 :: R=
3
17 NEXT J
18 DISPLAY AT(24,1):"Cho
ice?" :: ACCEPT AT(24,9)
VALIDATE(DIGIT):N :: IF
N=0 OR N>X-1 THEN 18
19 CALL CHARSET :: CALL
CLEAR :: CALL SCREEN(8):
: CALL PEEK(-31952,A,B):
: CALL PEEK(A*256+B-6553
4,A,B):: C=A*256+B-65534
:: A$="DSK1."&M$(N):: C
ALL LOAD(C,LEN(A$))
20 FOR J=1 TO LEN(A$)::
CALL LOAD(C+J,ASC(SEG$(A
$,J,1))): NEXT J :: CAL
L LOAD(C+J,0):: GOTO 100
00
21 CALL LOAD(8196,63,248
)
22 CALL LOAD(16376,67,85
,82,83,79,82,48,8)
23 CALL LOAD(12288,129,1
95,126,165,129,153,102,6
0)
24 CALL LOAD(12296,2,0,3
,240,2,1,48,0,2,2,0,8,4,
32,32,36,4,91)

```

```

25 CALL LINK("CURSOR")::
RETURN
10000 RUN "DSK1.12345678
90"

```

Next, key in this little routine and run it to create a file called CAT2.

```

100 OPEN #1:"DSK1.CAT1",
VARIABLE 163,INPUT
110 OPEN #2:"DSK1.CAT2",
VARIABLE 163,OUTPUT
120 FOR J=10 TO 26 :: LI
NPUT #1:M$ :: PRINT #2:C
HR$(0)&CHR$(J)&CHR$(156)
&CHR$(253)&CHR$(200)&CHR
$(1)&"2"&CHR$(181)&CHR$(
199)&CHR$(LEN(M$))&M$&C
H R$(0):: NEXT J
130 PRINT #2:CHR$(255)&C
HR$(255):: CLOSE #1 :: C
LOSE #2

```

Finally, key in CATMATRIX. Leave the line numbers as they are. we need that space after line 9.

Then MERGE in DSK1.CAT2 to combine the two, and SAVE.

```

1 CALL CLEAR :: CALL TIT
LE(16,"CATWRITER"):: CAL
L CHAR(124,"3C4299A1A199
423C"):: DISPLAY AT(2,10
):"Version 1.3":;TAB(8)
;"! Tigercub Software"
2 DISPLAY AT(15,1):"For
free":"distribution":"bu
t no price or":"copying
fee":"to be charged." ::
FOR D=1 TO 500 :: NEXT
D :: CALL DELSPRITE(ALL)
3 DISPLAY AT(2,3)ERASE A
LL:"TIGERCUB CATWRITER V
.1.3":;:" Will read a di
sk directory,":"request
an actual program":"name
for each program type"
4 DISPLAY AT(7,1):"filen
ame, and create a merg-
":"able Quickloader which
dis-":"plays full progr
am names and":"runs a se
lected program."

```

```

5 DISPLAY AT(12,1):" Pla
ce disk to be cataloged"
:"in drive 1 and press a
ny key" :: CALL KEY(0,K,
S):: IF S=0 THEN 5
9 OPEN #2:"DSK1.CATMERGE
",VARIABLE 163,OUTPUT
100 OPEN #1:"DSK1.",INPU
T,RELATIVE,INTERNAL ::
INPUT #1:N$,A,J,K :: LN=
1000 :: FN=1100
110 DISPLAY AT(12,1):"Di
sk name?":;N$ :: ACCEPT
AT(14,1)SIZE(-28):N$ ::
LX$=STR$(14-LEN(N$)/2):
: LXLEN=LEN(LX$)
120 PR$=CHR$(0)&CHR$(11)
&CHR$(162)&CHR$(240)&CHR
$(183)&CHR$(200)&CHR$(1)
&"1"&CHR$(179)&CHR$(200)
&CHR$(LXLEN)&LX$
130 PR$=PR$&CHR$(182)&C
H R$(181)&CHR$(199)&CHR$(L
EN(N$))&N$&CHR$(0):: PKI
NT #2:PR$
140 X=X+1 :: INPUT #1:P$
,A,J,B :: IF LEN(P$)=0 T
HEN 180 :: IF ABS(A)=5 O
R ABS(A)=4 AND B=254 THE
N 150 ELSE X=X-1 :: GOTO
140
150 DISPLAY AT(12,1):P$;
" PROGRAM NAME?" :: A
CCEPT AT(14,1)SIZE(25):F
$
160 PRINT #2:CHR$(INT(FN
/256))&CHR$(FN-256*INT(F
N/256))&CHR$(147)&CHR$(2
00)&CHR$(LEN(F$))&F$&C
H R$(0):: FN=FN+1
170 M$=M$&CHR$(200)&C
H R$(LEN(F$))&P$&CHR$(179)::
IF X<11 THEN 140
180 IF M$="" THEN 200
190 PRINT #2:CHR$(INT(LN
/256))&CHR$(LN-256*INT(L
N/256))&CHR$(147)&SEG$(M
$,1,LEN(M$)-1)&CHR$(0)::
LN=LN+1 :: M$="" :: X=0
:: IF LEN(P$)<>0 THEN 1
40
200 PRINT #2:CHR$(INT(LN
/256))&CHR$(LN-256*INT(L
N/256))&CHR$(147)&CHR$(2
00)&CHR$(3)&"END"&CHR$(0
)
210 PRINT #2:CHR$(255)&C
HR$(255):: CLOSE #1 :: C
LOSE #2

```

```

220 DISPLAY AT(8,1)ERASE
  ALL:"Enter -":::" NEW
":::" MERGE DSK1.CATMER
GE":::" DELETE ""DSK1.C
ATMERGE"":::" SAVE DSK
1.LOAD"
230 SUB TITLE(S,T$)
240 CALL SCREEN(S):: L=L
EN(T$):: CALL MAGNIFY(2)
250 FOR J=1 TO L :: CALL
  SFRITE(#J,ASC(SEG$(T$,J
,1)),J+1-(J+1=S)+(J+1=S+
13)+(J>14)*13,J*(170/L),
10+J*(200/L)):: NEXT J
260 SUBEND

```

Mike Stanfill and Ed Machonis and others have been publishing some neat little single-screen "tinygram" programs, so here is my contribution. It's a one-screen one-liner!

```

1 RANDOMIZE :: PRINT : :
: : : :: A=INT(RND*7)::
B=INT(RND*9+1):: FOR X=
1 TO 5 :: Y=A*X^2-B*X+B
:: PRINT Y::: NEXT X ::
Y=A*X^2-B*X+B :: PRINT :
: :: INPUT "GUESS NEXT
NUMBER":N :: IF N=Y THEN
PRINT "RIGHT" :: GOTO 1
ELSE PRINT "CORRECT IS
":Y :: GOTO 1

```

MEMORY FULL!
- Jim Peterson

DM-1000 BIT AND SECTOR COPIER
By Mel Samouri
Taken From Pamona Valley NL

BITMAP - Only the sectors mapped in the disk directory as being used are copied. This means that fractured files will be un-fractured on the copy disk and only those sectors are copied. This is best as it does not copy unused sectors so it is a little quicker.

SECTOR - This allows you to clone a disk, (Some software companies use re-numbered sectors as protection or purposely fracture files), or part of disk, (blown directory), and if the defaults are used, will make an exact duplicate of the disk. If you are not fairly familiar with how the disk is set up, use the default. If you do not, you could get only part of your programs copied, as it does not use the sector map. (For double density an extended bit map is used, they are not all in sectors 0 to 20, if you have a lot of files on the disk.

You should, if possible perform what is called, "OPTIMISING", on file disks that you use frequently or delete programs or add programs to, as the files become fractured, (part here, part there, etc.), and this can cause added loading times, worn drives and even, (worst case), the dreaded data loss. You can tell this is happening if you hear the drive head, "see-sawing", back and forth when you load your programs.

I forgot to add that, "OPTIMIZING", is performed by using the file copy, (option 1 of DM-1000), onto a clean disk. Use the Sweep Disk option on the original disk, (NOT THE COPY DISK), and then copy the files from the copy disk onto the original disk and presto, no fractured files. (Ed. note: Be SURE you have copied ALL the files from the original to the copy disk BEFORE you sweep the original).

THREE-IN-ONE DISKETTE
by Steven DeGeare

COMPUTER WAR

Here is a review on a disk that contains 3 popular games for the TI. RIVER RESCUE - COMPUTER WAR - and SUBMARINE COMMANDER. They are copy-righted commercial games released from THORN EMI VIDEO

SUBMARINE COMMANDER

Your mission is to attack and destroy the enemy convoy, which contains the following ships: TANKER, FREIGHTER, and DESTROYER that can and does drop depth charges on you once located.

There are 3 different screens. A: map screen, which shows you in the Metterian Sea and any ships which are close to you. B: the sonar screen, is used for detecting ships while underneath the water. C: the periscope screen. It shows you what is approaching you. If at the height of least then 40 feet. It also allows you to aim your torpedoes and see the destruction of the ship.

This program contains many gauges which guide you, the commander, in search of the enemy vessels. A fun game, but time consuming and frustrating to win.

RIVER RESCUE

It is JS operated game, where you guide the boat down a river avoiding logs, electric eels, etc. Your mission to rescue people on the North shore by docking whenever possible. This gives points. Then drop them off on the other side. You have 5 boats to use. The JS controls fire power, movement of the boat, and speed.

One thing to watch for, is the enemy tank which must be outran. I have not been very far in this one, but do like it best of all 3.

In this game you see a map of the US with blips, representing incoming missiles. You move your JS towards them and press the fire button. If in range, the screen will change to the cockpit of a fighter. Thus giving you a chance to destroy the missile. The length of time depends on where you locate it.

On destruction of the missile the first screen comes back on with a bonus screen. By moving the 3X3 square around till you find a copy or the time runs out. I have yet to understand this part of the game. Perhaps someone else can explain better. The game is over when the DEFCON factor reaches 1.

For those who are into graphic games. This disk will probably be worth the cost.

~~~~~

ALPHA LOCK  
-----

This was in the Micropendium and several newsletters so I guess it's OK to proceed without checking it out as I usually do before I pass something on. I have a spare keyboard and used that to check the following.

As everyone knows the joysticks won't work when the alpha lock is down. By soldering in a simple diode you can forget the necessity of letting the key up.

You can purchase a package of diodes, (a 1N914/1N4148, #276-1122, 10 for 99c), from your local Radio Shack store. Remove the bottom of your console and locate the alpha lock switch. It will be under the key board on your right, in the lower right corner, with the console upside down and the front facing you.

The switch should have a single foil trace leading away from it. It is the very last solder point in the lower right corner. Cut the trace and leave a 1/32" gap in it. Take the diode and solder it across the gap

with the cathode band, (small band on end of diode), away from the switch. Diodes have to be polarized so make sure of the band position.

On some key boards the trace from the switch cannot be seen, but you can put the diode where the ribbon cable connects to the key board. It is located at the top edge of the PC board, off center to the right. From the right of the connector count to the left to the sixth trace and make a gap there and solder the diode across the gap with the band toward the connector.

The Alpha Lock key should now work as normal, but no need to release it to use the joysticks.

~~~~~

Tingo Bingo Game
by Steven Karasek
given personally
to KC99ER president

Tingo (or TI-Bingo) will print Bingo cards and call the game for you. It starts out by asking how many cards it should print (you can reuse the same cards when you play later, or you can use your own cards if you have them). It will print them 4 to a page. These should be cut and glued or taped to cardboard to make them sturdier. It takes a while to print the cards, so get it started early, before you're actually ready to play. Pennies or buttons can be used as markers.

When everyone has their cards and markers, press the ENTER key as instructed to start the game. The computer will display the letters and numbers on the screen in large letters and also say them if you have a speech synthesizer attached. It will also display on the screen all of the numbers called so far. When someone has a TINGO or if you want to pause it to check the numbers, press any key. You will be instructed to press C to continue or N for a new game (make sure the ALPHA LOCK key is

depressed). If there was a TINGO and you're ready for a new game, press N, otherwise press C to continue the current game.

If you want to vary the speed at which the numbers are called, change the 150 in line 410 to a larger number for slower speed or a smaller number for faster speed.

If your printer is not named "PIO", change the name in line 140. 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 10 lines per inch.

```
100 L$=RPT$("-",80):: M$=": "&
RPT$("!" " ,4)&"!": M$=M$&M$
110 CALL MAGNIFY(2):: RANDOMIZE ::
DIM U(75,1),T$(7),U$(15):: FOR I=0
TO 9 :: READ P(I):: NEXT I
120 DATA 31599,18724,29671,31207,18
925,31183,31695,4775,31727,31215
130 DISPLAY ERASE ALL AT(8,12):"TIn
go" :: DISPLAY AT(10,7):"By Steve K
arasek"
140 INPUT "HOW MANY CARDS TO PRINT?
":N :: IF N THEN OPEN #1:"RS232/2.
LF",OUTPUT,VARIABLE 255 :: PRINT #1
:
150 FOR I=1 TO (N+1)/2 :: PRINT #1
:: FOR J=1 TO 10 :: PRINT #1:TAB(J*
8-4),SEG$("TINGOTINGO",J,1):: NEXT
J :: PRINT #1
160 FOR H=1 TO 5 :: PRINT #1:L$:M$
:: FOR M=0 TO 1 :: FOR J=0 TO 4
170 K=INT(RND*15)+1+J*15 :: IF U(K,
M)THEN 170
180 C(J,M)=K :: U(K,M)=1 :: NEXT J
:: NEXT M :: FOR K=0 TO 4 :: FOR M=
0 TO 1 :: FOR J=0 TO 4 :: N$=N$&"!"
:: IF H<>3 OR J<>2 THEN 210
190 IF K=2 THEN N$=N$&"*FREE*" ELS
E N$=N$&" "
200 GOTO 250
210 X=0 :: FOR W=1 TO 0 STEP -1 ::
X=INT(C(J,M)/10^W)-X*10
220 FOR L=0-(J=0 AND W=1)TO 2 :: IF
(P(X)AND 2^(L+K*3))>0 AND(W=0 OR X
>0)THEN N$=N$&"#" ELSE N$=N$&" "
230 NEXT L :: IF W THEN N$=N$&" "
240 NEXT W
```



```

250 NEXT J :: N$=N$&"!" :: NEXT M :
: FOR M=1 TO LEN(N$):: IF SEG$(N$,M
,1)="#" THEN PRINT #1:"@";ELSE PRIN
T #1:" ";
260 NEXT M :: PRINT #1:CHR$(13);N$
:: N$="" :: NEXT K :: PRINT #1:M$ :
: NEXT H :: PRINT #1:L$
270 IF INT(I/2)*2=I THEN PRINT #1:C
HR$(12);
280 GOSUB 470 :: NEXT I :: IF N THE
N CLOSE #1
290 FOR I=2 TO 7 :: READ T$(I):: NE
XT I :: FOR I=1 TO 15 :: READ U$(I)
:: NEXT I
300 DATA TWENTY,THIRTY,FORTY,FIFTY,
SIXTY,SEVENTY
310 DATA ONE,TWO,THREE,FOUR,FIVE,SI
X,SEVEN,EIGHT,NINE,TEN,ELEVEN,TWELV
E,THIRTEEN,FOURTEEN,FIFTEEN
320 CALL CLEAR :: FOR I=9 TO 14 ::
READ J :: CALL COLOR(I,J,J):: NEXT
I
330 DATA 6,7,13,5,14,3
340 Z=0 :: CALL DELSPRITE(ALL):: IN
PUT "PRESS <ENTER> WHEN READY ":X$
: DISPLAY ERASE ALL AT(1,5):"T
I N G O"
350 J=4 :: FOR I=96 TO 136 STEP 8 :
: CALL VCHAR(1,J,I,17):: J=J+5 :: N
EXT I
360 IF Z=75 THEN 460
370 J=INT(RND*75):: IF U(J,0)THEN 3
70
380 Z=Z+1 :: U(J,0)=1 :: I=INT(J/15
):: J=J+1 :: DISPLAY AT(J-I*15+2,4+
I*5)SIZE(2):USING ("##"):J :: X$=SE
G$("TINGO",I+1,1)
390 CALL SPRITE(#4,ASC(X$),2,144,10
4):: Y$=STR$(J)&" " :: FOR I=1 TO L
EN(Y$):: CALL SPRITE(#I,ASC(SEG$(Y$
,I,1)),2,144,114+I*14):: NEXT I
400 CALL SAY(X$):: IF J>15 AND J<20
THEN CALL SAY(U$(J-10),,"TEEN")ELS
E X=INT(J/10)*-(J>19):: CALL SAY(T$
(X),,U$(J-X*10))
410 FOR I=1 TO 150 :: CALL KEY(0,X,
S):: IF S THEN 430
420 NEXT I
430 IF S=0 THEN 360 ELSE DISPLAY AT
(22,1):"PRESS C TO CONTINUE OR N FO
RA NEW GAME"
440 CALL KEY(0,X,S):: IF X=-1 THEN
440
450 X$=CHR$(X):: IF X$="C" THEN CAL
L HCHAR(22,1,32,64):: GOTO 360 ELSE
IF X$<>"N" THEN 440
460 GOSUB 470 :: GOTO 340
470 FOR J=0 TO 75 :: U(J,0),U(J,1)=
0 :: NEXT J :: RETURN

```

PUBLIC DOMAIN GAMES

```

*****
*HXYEYXRTPEMBJFHDISBVBZCPJ*
*PYDPOVYUNYRIIXOZRNVZELCCPE*
*QENWYVLDVVCWCUQWVJTVIVLOKLS*
*EBFFZMCPZEJBWTAIPAFQTQLOZU*
*UXZBHNLDYVVKFCJZYVMVKCYWUS*
*PBIOTVTEBVKHOAVCBSEGCQKOD*
*YQNMZXWLIRXSHLLCRIXSMAHI*
*NFWBZWBGAALUBXQYTVNZNPSIE*
*VDSEOZTHMPRUFVVKRHMLOHCTFD*
*CKGRWETQDTGDRGAEGTSREKGUHF*
*DFSJATFDFTZJRAPZMZRBCLUMIO*
*YFSLOFBKOBKJUTIKNLCJWWKYFR*
*JLCUKZNOEZRWSCNBIFBRYYHIXY*
*SWRQCABKURKEEIJYYPCADRIED*
*YRZUBPDHRNFXFGHCZPQWBPBJKU*
*OFQLJUMQZWCWFVMNBZKKJUCXOR*
*OGNIBITFBYEVCM SVRWPZVMPNS*
*UIEATDLJJHYEVADNTYGVJUNMYGI*
*ZWAVRTLQJZXJSPDIKKDJFWHLNS*
*RBIMRGGDATUSNTUONMXJSRVFAS*
*AEPOPGIENEJQKVTJJDKJXZXTWS*
*SPFAWZPOJCATAPULHTRANEAPP*
*SJFCCIYXFNFVUYPTUHBUDWWVZ*
*WILCWNUFHJGGXTKMFZLCARSOO*
*KRWSHIPWRECKZRLWAGYXMQENKO*
*****

```

WORD LIST

```

-----
BOMBER          SWATFLY
BOUNCY          TIBINGO
XATAPULT
KNIGHTTOUR
KONG
MARKET
SHIPWRECK
SUPERTREK

```

KANSAS CITY TI99/4A COMPUTER USERS' GROUP
 POST OFFICE BOX 12591
 NORTH KANSAS CITY, MISSOURI 64116

FIRST CLASS MAIL

The K.C. TI99/4A Users' Group is in no way affiliated with or sponsored by TI. Mention of a company or product is not considered an endorsement. Opinions expressed are those of the writers and not necessarily those of the Group, its officers, editor and the membership. Information from this Newsletter may be reproduced only if credit is given to both the author and this newsletter.

 | NEWSLETTER ONLY MEMBERSHIP \$12/yr |
 | USER GROUP MEMBERSHIP IS \$20/vr |

BLOODBANK -- JUNE 1989 -- Walter H. Blood -- Kansas City, Kansas

```

=====
BYTE      " ~~
BACK      "(c o
by        " >
RICHARD   " / \ " ( )
ORLIN     " ( : " < )
in        " (-| lo " < )
          " '-| lo " ::'
SYSTEMS/3X " : o' : ( ) : \ 'o /
& AS WORLD "#|#!| o' : ( ) : \o/
12/88     "#|#!| " : : :
          "#|~\ " : " : :
          "~\ ) " : " : :
          "L\ ) " : ^ : :
          "----- ( ) ( ) : :
  
```

Get help quick! There is a DP manager buried under all the computer magazines.