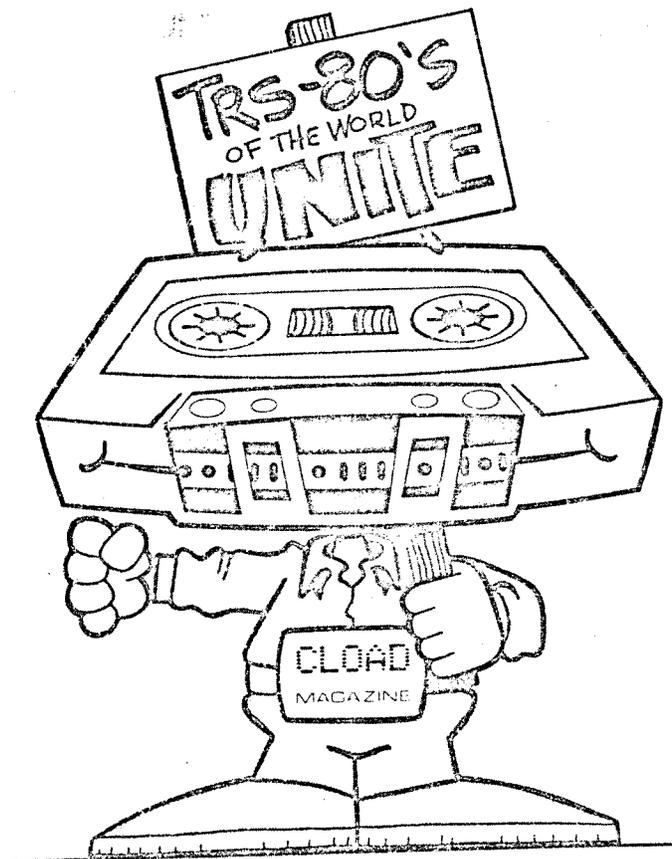


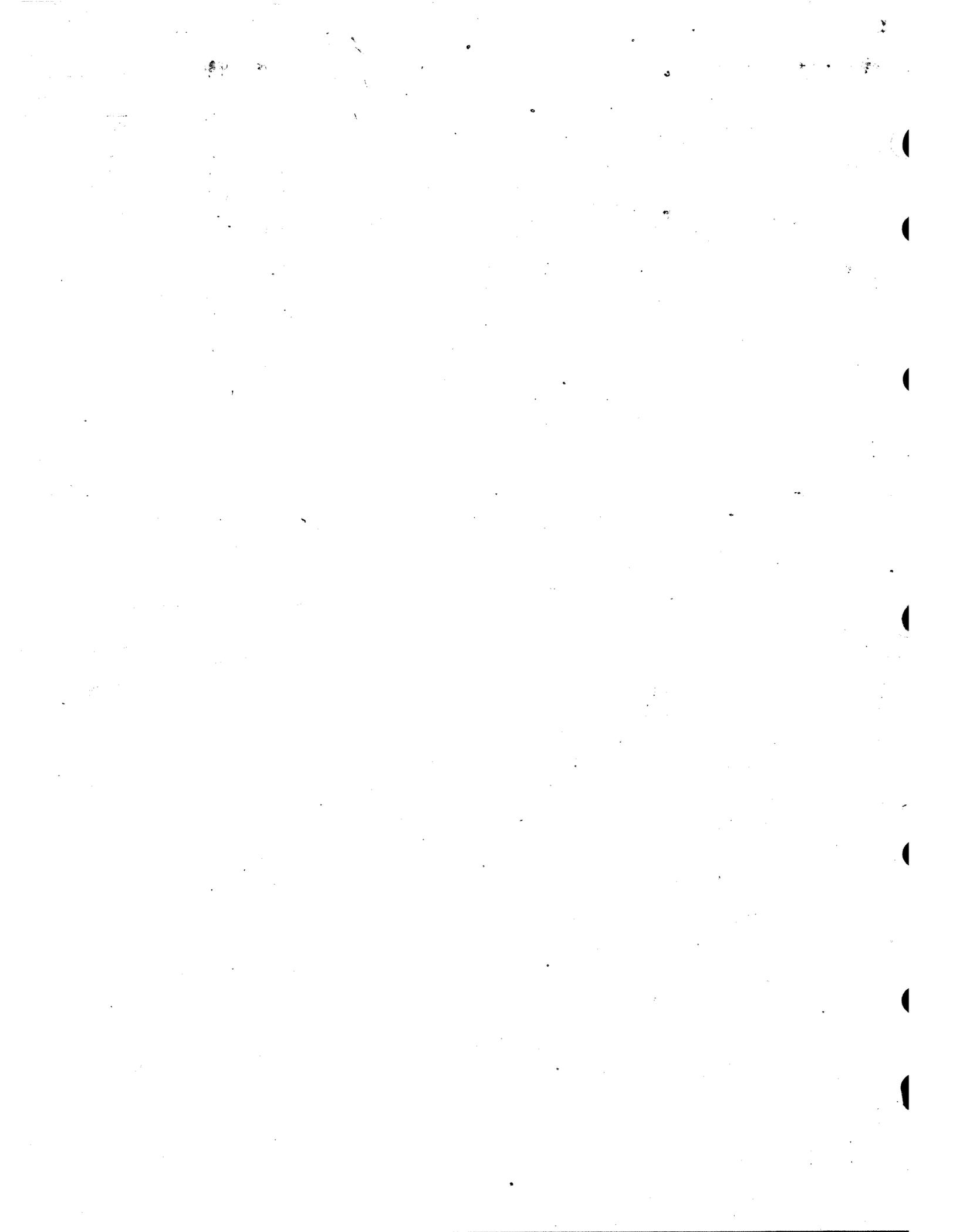
THE
BEST OF CLOAD
VOLUME ONE

or,
SIX MONTHS OF TRYING
[desperately]

(and failing) to meet our
publishing deadline of the
first of each month.

This compendium published
at Goleta, California, one
year behind schedule.





The CLOAD Saga (our first six months)
by Ralph McElroy, Publisher

In the beginning, there were no personal computers. For present purposes, the beginning can be assumed to be reasonably close to the winter and spring of 1977. At the time, there was a small company (1 1/2 people) eking out a nominal to meager existence in a very old building on the Santa Barbara Airport. One of these one and one half people was me, and the company was "us". Then, as now, we were involved in the design and production of microprocessor based products for industrial customers, using the Zilog Z-80 chip almost exclusively. The only problem we faced at the time was that nobody wanted any of our products, save an electromechanical doohickey that properly belonged in the Stone Age. Minor problem. As we stumbled along in the middle of a project that was eventually to fail, we came to the realization that one computer (a venerable "old" Altair) was not enough. Fred (Frederick Q. Altair - the computer) was both hardware and software development system, was torn apart for modifications about half the time, and was in use eighteen hours a day seven days a week, those being my working hours. What we needed was a separate computer for text entry and general office work.

Commodore Business Machines had by this time announced a new computer, the PET - a 6502 based machine for \$500. Now I'm no 6502 fan, but at the time, this machine was all that was available ("available" is a loose word in the computer industry), so we decided to get one. The check had been written and almost mailed when we got the word that Tandy Corporation / Radio Shack had announced a Z-80 based system, the TRS-80. I hadn't quite forgiven Radio Shack for buying out Allied Radio some ten years back (I don't hold grudges long), but I could relate to that Z-80 in there. I had many months of machine level software which would run on the beast.

You see, the microprocessor field was marked at the time by a marked lack of software. The only software around was difficult to get without stealing, and theft was rampant, which discouraged software distribution, which made it difficult to get without stealing, and so on. The alternative was to buy a bunch of MITS 4K dynamic memory boards (16K of it cost \$1,600) which were known to be inoperative by design, but which "qualified" you to purchase MicroSoft BASIC (with these computers, even BASIC had to be loaded). The problem boiled down to this: If you were poor - but - honest, what you had was what you wrote, and its value was not in dollars, but in hours, which are harder to make.

Thus the trip to the local Radio Shack store to put a down payment on one of the beasts. People who put down payments on products known to be in development are a strange subset of humanity. Their mentality is inversely proportional to their knowlege. After receiving assurances that "it couldn't be more than two weeks or so" we went back to the shop and continued our hammer - and - bang routine for a few months. I was thoroughly amazed when the thing came in in a mere four months (those of you who think that is an excessive wait have not been involved in a microprocessor development project).

The wait had, however, given me time to cogitate on the opportunities involved with having a googol of these things around. For one thing, the rampant software theft in the hobbyist market had subsided, and prices and availability eased considerably. Software development had progressed from certain economic suicide to mere economic foolish-

ness, and was promising to be profitable, though not lucrative, in the near future. The companies who were in prime position to pick up the chips were those companies who were in on the first deal and weathered the cheating.

"Of course the game is rigged - you can't win if you don't bet".
(Robert Heinlen)

Here, as I saw it, would be the future - told (conveniently) in retrospect:

1) The market for hardware, add-on peripherals, and modifications would hit as soon as the computers were delivered. They would hit the wrong market. TRS-80 buyers would not be hardware diddlers. And anyone who succeeded in developing a widely desired hardware item would get creamed when Radio Shack brought out their version.

2) The market for software would be explosive - every fool who ever read Dave Ahl's "101 Computer Games" (a very good book, by the way) would be out there offering Yet Another Version of Biorythm, or Lunar Lander.

3) Those who had good software would see it ripped off and sold by others who were aware that software copyright laws are totally unenforcable in any form.

4) The software market would begin to saturate, and the folks who had good stuff would have a difficult time marketing it. Fortunately, so would the ripoff artists. Most software would be sold by distributors, and brand name association would be the dominant marketing factor.

5) At this point, a half dozen companies or so would have built a reputation for honesty and decent software and would be in a good market position to continue operation.

O.K, I'm interested - how does one go from zero to a strong marketing position? The logical thing was to be in a situation where one is delivering software in a continuing stream, and to ship it to a list of people on a subscription basis. This allows a more vocal feedback path - someone who has signed up for twelve issues will let you know what that first issue looks like. There is also a source of software - the readers would submit material of general interest much as in a regular magazine.

Important points:

1) At first, sell subscriptions on an open credit basis at cost. This builds up the list (we only lost about \$5,000 to bad debts).

2) Don't get ulcers over piracy - either it will survive or you will (of course the game is rigged...).

3) Pay for submissions. Cash in full before publication - even if it's not much, it's better than vague promises and the bookkeeping is a lot easier.

The first six months went according to plan, though the market was a lot smaller than we thought (we didn't break through to 1000 subscribers for five months) and it took awhile before the submissions started coming in. Quite a bit of what follows was written in-house to fill out the early issues.



FLAGS
by Neal P. Jensen

"C"

Flags is a fantastic program which draws several versions of the American flag. The inspiration was from a series of stamps put out by the U.S. Post Office to commemorate the Bicentennial Year. It is set up as one of our "front cover" programs - a logical place to put programs which utilize the TRS-80's graphic capabilities as an art medium. (Besides, all magazines should have a front cover...).



YIELD TO MATURITY
by Charles W. Evans

"Y"

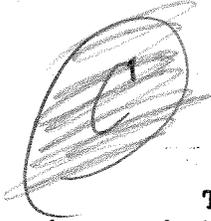
Yield to maturity is a financial program, set up to calculate the yield of a bond, such as the kind which finances the construction of the new municipal swimming pool.

These bonds are government promises to pay a certain sum (invariably \$100) at a specified future date (maturity). They are bought and sold freely on the open market (check your local paper for "bonds" in the financial section). The price at which they are sold is invariably lower than this \$100 face value by some amount (the discount), or nobody would buy them. The size of this discount is usually dependent on the current inflation rate, the taxable status of the bond, the phase of the moon, etc. Example: a bond currently is for sale at \$90 and will mature in exactly one year (at that time the bond maker buys it back for \$100). The yield to maturity is the \$10 gain in value divided by the \$90 investment for the time period, or 11.1% per year. Not enough? Don't buy it. If enough people don't buy, it will start selling at \$89 (note that if inflation / interest rates go up, the bond values go down).

One complication. Sometimes a bond maker will create bonds at a time when interest rates are exceptionally high. It is common to "hedge the bet" by writing a "call" into the bond. This is a clause on the bond which says that on a certain date, the maker of the bond can buy it back at a certain price. This affects the current value of the bond in that it will never sell at higher than this amount before this date is passed, and a bond buyer looks at the yield to call as well as the yield to maturity. The lesser of the two is the effective one.

This program sorts out all the complications of time and amounts to give a correct value of yield to call and maturity. We have verified that it does indeed work to sufficient accuracy to base business decisions on, although there obviously can be no warranty that it is absolutely bug free.

The code is as good as you'll ever see in BASIC. If you're looking for a programming style to emulate, this is it.



(18) 5 17

SANDCASTLE by Yours Truly

This was written by a drunken sot who was employed at the time by a shaky magazine with no future. The object of the game was to make money for us by providing filler material for the first issue of CLOAD magazine, there being no software to speak of. The game demonstrates the engineering concept of slump, or how steeply you can pile material. It turns out that any given pile of material (no comments from the Peanut Gallery) can be stacked in a pile anywhere from level to a certain critical value of steepness (the slump ratio, height to width). If a steeper pile is attempted, the sides will shift and settle until this ratio is achieved (which is why children should never be allowed to play on a commercial sand storage pile - they might get buried in the shifting and settling).

You start out with a supply of three different kinds of material (sand, gravel and brick). These materials all have different slump ratios. Specifically, sand must be stacked 6 or more meters narrower than the previous layer, gravel 4 or more meters, and brick 2 or more meters. Example: if the base layer is 14 meters wide, the layer placed on top must be 1 through 8 meters if you choose to lay sand. If a layer of sand 9 meters wide is attempted, it will slump and you will have to start all over. Gravel is better stuff - if the base layer is 14 meters wide, a layer of gravel 1 through 10 meters wide can be laid on top (an 11 meter wide layer will slump). Brick is best of all - if the base layer is 14 meters wide, a layer of brick can be laid up to 12 meters wide. It's a shame that you only get a little bit of gravel and brick. As you build your pile up, the volume of material used is deducted from your supply. For planning purposes, each layer is 4 meters thick and is circular in cross - section, and the material volume is therefore $(3.141) * (4 \text{ meters}) * (\text{one half width of layer in meters, squared})$. Armed with this information, you can work out the problem analytically (hint: work it backwards), but you can also get it by trial and error. You get the fireworks only if you succeed in building 18 layers.



"K"

KNIGHT'S TOUR by Paul Kafoure

This is an implementation of an old mental teaser where a chess piece (the knight) is moved around a chessboard such that every square is visited once and only once. Only legal moves are allowed (one square in any direction, two squares in any perpendicular direction). This is an example where the computer is the referee and scorekeeper - it does not actively participate in the game, though in this example it serves a useful purpose.

This is also an example of "tight code". The author chose to write in a terse style, allowing a complex sequence of tasks to be handled by a 4K level I system, the only one available at the time.



"P"

POOL
by Richard Fuller

This is another program written in those beginning months where every little bit helped. Unlike "Sandcastle", however, this program was written for a legitimate reason - to improve the author's skill at bank shots! It works on the same principle as the pool table (surprisingly enough) which is simply stated: the angle of incidence is equal to the angle of reflection. The way pool sharks operate is to set up an aiming point based on the geometry of the shot, and knock the cue ball towards that point. It is valuable, therefore, to be able to predict ("call") a shot based on an imagined aiming point. In this program, the aiming point is set by the player, and then modified as necessary if it "looks wrong".

Computers have applications limited only by imagination...
Advice: don't play pool with Dick Fuller.



"L"

Y=mX+b
by Richard Fuller

Another program by Dick Fuller, This one performs an operation known as "linear regression".

Often there is a need to enter what a mathematician would call a "functional relationship" in the form of a table of individual points (although true mathematicians flinch at the thought) and use the normally valid assumption that the world is well - behaved to use this set of points to predict the location of other points not listed. In the example given in the instructions, the table of points is the calibration data on a speedometer. Given this table of data, the program can predict either the true speed when given the speedometer reading, or predict the correct speedometer reading for a given true speed.



"B"

BREAK
by Yours Truly

This is the first program I wrote after receiving the TRS-80. It was initially an exercise in mastering the graphics ability of the beast, added to a sick desire to perpetrate a horrible pun. The TRS-80 has a special key ("break") to bail the user out of an infinite loop (traditionally this was done with a control "C" - shift "C" works on a TRS-80). What better use could one ask of this key than to actually break something? Or for that matter, why not really get absurd and require the user to "run" after breaking it?

Well, here it is, folks - you get to try to judge the instant of impact, with a score which gradually increases until that instant, when it drops to zero. There is one interesting point in the code, that of testing the variable N prior to setting it. This is a "switch" to allow the program to know whether or not this is the first time the program has been RUN. On the initial RUNNING, the program goes through the initializing routine, and thereafter ignores it.

"I" PINBALL
by Paul Kafoure

This game is based on the video game "breakout". Unlike "Break", this game is executed in "real time" - that is, it continually runs while simultaneously gathering data used to control the paddle. The TRS-80 level I machine, however, does not have an "INSTR\$" function. This means that some ingenuity is called for (true programmers have never been short in this department - they're often accused of having too much).

The technique that Paul uses is to take advantage of the fact that the level I TRS-80 automatically "echoes" any character to the screen, and the fact that a "set point" can be tested by the program while executing. The space bar and backspace key are used in this fashion to wipe out two strategically placed set points, and the program senses this by continual testing of these points. This technique can be expanded to yet a third set point, by setting it up such that an ENTER (carriage return) will wipe out the third point, placed on the next print line.

"J" JUKEBOX
by James Garon

This program uses the TRS-80 in about as unconventional a fashion as can be. All digital electronic circuits radiate "hash", or electromagnetic noise which often interferes with nearby radio and TV sets. This noise can be "tuned" by doing various things within a program. As you might guess by looking at the listings, the term "various" includes some strange stuff, and it doesn't convert from level I to level II at all (it was re-written instead).

This program brings back fond memories of "Music of a Sort", an article written by Steve Dompier in the old PCC Newsletter way back in 1976. At that time, the MITS Altair had just hit the scene, and there was no I/O available at all (imagine a TRS-80 with no video, keyboard or cassette and you've got the picture). Steve was toggling along on the front panel switches, "writing" machine level programs to test out the 8080 chip, and got a WHOOP out of a nearby weather radio when he executed a sort routine. He immediately recognized the importance of this and wrote a program which beeped out "Fool on the Hill" and "Daisy", creating what was at that time the only task that a home computer could do - emulate a kazoo.

FLAGS LEVEL I

```

00002  REM AUTHOR NEAL P. JENSEN, 1589 BLOSSOM PK, LAKEWOOD, OH 44107
00003  REM BYTES LEFT 13719/1431
00010  CLS
00020  READX
00030  IFX=100T.100
00040  G.20
00100  READA$,Q
00110  Y=2:I=1
00120  READX
00130  S.(X,Y)
00140  IF(I=14)+(I=19)+(I=23)+(I=27)+(I=34)+(I=38)+(I=42)+(I=47)T.Y=Y+1
00150  I=I+1
00155  IFI=65T.170
00160  G.120
00170  P.AT98,"TM"
00180  P.AT292,"BEST OF CLOAD VOL. 1A"
00190  F.X=0T0127
00200  S.(X,0)
00205  S.(X,15)
00220  N.X
00230  F.Y=0T015
00240  S.(0,Y)
00245  S.(1,Y)
00250  S.(127,Y)
00255  S.(126,Y)
00260  N.Y
00265  R.(2,Y):R.(125,Y)
00270  REST.
00280  G.615
00400  F.Y=21T033S.4
00410  F.X=QT0127
00420  S.(X,Y):S.(X,Y+1)
00430  N.X:N.Y
00450  F.Y=37T045S.4
00460  F.X=0T0127
00470  S.(X,Y):S.(X,Y+1)
00480  N.X:N.Y
00510  F.L=1T05000:N.L
00515  GOS.980
00520  IFZ=100T.REST.
00530  F.M=384T0896S.64
00540  P.ATM
00550  N.M
00560  P.AT960;"
00570  P."      ";
00615  READX
00620  IFX=0T.650
00625  IFX<0T.Y=-X:G.615
00630  IFX>99T.Z=X:G.680
00635  S.(X,Y)
00640  S.(X+1,Y)
00645  G.615
00650  READA

```

```

00655 READB
00660 F.X=ATOB
00665 S.(X,Y)
00670 N.X
00675 G.615
00680 READA$,Q
00690 P.AT409;"FLAG OF ";A$;
00695 S.(0,15):S.(1,15)
00700 G.400
00800 D.-21,0,0,3,22,26,48,50,-22,4,6,22,26,44,46,-23,6,8,22,26
00805 D.40,42,-24,10,12,22,26,36,38,-25,14,16,22,26,32,34,-26,0
00810 D.18,23,0,26,31,-27,0,0,23,0,26,51,-29,0,0,23,0,26,51,-30
00815 D.0,18,23,0,26,31,-31,14,16,22,26,32,34,-32,10,12,22,26,36
00820 D.38,-33,6,8,22,26,40,42,-34,4,6,22,26,44,46,-35,0,0,3,22
00825 D.26,48,50,101,1776,53
00830 D.-21,3,41,-23,17,22,27,-24,12,32,-27,9,36,-30,8,0,16,21
00835 D.0,25,28,36,-31,20,24,-32,19,0,24,28,-33,8,19,24,28,36,-34
00840 D.19,0,25,28,101,1777,46
00845 D.-22,14,-23,8,20,-25,4,24,-27,2,25,-29,4,24,-31,7,21,-32
00850 D.11,17,101,1777,32
00855 D.-22,4,16,28,-25,10,22,34,-28,4,16,28,-31,10,22,34,-34
00860 D.4,16,28,101,1795,40
00865 D.-21,2,8,14,20,26,32,38,44,-24,2,8,14,20,26,32,38,44
00870 D.-27,2,8,14,20,26,32,38,44,-30,1,8,14,20,26,32,38,44
00875 D.-33,2,8,14,20,26,32,38,44,-36,2,8,14,20,26,32,38,44
00880 D.102,1912,49
00885 D.-22,2,8,14,20,26,32,38,-24,5,11,17,23,29,35,41
00890 D.-26,2,8,14,20,26,32,38,-28,5,11,17,23,29,35,41
00895 D.-30,2,8,14,20,26,32,38,-32,5,11,17,23,29,35,41
00900 D.-34,2,8,14,20,26,32,38
00905 D.100,1959,46
00950 D.11,13,15,21,35,37,39,47,49,51,57,59,61,63,9,17,33,41,65
00955 D.21,45,53,57,9,33,41,65,21,45,47,49,51,53,57,9,33,41,65
00960 D.21,45,53,57,9,17,33,41,65,11,13,15,21,23,25,27,29,35,37
00965 D.39,45,53,57,59,61,63
00980 P.AT392,"
01000 P.AT456," ALL PROGRAMS CONTAINED HEREIN ";
01010 P.AT520," COPYRIGHT (C) 1978, CLOAD MAGAZINE ";
01020 P.AT584," BOX 1267 GOLETA, CA 93017. ";
01030 P.AT648," REPRODUCTION BY ANY MEANS PROHIBITED. ";
01040 P.AT712," CLOAD MAGAZINE ASSUMES NO RESPONSIBILITY ";
01050 P.AT776," FOR ERRORS, OMISSIONS OR DAMAGES CAUSED ";
01060 P.AT840," BY THE USE OF THESE PROGRAMS. ";
01065 P.AT904," ";
01070 F.L=1TO3500:N.L:RET.

```

FLAGS LEVEL II

```

00002  REM AUTHOR NEAL P. JENSEN, 1589 BLOSSOM PK, LAKEWOOD, OH 44107
00003  REM BYTES LEFT 13719/1431
00010  CLS
00020  READX
00030  IFX=100THEN100
00040  GOTO20
00100  READA$,Q
00110  Y=2:I=1
00120  READX
00130  SET(X,Y)
00140  IF(I=14)OR(I=19)OR(I=23)OR(I=27)OR(I=34)OR(I=38)OR(I=42)OR(I=47)THENY
00150  I=I+1
00155  IFI=65THEN170
00160  GOTO120
00170  PRINT@98,"TM";
00180  PRINT@292,"BEST OF CLOAD VOL. 1A";
00190  FORX=0TO127
00200  SET(X,0)
00205  SET(X,15)
00220  NEXTX
00230  FORY=0TO15
00240  SET(0,Y)
00245  SET(1,Y)
00250  SET(127,Y)
00255  SET(126,Y)
00260  NEXTY
00265  RESET(2,Y):RESET(125,Y)
00270  RESTORE
00280  GOTO615
00400  FORY=21TO33STEP4
00410  FORX=0TO127
00420  SET(X,Y):SET(X,Y+1)
00430  NEXTX:NEXTY
00450  FORY=37TO45STEP4
00460  FORX=0TO127
00470  SET(X,Y):SET(X,Y+1)
00480  NEXTX:NEXTY
00510  FORL=1TO5000:NEXTL
00515  GOSUB980
00520  IFZ=100THENRESTORE
00530  FORM=384TO896STEP64
00540  PRINT@M,,,," ";
00550  NEXTM
00560  PRINT@960," ";
00570  PRINT" ";
00615  READX
00620  IFX=0THEN650

```

Y=Y+1

```

00625 IFX<0THENY=-X:GOTO615
00630 IFX>99THENZ=X:GOTO680
00635 SET(X,Y)
00640 SET(X+1,Y)
00645 GOTO615
00650 READA
00655 READB
00660 FORX=ATOB
00665 SET(X,Y)
00670 NEXTX
00675 GOTO615
00680 READA$,Q
00690 PRINT@409,"FLAG OF ";A$;
00695 SET(0,15):SET(1,15)
00700 GOTO400
00800 DATA-21,0,0,3,22,26,48,50,-22,4,6,22,26,44,46,-23,6,8,22,26
00805 DATA40,42,-24,10,12,22,26,36,38,-25,14,16,22,26,32,34,-26,0
00810 DATA18,23,0,26,31,-27,0,0,23,0,26,51,-29,0,0,23,0,26,51,-30
00815 DATA0,18,23,0,26,31,-31,14,16,22,26,32,34,-32,10,12,22,26,36
00820 DATA38,-33,6,8,22,26,40,42,-34,4,6,22,26,44,46,-35,0,0,3,22
00825 DATA26,48,50,101,1776,53
00830 DATA-21,3,41,-23,17,22,27,-24,12,32,-27,9,36,-30,8,0,16,21
00835 DATA0,25,28,36,-31,20,24,-32,19,0,24,28,-33,8,19,24,28,36,-34
00840 DATA19,0,25,28,101,1777,46
00845 DATA-22,14,-23,8,20,-25,4,24,-27,2,25,-29,4,24,-31,7,21,-32
00850 DATA11,17,101,1777,32
00855 DATA-22,4,16,28,-25,10,22,34,-28,4,16,28,-31,10,22,34,-34
00860 DATA4,16,28,101,1795,40
00865 DATA-21,2,8,14,20,26,32,38,44,-24,2,8,14,20,26,32,38,44
00870 DATA-27,2,8,14,20,26,32,38,44,-30,1,8,14,20,26,32,38,44
00875 DATA-33,2,8,14,20,26,32,38,44,-36,2,8,14,20,26,32,38,44
00880 DATA102,1912,49
00885 DATA-22,2,8,14,20,26,32,38,-24,5,11,17,23,29,35,41
00890 DATA-26,2,8,14,20,26,32,38,-28,5,11,17,23,29,35,41
00895 DATA-30,2,8,14,20,26,32,38,-32,5,11,17,23,29,35,41
00900 DATA-34,2,8,14,20,26,32,38
00905 DATA100,1959,46
00950 DATA11,13,15,21,35,37,39,47,49,51,57,59,61,63,9,17,33,41,65
00955 DATA21,45,53,57,9,33,41,65,21,45,47,49,51,53,57,9,33,41,65
00960 DATA21,45,53,57,9,17,33,41,65,11,13,15,21,23,25,27,29,35,37
00965 DATA39,45,53,57,59,61,63
00980 PRINT@392,"
01000 PRINT@456," ALL PROGRAMS CONTAINED HEREIN
01010 PRINT@520," COPYRIGHT (C) 1978, CLOAD MAGAZINE
01020 PRINT@584," BOX 1267 GOLETA, CA 93017.
01030 PRINT@648," REPRODUCTION BY ANY MEANS PROHIBITED.
01040 PRINT@712," CLOAD MAGAZINE ASSUMES NO RESPONSIBILIY
01050 PRINT@776," FOR ERRORS, OMISSIONS OR DAMAGES CAUSED
01060 PRINT@840," BY THE USE OF THESE PROGRAMS.
01065 PRINT@904,"
01075 FORL=1TO3500:NEXTL:RETURN

```

";
";
";
";
";
";
";
";
";

";

YIELD-TO-MATURITY LEVEL I

```

00002 REM C.W. EVANS, (602) 933-1616
00003 REM BYTES LEFT 792
00005 REM-----YIELD-TO-MATURITY-----
00009 REM 7/7/78
00010 REM 10 A.M.
00040 S=10:REM CONVERGENCE FACTOR
00049 CLS
00050 P.TAB(20);"YIELD-TO-MATURITY":P.
00052 P." THIS PROGRAM COMPUTES BASIS BOOK YIELD-TO-MATURITY
00054 P."(YTM) OR YIELD-TO-CALL FOR A BOND USING THE CONVENTIONAL
00055 P."30-DAY MONTH FOR CORPORATE, MUNICIPAL, AND MOST FEDERAL
00056 P."AGENCY BONDS. IT APPROXIMATES TREASURIES WHICH ARE
00058 P."COMPUTED ON ACTUAL DAYS.":P.
00060 P." SALE AND MATURITY DATES ARE ENTERED WITH THE MONTH
00062 P."(NUMERICAL), DAY, AND YEAR SEPARATED BY COMMAS.
00063 P." NOTE: USE LAST 2 DIGITS ONLY FOR THE YEAR.
00064 P."SALE AND CALL PRICES ARE ENTERED PER HUNDRED DOLLARS OF
00066 P."MATURITY VALUE. COUPON RATES AND 'YTM' ARE IN PER CENT.
00067 P.:I."HIT ENTER WHEN READY ";A$:CLS:P.:P.
00068 P." THE PROGRAM PERFORMS AN ITERATIVE SOLUTION, INITIALLY
00070 P."ESTIMATING 'YTM', COMPARING PRICE COMPUTED BY FORMULA TO
00072 P."THE SALE PRICE, THEN REFINING 'YTM', REPEATING THE PROCESS
00074 P."UNTIL THE DIFFERENCE BETWEEN THE COMPUTED AND SALE PRICE
00076 P."IS INSIGNIFICANT.
00077 P.:I."HIT ENTER WHEN READY ";A$:CLS:P.:P.
00079 I."WHAT IS COUPON RATE (%)" ;K
00080 I."WHAT IS SALE DATE ";M,D,Y:V=Y+M/12+D/360
00085 I."WHAT IS MATURITY DATE ";M,D,Y:W=Y+M/12+D/360
00086 I."WHAT IS THE ASKED PRICE ";F
00088 IFW<VW=W+100
00090 T=K:N=W-V
00091 I."ENTER 1 IF BOND CALLABLE, 0 IF NOT.":I
00092 IFI=0G.95
00093 I."WHAT IS CALL DATE ";M,D,Y:Q=Y+M/12+D/360
00094 I."WHAT IS CALL PRICE ";R
00095 J=1-((2*N-INT(2*N))/2)
00099 P."DATES, COUPON RATE (%):
00100 P.V;W,K
00105 P.:P."ITERATION--PRICE DIFFERENCE:
00106 GOS.110
00107 P.:P.:P."YIELD-TO-MATURITY IS ";T;"%
00108 G.200
00110 U=T/200+1
00120 X=U:Y=J:GOS.30120:G=P

```

```

00130 X=U:Y=2*N:GOS.30120:G=G-1/P
00140 G=G*K/T
00150 X=U:Y=2*N:GOS.30120:G=G+1/P-K*J/200
00160 G=G*100
00170 P.G-F;
00180 IFABS(G-F)<.01G.197
00190 T=T+(G-F)/S
00195 G.110
00197 T=INT(100*T+.5)/100
00198 RET.
00200 IFI=0E.
00210 P.
00260 IFQ<VQ=Q+100
00270 N=Q-V:F=100*F/R
00280 GOS.110
00290 P.:P.:P."YIELD-TO-CALL IS ";T;"%
30000 E.
30100 REM EXPONENTIATION INPUT X,Y; OUTPUT P
30110 REM ALSO USES E,L,A,B,C INTERNALLY
30120 P=1:E=0:IFY=0T.RET.
30130 IF(X<0)*(INT(Y)=Y)T.P=1-2*Y+4*INT(Y/2):X=-X
30140 IFX<>0T.GOS.30190:X=Y*L:GOS.30250
30150 P=P*E:RET.
30170 REM IN. X, OUTPUT L=LN, X=LOG
30180 REM ALSO USES A,B,C INTERNALLY
30190 E=0:IFX<0T.P."LOG UNDEFINED AT";X:STOP
30195 A=1:B=2:C=.5
30200 IFX>AT.X=C*X:E=E+A:G.30200
30205 IFX<CT.X=B*X:E=E-A:G.30205
30210 X=(X-.707107)/(X+.707107):L=X*X
30215 L=(((.598979*L+.961471)*L+2.88539)*X+E-.5)*.693147
30220 IFABS(L)<1E-6T.L=0
30225 X=L/2.302585:RET.
30240 REM EXPONENTIAL INPUT X,OUTPUT E
30245 REM ALSO USES L,A INTERNALLY
30250 L=INT(1.4427*X)+1:IFL<127T.30265
30255 IF X>0T.P."OVERFLOW":STOP
30260 E=0:RET.
30265 E=.693147*L-X:A=1.32988E-3-1.41316E-4*E
30270 A=((A*E-8.30136E-3)*E+4.16574E-2)*E
30275 E=((A-.166665)*E+.5)*E-1)*E+1:A=2
30280 IFL<0T.A=.5:L=-L:IFL=0T.RET.
30285 F.X=1TOL:E=A*E:N.X:RET.

```

YIELD-TO-MATURITY LEVEL II

```

00002 REM C.W. EVANS, (602) 933-1616
00003 REM 9806 AMBER TRAIL, SUN CITY AZ. 85351
00005 REM-----YIELD-TO-MATURITY-----
00009 :REM.. 8/31/78
00040 S=10:REM CONVERGENCE FACTOR
00049 CLS
00050 PRINTTAB(20);"YIELD-TO-MATURITY":PRINT
00052 PRINT" THIS PROGRAM COMPUTES BASIS BOOK YIELD-TO-MATURITY
00054 PRINT"(YTM) OR YIELD-TO-CALL FOR A BOND USING THE CONVENTIONAL
00055 PRINT"30-DAY MONTH FOR CORPORATE, MUNICIPAL, AND MOST FEDERAL
00056 PRINT"AGENCY BONDS. IT APPROXIMATES TREASURIES WHICH ARE
00058 PRINT"COMPUTED ON ACTUAL DAYS.":PRINT
00060 PRINT" SALE AND MATURITY DATES ARE ENTERED.WITH THE MONTH
00062 PRINT"(NUMERICAL), DAY, AND YEAR SEPARATED BY COMMAS.
00063 PRINT" NOTE: USE LAST 2 DIGITS ONLY FOR THE YEAR.
00064 PRINT"SALE AND CALL PRICES ARE ENTERED PER HUNDRED DOLLARS OF
00066 PRINT"MATURITY VALUE. COUPON RATES AND 'YTM' ARE IN PER CENT.
00067 PRINT:INPUT"HIT ENTER TO CONTINUE ";A$:CLS:PRINT:PRINT
00068 PRINT" THE PROGRAM PERFORMS AN ITERATIVE SOLUTION, INITIALLY
00070 PRINT"ESTIMATING 'YTM', COMPARING PRICE COMPUTED BY FORMULA TO
00072 PRINT"THE SALE PRICE, THEN REFINING 'YTM', REPEATING THE PROCESS
00074 PRINT"UNTIL THE DIFFERENCE BETWEEN THE COMPUTED AND SALE PRICE
00075 PRINT"IS INSIGNIFICANT.
00076 GOTO300
00077 PRINT:INPUT"HIT ENTER WHEN READY ";A$:CLS:PRINT:PRINT
00079 INPUT"WHAT IS COUPON RATE (%)" ;K
00080 INPUT"WHAT IS SALE DATE " ;M,D,Y:V=Y+M/12+D/360
00085 INPUT"WHAT IS MATURITY DATE " ;M,D,Y:W=Y+M/12+D/360
00086 INPUT"WHAT IS THE ASKED PRICE " ;F
00088 IFW<VTHENW=W+100
00090 T=K:N=W-V
00091 INPUT"TYPE 1 IF BOND CALLABLE, 0 IF NOT." ;I
00092 IFI=0GOTO95
00093 INPUT"WHAT IS CALL DATE " ;M,D,Y:Q=Y+M/12+D/360
00094 INPUT"WHAT IS CALL PRICE " ;R
00095 J=1-((2*N-INT(2*N))/2)
00099 PRINT"DATES, COUPON RATE (%):
00100 PRINTV;W,K
00105 PRINT:PRINT"ITERATION--PRICE DIFFERENCE:
00106 GOSUB110
00107 PRINT:PRINT" YIELD-TO-MATURITY IS " ;T;"%
00108 GOTO200
00110 U=T/200+1

```

```
00120 G=(U^J-U^(-2*N))*K/T+U^(-2*N)-K*J/200
00160 G=G*100
00170 PRINTG-F;
00180 IFABS(G-F)<.01GOTO197
00190 T=T+(G-F)/S
00195 GOTO110
00197 T=INT(100*T+.5)/100
00198 RETURN
00200 IFI=0END
00210 PRINT
00260 IFQ<VTHENQ=Q+100
00270 N=Q-V:F=100*F/R
00280 GOSUB110
00290 PRINT:PRINT"      YIELD-TO-CALL IS ";T;"%"
00291 PRINT
00295 INPUT"TO WORK ANOTHER PROBLEM--HIT ENTER.";A$
00296 CLS:PRINT:PRINT:GOTO79
00300 PRINT
00310 PRINT"      THE PROGRAM USES THE FORMULA GIVEN IN THE
00320 PRINT"MAY 1973 HEWLETT-PACKARD JOURNAL AS USED FOR THE
00330 PRINT"HP-80 BUSINESS CALCULATOR.
00340 GOTO77
```

SANDCASTLE LEVEL I

```

00500 CLS:N=2.434E-4:IN."HI THERE! WHAT'S YOUR NAME";A$
00510 P.:P."THIS IS THE GAME OF SAND CASTLE."
00520 P.:IN."WOULD YOU LIKE INSTRUCTIONS";A
00530 IFA=NT.P."WISE GUY":G.700
00540 CLS
00550 P."THE OBJECT OF THIS GAME IS TO BUILD A MOUNTAIN 18 LAYERS "
00560 P."HIGH. YOU ARE GIVEN THREE MATERIALS TO WORK WITH: SAND, GRAVEL,"
00570 P."AND BRICK. THE ADVANTAGE OF SAND IS THAT YOU GET A LOT OF IT."
00580 P."THE DISADVANTAGE IS THAT IT WILL SLIDE IF YOU TRY TO STACK IT"
00590 P."TOO STEEP, FORCING YOU TO START ALL OVER. (EVERY LAYER OF SAND"
00600 P."MUST BE AT LEAST 6 METERS NARROWER THAN THE LAYER UNDERNEATH)"
00610 P."GRAVEL CAN BE STACKED STEEPER (EVERY LAYER MUST BE AT LEAST 4"
00620 P."METERS SMALLER THAN THE ONE BELOW). YOU ALSO GET LESS OF IT."
00630 P."BRICKS CAN BE LAID STEEPEST OF ALL. (WITHIN TWO METERS)"
00640 P."BUT IT'S EXPENSIVE - YOU GET VERY LITTLE OF IT."
00650 P.
00660 P."THE STARTING FOUNDATION IS 100 METERS WIDE, SO YOUR FIRST LAYER"
00670 P."MUST BE LESS THAN 95 METERS WIDE (OF SAND)."

```

```
00910 IFZ>F-4T.H=1
00920 F=Z:G.970
00930 C=C-V:IFC<0T.C=C+V:G.960
00940 IFZ>F-2T.H=1
00950 F=Z:G.970
00960 P.:P.:P."NOT ENOUGH ";B$:P.:F.I=1TO2000:N.I:RET.
00970 F.X=64-INT(Z/2)TO63+INT(Z/2+.6):S.(X,Y):S.(X,Y-1):N.X:Y=Y-2:RET.
00980 P.AT0,"":P.:P.:P." * * * ";B$;"SLIDE * * *"
00990 F.I=1TO2000:N.I
00992 CLS:P."YOU BUILT ";(47-Y)/2;" LAYERS. THAT'S QUITE A PILE."
01000 P.:IN."WOULD YOU LIKE TO TRY AGAIN";A:IFA=NT.P."O.K, BYE":END
01010 IN."WOULD YOU LIKE TO REVIEW THE INSTRUCTIONS";A
01020 IFA=NT.730
01030 G.540
01040 P.AT0:P.:P.:P.
01050 F.Y=47TO16STEP-1:R.(64,Y):N.Y
01060 F.Y=11TO0STEP-1:S.(64,Y):N.Y
01070 F.X=65TO79:S.(X,0):S.(X,1):S.(X,2):S.(X,3):S.(X,4):N.X
01080 P.A$;"'S CASTLE";
01090 G.1090
```

SANDCASTLE LEVEL II

```

00500 CLS:N=2.434E-4:INPUT"HI THERE! WHAT'S YOUR NAME";A$
00510 PRINT:PRINT"THIS IS THE GAME OF SAND CASTLE."
00520 PRINT:INPUT"WOULD YOU LIKE INSTRUCTIONS";A$
00530 IFLEFT$(A$,1)="N"THENPRINT"WISE GUY":GOTO700
00540 CLS
00550 PRINT"THE OBJECT OF THIS GAME IS TO BUILD A MOUNTAIN 18 LAYERS "
00560 PRINT"HIGH. YOU ARE GIVEN THREE MATERIALS TO WORK WITH: SAND, GRAVEL,
00570 PRINT"AND BRICK. THE ADVANTAGE OF SAND IS THAT YOU GET A LOT OF IT."
00580 PRINT"THE DISADVANTAGE IS THAT IT WILL SLIDE IF YOU TRY TO STACK IT"
00590 PRINT"TOO STEEP, FORCING YOU TO START ALL OVER. (EVERY LAYER OF SAND"
00600 PRINT"MUST BE AT LEAST 6 METERS NARROWER THAN THE LAYER UNDERNEATH)"
00610 PRINT"GRAVEL CAN BE STACKED STEEPER. (EVERY LAYER MUST BE AT LEAST 4"
00620 PRINT"METERS SMALLER THAN THE ONE BELOW). YOU ALSO GET LESS OF IT."
00630 PRINT"BRICKS CAN BE LAID STEEPEST OF ALL. (WITHIN TWO METERS)"
00640 PRINT"BUT IT'S EXPENSIVE - YOU GET VERY LITTLE OF IT."
00650 PRINT
00660 PRINT"THE STARTING FOUNDATION IS 100 METERS WIDE, SO YOUR FIRST LAYER
00670 PRINT"MUST BE LESS THAN 95 METERS WIDE (OF SAND)."
00680 PRINT"EACH LAYER IS 4 METERS THICK AND CIRCULAR IN CROSS SECTION, SO"
00690 PRINT"WIDE LAYERS USE UP A LOT OF MATERIAL."
00700 INPUT"PRESS 'ENTER' TO START";B$
00720 S=7.143E-6:G=4.552E-9:B=3.3365E3
00730 CLS:F=100:Y=47:H=0
00735 E=88167:D=5039:C=262
00740 PRINT@0," MATERIALS - SAND:";E;" GRAVEL:";D;" BRICK:";C
00750 PRINT"PROGRESS - LAYERS TO GO:";(Y-11)/2;" THE BASE LAYER IS";F;
00760 PRINT"METERS WIDE."
00765 PRINT:PRINT:PRINT@128,"";
00770 INPUT"HOW MANY METERS WIDE";Z:IFZ>100THENZ=100
00780 V=INT(3.1416*(Z/2)*(Z/2)*4)
00784 INPUT"WHICH MATERIAL (SAND, GRAVEL, OR BRICK) FOR NEXT LAYER"
;A$:A$=LEFT$(A$,1)
00790 PRINT@0,,,, " ":PRINT:PRINT:PRINT:IFY>17THENPRINT:PRINT
00792 PRINT@0,"";
00800 IFZ<1THEN740
00810 IFA$="S"THENB$="SAND":A=S:GOSUB870
00820 IFA$="G"THENB$="GRAVEL":A=G:GOSUB900
00830 IFA$="B"THENB$="BRICK":A=B:GOSUB930
00840 IFH=1THENH=0:GOTO980
00850 IFY=11THEN1040
00860 GOTO740
00870 E=E-V:IFE<0THENE=E+V:GOTO960
00880 IFZ>F-6THENH=1
00890 F=Z:GOTO970
00900 D=D-V:IFD<0THEND=D+V:GOTO960
00910 IFZ>F-4THENH=1
00920 F=Z:GOTO970
00930 C=C-V:IFC<0THENC=C+V:GOTO960

```

```
00940 IFZ>F-2THENH=1
00950 F=Z:GOTO970
00960 PRINT:PRINT:PRINT"NOT ENOUGH ";B$:PRINT:FORI=1TO2000:NEXTI:RETURN
00970 FORX=64-INT(Z/2)TO63+INT(Z/2+.6):SET(X,Y):SET(X,Y-1):NEXTX:Y=Y-2:RETU
00980 PRINT@0,"":PRINT:PRINT:PRINT" * * * ";B$;"SLIDE * * *"
00990 FORI=1TO2000:NEXTI
00992 CLS:PRINT"YOU BUILT ";(47-Y)/2;" LAYERS. THAT'S QUITE A PILE."
01000 PRINT:INPUT"WOULD YOU LIKE TO TRY AGAIN";A$:IFLEFT$(A$,1)="N"THEN
PRINT"O.K, BYE":END
01010 INPUT"WOULD YOU LIKE TO REVIEW THE INSTRUCTIONS";A$
01020 IFLEFT$(A$,1)="N"THEN730
01030 GOTO540
01040 PRINT@0,,,,":PRINT:PRINT:PRINT
01050 FORY=47TO16STEP-1:RESET(64,Y):NEXTY
01060 FORY=11TO0STEP-1:SET(64,Y):NEXTY
01070 FORX=65TO79:SET(X,0):SET(X,1):SET(X,2):SET(X,3):SET(X,4):NEXTX
01080 PRINTA$;"'S CASTLE";
01090 GOTO1090
```

~~KTH~~ KNIGHT'S TOUR
LEVEL I

```

00100 GOS.5050:GOS.1500:GOS.2000:G.3000
00200 REM DRAW KNIGHT
00250 F.I=1TO5
00260 Z=V+1-1
00270 IFI=1J=W:K=W+2
00280 IFI=2J=W-1:K=W+3
00290 IFI=3J=W+1:K=W+4
00300 IFI=4J=W:K=W+3
00310 IFI=5J=W-1:K=W+4
00400 F.X=JTOK
00410 IFS=1S.(X,Z):G.450
00420 R.(X,Z)
00450 N.X
00460 N.I
00470 IFS=1S.(W-2,V+2):S.(W-1,V+2)
00480 IFS<>1R.(W-2,V+2):R.(W-1,V+2)
00500 RET.
01500 REM INITIALIZE
01510 A=1:B=2:C=3:D=4:E=5:F=6:G=7:H=8:Y=1:N=2
01515 S=2:O=0
01520 CLS:P."INITIALIZING";
01530 F.I=1TO128:A(I)=0:N.I
01990 RET.
02000 REM BOARD
02010 CLS:F.I=0TO3
02030 F.J=0TO60STEP20
02040 F.V=6TO10
02050 F.K=0TO9
02060 S.(K+J,V+I*10)
02070 S.(K+J+10,V+5+I*10)
02080 N.K:N.V:N.J:N.I
02150 F.I=1TO4
02155 J=43+128*I
02160 P.A.J,I;
02165 P.A.J+448,I+4;
02170 N.I
02200 IFO=0G.2800
02210 F.I=65TO128
02220 IFA(I)=0G.2500
02230 J=A(I)/8
02240 K=8*(J-INT(J))
02245 J=INT(J)+1
02250 IFK=0K=8:J=J-1
02255 A=1
02256 IF(INT((J+K)/2)-(J+K)/2)=0R.(10*K-9,5*J+5):R.(10*K-8,5*J+5):A=0
02260 IFA=1S.(10*K-9,5*J+5):S.(10*K-8,5*J+5)
02280 A=1
02300 GOS.7000
02500 N.I
02800 P.A.66,"A B C D E F G H";
02900 RET.
03000 REM MOVE
03100 P.A.1;:IN."START AT (X,Y)";L,M
03102 GOS.2800
03105 P=L+2:Q=M+1:GOS.9000:IFI=0G.3250
03110 G.3100
03130 P.A.40,"ILLEGAL MOVE ";
03140 F.I=1TO1500:N.I
03145 GOS.2800
03150 P.A.55," ";:P.A.40;:IN."NEXT MOVE (X,Y)";L,M
03155 GOS.2800
03158 F.I=0TO1:F.J=6TO8:S.(I,J):N.J:N.I
03160 IFL=0GOS.2000:G.3350
03210 GOS.9000:ONI+1G.3250,3130

```

```

03220 P.A.40,"ALREADY VISITED";:G.3140
03250 REM ERASE
03255 IF S=2G.3300
03260 S=ABS(S-1)
03270 GOS.200
03290 IFS=1R.(W-4,V+4):R.(W-3,V+4)
03295 IFS<>1S.(W-4,V+4):S.(W-3,V+4)
03300 P=L:Q=M:S=1
03310 IF INT((P+Q)/2)=(P+Q)/2S=0
03320 O=O+1:I=O+64:J=Q:K=P:GOS.7000
03330 I=P+8*(Q-1)
03340 A(I)=1:A(O+64)=I
03350 V=5*Q+1
03360 W=10*(P-1)+5
03370 GOS.200
03390 J=0:P.A.40,"":GOS.2800
03395 P.A.9,"";
03400 F.L=P-2TOP+2:F.M=Q-2TOQ+2:GOS.9000:IFI=0J=J+1
03410 N.M:N.L
03450 P.A.0,"YOU HAVE";J;"LEGAL MOVES";:IFJ=0G.9700
04800 G.3150
05050 CLS:P."KNIGHT'S TOUR IS A CLASSIC CHESS PROBLEM THE OBJECT"
05060 P."OF WHICH IS TO MOVE A SINGLE KNIGHT TO EACH SQUARE OF"
05070 P."AN EMPTY CHESS BOARD WITHOUT VISITING THE SAME SQUARE"
05080 P."TWICE. YOU CAN START ON ANY SQUARE."
06020 P."":P."THE COMPUTER WILL KEEP TRACK OF ALL MOVES AND"
06030 P."WILL NOT ALLOW AN ILLEGAL MOVE. IF NO LEGAL MOVE TO"
06035 P."AN UNVISITED SQUARE IS POSSIBLE, THE GAME ENDS.. SQUARES"
06040 P."ARE REFERENCED BY A LETTER (A-H) AND A NUMBER (1-8)."
06050 P."ENTER YOUR MOVE BY TYPING THE LETTER FIRST, FOLLOWED"
06060 P."IMMEDIATELY BY A COMMA AND THE NUMBER. THEN HIT ENTER."
06070 P."TO REDRAW THE BOARD AND ALL MOVES, ENTER 0,0":P.""
06080 INPUT"HIT ENTER WHEN YOU'RE READY TO BEGIN";A$
06500 RET.
07000 R=I:I=I-64
07020 P.A.178,I;"SQUARES";
07040 I=I/5
07050 I=240+15*(I-INT(I))+64*INT(I)
07060 IFI<>INT(I)I=INT(I)+1
07100 IFK=1P.A.I,"A";
07110 IFK=2P.A.I,"B";
07120 IFK=3P.A.I,"C";
07130 IFK=4P.A.I,"D";
07140 IFK=5P.A.I,"E";
07150 IFK=6P.A.I,"F";
07160 IFK=7P.A.I,"G";
07170 IFK=8P.A.I,"H";
07200 I=I+1
07210 IFJ=1P.A.I,"1";
07220 IFJ=2P.A.I,"2";
07230 IFJ=3P.A.I,"3";
07240 IFJ=4P.A.I,"4";
07250 IFJ=5P.A.I,"5";
07260 IFJ=6P.A.I,"6";
07270 IFJ=7P.A.I,"7";
07280 IFJ=8P.A.I,"8";
07300 I=R
07500 RET.
09000 I=0:IF(L<1)+(L>8)+(M<1)+(M>8)I=1:G.9500
09100 IFABS((P-L)*(Q-M))<>2I=1:G.9500
09200 IFA(L-8+8*M)<>0I=2
09500 RET.
09700 CLS:IN."PLAY AGAIN";A
09710 IFA=YCLS:G.200
09800 END

```

Knight's Tour Level II

```

00010 DIMA(128)
00100 GOSUB5050:GOSUB1500:GOSUB2000:GOTO3000
00200 REM DRAW KNIGHT
00250 FORI=1TO5
00260 Z=V+I-1
00270 IFI=1J=W:K=W+2
00280 IFI=2J=W-1:K=W+3
00290 IFI=3J=W+1:K=W+4
00300 IFI=4J=W:K=W+3
00310 IFI=5J=W-1:K=W+4
00400 FORX=JTOK
00410 IFS=1SET(X,Z):GOTO450
00420 RESET(X,Z)
00450 NEXTX
00460 NEXTI
00470 IFS=1SET(W-2,V+2):SET(W-1,V+2)
00480 IFS<>1RESET(W-2,V+2):RESET(W-1,V+2)
00500 RETURN
01500 REM INITIALIZE
01515 S=2:O=0
01520 CLS:PRINT"INITIALIZING";
01530 FORI=1TO128:A(I)=0:NEXTI
01990 RETURN
02000 REM BOARD
02010 CLS:FORI=0TO3
02030 FORJ=0TO60STEP20
02040 FORV=6TO10
02050 FORK=0TO9
02060 SET(K+J,V+I*10)
02070 SET(K+J+10,V+5+I*10)
02080 NEXTK:NEXTV:NEXTJ:NEXTI
02150 FORI=1TO4
02155 J=43+128*I
02160 PRINT@J,I;
02165 PRINT@J+448,I+4;
02170 NEXTI
02200 IFO=0GOTO2800
02210 FORI=65TO128
02220 IFA(I)=0GOTO2500
02230 J=A(I)/8
02240 K=8*(J-INT(J))
02245 J=INT(J)+1
02250 IFK=0K=8:J=J-1
02255 A=1
02256 IF(INT((J+K)/2)-(J+K)/2)=0RESET(10*K-9,5*J+5):RESET(10*K-8,5*J+5):A
02260 IFA=1SET(10*K-9,5*J+5):SET(10*K-8,5*J+5)
02280 A=1
02300 GOSUB7000
02500 NEXTI
02800 PRINT@66,"A B C D E F G H";
02900 RETURN
03000 REM MOVE
03100 PRINT@1,"";:INPUT"START AT (X,Y)";L$,M
03101 L=ASC(L$):L=L-64
03102 GOSUB2800
03105 P=L+2:Q=M+1:GOSUB9000:IFI=0GOTO3250
03110 GOTO3100
03130 PRINT@40,"ILLEGAL MOVE ";
03140 FORI=1TO1500:NEXTI
03145 GOSUB2800
03150 PRINT@55,"";:PRINT@40,"";:INPUT"NEXT MOVE (X,Y)";L$,M
03151 L=ASC(L$):L=L-64
03155 GOSUB2800

```

```

03158 FORI=0TO1:FORJ=6TO8:SET(I,J):NEXTJ:NEXTI
03160 IFL=0GOSUB2000:GOTO3350
03210 GOSUB9000:ONI+1GOTO3250,3130
03220 PRINT@40,"ALREADY VISITED";:GOTO3140
03250 REM ERASE
03255 IF S=2GOTO3300
03260 S=ABS(S-1)
03270 GOSUB200
03290 IFS=1RESET(W-4,V+4):RESET(W-3,V+4)
03295 IFS<>1SET(W-4,V+4):SET(W-3,V+4)
03300 P=L:Q=M:S=1
03310 IF INT((P+Q)/2)=(P+Q)/2S=0
03320 O=O+1:I=O+64:J=Q:K=P:GOSUB7000
03330 I=P+8*(Q-1)
03340 A(I)=1:A(O+64)=I
03350 V=5*Q+1
03360 W=10*(P-1)+5
03370 GOSUB200
03390 J=0:PRINT@40,"":GOSUB2800
03395 PRINT@9,"";
03400 FORL=P-2TOP+2:FORM=Q-2TOQ+2:GOSUB9000:IFI=0J=J+1
03410 NEXTM:NEXTL
03450 PRINT@0,"YOU HAVE";J;"LEGAL MOVES";:IFJ=0GOTO3450
04800 GOTO3150
05050 CLS:PRINT"KNIGHT'S TOUR IS A CLASSIC CHESS PROBLEM THE OBJECT"
05060 PRINT"OF WHICH IS TO MOVE A SINGLE KNIGHT TO EACH SQUARE OF"
05070 PRINT"AN EMPTY CHESS BOARD WITHOUT VISITING THE SAME SQUARE"
05080 PRINT"TWICE. YOU CAN START ON ANY SQUARE."
06020 PRINT"":PRINT"THE COMPUTER WILL KEEP TRACK OF ALL MOVES AND"
06030 PRINT"WILL NOT ALLOW AN ILLEGAL MOVE. IF NO LEGAL MOVE TO"
06035 PRINT"AN UNVISITED SQUARE IS POSSIBLE, THE GAME ENDS. SQUARES"
06040 PRINT"ARE REFERENCED BY A LETTER (A-H) AND A NUMBER (1-8)."

```

```

00002  RFM AUTHOR   CLOAD STAFF      (RF)
00003  REM BYTES LEFT 12605/317
00010  CLS
00015  N=12345
00020  P."***** WELCOME TO ONE POCKET POOL *****"
00030  P.:IN."WOULD YOU LIKE TO READ THE INSTRUCTIONS ";I
00040  IF I=N T. 60
00050  GOS.1000
00060  P=54321
00080  CLS
00085  P.AT898,"***** WELCOME TO ONE POCKET POOL *****"
00090  V=0: Z=0: C=50
00100  F. X=10 TO 110
00105  Y=5
00110  SET (X,Y)
00115  Y=30
00120  SET (X,Y)
00130  N.X
00150  F. Y=5 TO 30
00160  X=8
00170  SET(X,Y)
00174  X=9
00178  SET (X,Y)
00180  X=110
00190  SET (X,Y)
00200  X=111
00210  SET (X,Y)
00220  NEXT Y
00305  F.Y=28 TO 30
00310  F.X=106 TO 112
00320  RESET (X,Y )
00330  NEXT X
00335  NEXT Y
00350  R=RND(98)+10
00360  S=RND(23)+6
00370  SET (R,S)
00371  R=R+1
00372  SET (R,S)
00380  B=5
00381  P. AT 0
00382  P.AT3,"1"
00383  P.AT(R/2)-2,R-9
00384  P.AT56,"100"
00385  A=60
00390  C=A
00395  P.AT820
00400  P.AT774;:IN."WHERE WOULD YOU LIKE THE BALL TO HIT THE RAIL";A
00405  A=A+9
00410  IF A=54330 T. 500
00414  IF (A>109)+(A<10)T.V=V+1:G.695
00420  SET(C,B)
00430  RESET (A,B)
00440  G.390
00500  REM.
00505  B=6
00510  D=(C-R)/(B-S)*(-1)
00520  E=D*S+R
00525  K=RND(3)
00526  IF K=2 T. K=10
00527  IF K=3 T. K=100
00530  F.M=S TO B STEP -1
00540  N=(M*D-E)*(-1)
00550  SET(N,M)

```

```

00555 F.I=1 TO K:N.I
00560 N.M
00600 T=((C-R)*2)+R
00610 D=(C-T)/(B-S)*(-1)
00620 E=D*S+T
00630 F.M=B TO 30
00640 N=(M*D-E)*(-1)
00645 IF (N>110)*(M<28)+(N<10)T.680
00650 SET(N,M)
00660 F.I=1 TO K:N.I
00670 N.M
00680 F.I=1 TO 1000 :N.I
00694 IF(N>=106)*(M>27)T.Z=Z+1 :G.699
00695 V=V+1
00699 P.AT896
00700 P.AT903,"OF";V+Z;"SHOT(S); ";Z;"IN THE POCKET, ";V;"MISSED."
00710 F.I=133 TO 581 STEP 64
00720 P.AT I
00730 N.I
00735 P. AT 693
00740 SET (C,5)
00750 F.I=5 TO 27
00755 J=111
00760 SET(J,I)
00770 J=112
00780 SET(J,I)
00790 N.I
00800 G.350
00999 G.999
01000 REM.
01005 P.
01010 P." THIS IS A GAME OF SKILL. THE COMPUTER WILL DRAW A POOL"
01020 P."TABLE ON THE SCREEN. THIS POOL TABLE ONLY HAS ONE POCKET."
01030 P."THE COMPUTER THEN PLACES A SQUARE BALL SOME WHERE ON THE "
01040 P."TABLE AT RANDOM. "
01050 P." THE OBJECT IS TO PUT THE 'BALL' IN THE CORNER POCKET BY"
01060 P."BOUNCING THE 'BALL' OFF THE OPPOSITE RAIL."
01070 P." YOU DETERMINE WHERE THE BALL HITS THE RAIL AT THE TOP.OF"
01080 P."THE SCREEN. THE COMPUTER WILL ASK: 'WHERE WOULD YOU LIKE"
01090 P."THE BALL TO HIT THE RAIL ?' ENTER A NUMBER BETWEEN 1 AND"
01100 P."100 (DON'T GET CUTE HERE, OR THE COMPUTER WILL GET YOU)."
01110 IN." WHEN YOU WISH TO TURN THE PAGE PRESS ENTER";A$
01120 CLS
01130 P."THE COMPUTER WILL 'NOTCH' THE RAIL TO SHOW YOU WHERE YOU"
01140 P."WILL SEE THE 'BALL' BOUNCE OFF THE RAIL. IF YOU DON'T LIKE"
01150 P."THAT SPOT ENTER ANOTHER NO. BETWEEN 1 AND 100 TILL THE NOTCH"
01160 P."IS JUST WHERE YOU WANT IT. WHEN YOU'RE SATISFIED, PRESS"
01165 P."P (FOR POOL OF COURSE) AND ENTER. A DOTTED LINE WILL"
01180 P."THEN SHOW YOU THE PATH OF THE BALL. IF IT GOES INTO THE"
01190 P."POCKET YOU CHOSE WELL; IF NOT, WELL TRY AGAIN. THE SCORE IS "
01200 P."KEPT FOR YOU AT THE BOTTOM OF THE SCREEN."
01210 P." IF YOU GOOF AND PRESS ENTER WITHOUT A NUMBER TO ENTER THE"
01220 P."COMPUTER WILL ASK 'WHAT?'. THIS WILL DISPLACE THE POOL TABLE"
01230 P."SO MUCH THE ONLY WAY OUT IS TO START OVER. BE CAREFUL:!"
01232 P." ONE MORE THING: WHEN THE 'BALL' IS PLACED ON THE TABLE"
01234 P."YOU WILL NOTICE THE COMPUTER HAS LISTED THE POSITION ON THE"
01236 P."RAIL DIRECTLY OPPOSITE THE BALL. THIS IS TO AID YOU IN YOUR"
01250 IN." WHEN YOU WISH TO TURN THE PAGE PRESS ENTER";A$
01251 CLS
01252 P."SELECTION OF THE SPOT WHERE YOU WANT THE BALL TO BOUNCE OFF"
01254 P."THE RAIL. "
01260 P." GOOD LUCK"
01290 IN." WHEN YOU ARE READY TO PLAY 'ONE POCKET POOL' PRESS ENTER";A$
01300 RET.

```

```

00002 REM AUTHOR CLOAD STAFF . (RF)
00003 REM BYTFS LEFT 12605/317
00010 CLS
00015 N=12345
00020 PRINT"***** WELCOME TO ONE POCKET POOL *****"
00030 PRINT:INPUT"WOULD YOU LIKE TO READ THE INSTRUCTIONS ";I$
00040 IFLEFT$(I$,1)="N" THEN 60
00050 GOSUB1000
00060 P=54321
00080 CLS
00085 PRINT@898,"***** WELCOME TO ONE POCKET POOL *****"
00090 V=0: Z=0: C=50
00100 FOR X=10 TO110
00105 Y=5
00110 SET (X,Y)
00115 Y=30
00120 SET (X,Y)
00130 NEXTX
00150 FOR Y=5 TO 30
00160 X=8
00170 SET(X,Y)
00174 X=9
00178 SET (X,Y)
00180 X=110
00190 SET (X,Y)
00200 X=111
00210 SET (X,Y)
00220 NEXT Y
00305 FORY=28 TO 30
00310 FORX=106 TO 112
00320 RESET (X,Y )
00330 NEXT X
00335 NEXT Y
00350 R=RND(98)+10
00360 S=RND(23)+6
00370 SET (R,S)
00371 R=R+1
00372 SET (R,S)
00380 B=5
00381 PRINT@0,,,,";
00382 PRINT@3,"1";
00383 PRINT@(R/2)-2,R-9;
00384 PRINT@56,"100";
00385 A=60
00390 C=A
00395 PRINT@820," ";
00400 PRINT@774,"";:INPUT"WHERE WOULD YOU LIKE THE BALL TO HIT THE RAIL";Z
00402 IFVAL(A$)=0GOTO500
00403 A=VAL(A$)
00405 A=A+9
00410 IF A=54330 THEN 500
00414 IF (A>109)+(A<10)THENV=V+1:GOTC695
00420 SET(C,B)
00430 RESET (A,B)
00440 GOTO390
00500 REM.
00505 B=6
00510 D=(C-R)/(B-S)*(-1)
00520 E=D*S+R
00525 K=RND(3)
00526 IF K=2 THEN K=10
00527 IF K=3 THEN K=100
00530 FORM=S TO B STEP -1
00540 M=(M*D-E)*(-1)

```

```

00550 SET(N,M)
00555 FORI=1 TO K:NEXTI
00560 NEXTM
00600 T=((C-R)*2)+R
00610 D=(C-T)/(B-S)*(-1)
00620 E=D*S+T
00630 FORM=B TO 30
00640 N=(M*D-E)*(-1)
00645 IF (N>110)*(M<28)+(N<10)THEN680
00650 SET(N,M)
00660 FORI=1 TO K:NEXTI
00670 NEXTM
00680 FORI=1 TO 1000 :NEXTI
00694 IF(N>=106)*(M>27)THENZ=Z+1 :GOTO699
00695 V=V+1
00699 PRINT@896,,,,;" ";
00700 PRINT@903,"OF";V+Z;"SHOT(S); ";Z;"IN THE POCKET, ";V;"MISSED."
00710 FORI=133 TO 581 STEP 64
00720 PRINT@ I,,,,;" ";
00730 NEXTI
00735 PRINT@693,""
00740 SET (C,5)
00750 FORI=5 TO 27
00755 J=111
00760 SET(J,I)
00770 J=112
00780 SET(J,I)
00790 NEXTI
00800 GOTO350
00999 GOTO999
01000 REM.
01005 PRINT
01010 PRINT" THIS IS A GAME OF SKILL. THE COMPUTER WILL DRAW A POOL"
01020 PRINT"TABLE ON THE SCREEN. THIS POOL TABLE ONLY HAS ONE POCKET."
01030 PRINT"THE COMPUTER THEN PLACES A SQUARE BALL SOME WHERE ON THE "
01040 PRINT"TABLE AT RANDOM. "
01050 PRINT" THE OBJECT IS TO PUT THE 'BALL' IN THE CORNER POCKET BY"
01060 PRINT"BOUNCING THE 'BALL' OFF THE OPPOSITE RAIL."
01070 PRINT" YOU DETERMINE WHERE THE BALL HITS THE RAIL AT THE TOP OF"
01080 PRINT"THE SCREEN. THE COMPUTER WILL ASK: 'WHERE WOULD YOU LIKE"
01090 PRINT"THE BALL TO HIT THE RAIL ?' ENTER A NUMBER BETWEEN 1 AND"
01100 PRINT"100 (DON'T GET CUTE HERE, OR THE COMPUTER WILL GET YOU)."

```

Y = MX + B LEVEL I

```

00001 N=123
00002 REM AUTHOR CLOAD STAFF (RF)
00003 REM BYTES LEFT 13360/1072
00005 CLS
00008 R=0:Q=0:S=0:O=0:P=0
00009 W=0:E=0
00010 P."THIS PROGRAM WILL TAKE TWO SETS OF DATA AND REDUCE THEM TO AN
00012 P."EQUATION FOR A LINE THAT CAN BE USED TO ESTIMATE OTHER VALUES.
00013 P.:IN."DO YOU WISH INSTRUCTIONS ";X
00014 N=0:IFX=123G.66
00015 P.:P."LETS TAKE AN EXAMPLE.
00016 P."YOUR SPEEDOMETER WAS JUST CALIBRATED BY AAA. YOU NOTICE THAT
00018 P."THEY STOP AT 55 MPH. YOU NEED TO KNOW HOW FAST YOUR GOING WHEN
00020 P."YOUR SPEEDOMETER SAYS 100 MPH, NEGITIVE PERSPIRATION.
00021 GOS.1000
00022 P."THE DATA THEY GAVE YOU IS AS FOLLOWS:
00024 P."SPEEDOMETER          ACTUAL SPEED"
00026 P."  30";T.(23);"      30"
00027 P."  35";T.(23);"      34"
00028 P."  40";T.(23);"      38"
00030 P."  45";T.(23);"      43"
00032 P."  50";T.(23);"      47"
00034 P."  55";T.(23);"      52"
00036 P.:P."THE COMPUTER WILL ASK THE NAMES OF THE TWO SETS OF DATA.
00037 P."YOU WILL HAVE YOUR OWN NAMES. IN THE EXAMPLE USED HERE
00038 P."THE FIRST NAME IS SPEEDOMETER; THE SECOND IS ACTUAL SPEED.
00040 P."THEN THE COMPUTER WILL ASK FOR THE DATA, ALTERNATELY. WHEN YOU
00042 P."HAVE PLACED ALL THE DATA YOU HAVE IN THE COMPUTER ENTER ZERO
00043 GOS.1000
00044 P."FOR BOTH SPEEDOMETER AND ACTUAL SPEED, THIS STARTS THE
00046 P."CALCULATING. THE COMPUTER WILL THEN SHOW YOU THE EQUATION FOR
00048 P."THE CONVERSION. IT WILL ASK YOU FOR A VALUE FOR THE SPEED-
00050 P."OMETER AND GIVE YOU THE CALCULATED VALUE OF ACTUAL SPEED.
00052 P."IF YOU WOULD LIKE THE INFORMATION IN REVERSE; THAT IS,
00054 P."WHAT WILL YOUR SPEEDOMETER SAY WHEN YOUR GOING 55 OR EVEN
00056 P."100 MPH ACTUAL SPEED, ENTER 12345. THIS WILL TRIGGER THE
00058 P."COMPUTER TO ASK THE QUESTION THE OTHER WAY. AFTER ALL,
00060 P."A MAN NEEDS A CAR THAT WILL GO 100 MPH TO GET TO WORK,
00062 P."DOESN'T HE?
00064 GOS.1000
00065 CLS
00066 IN."WHAT WILL YOU CALL YOUR FIRST SET OF DATA ";A$
00068 IN."WHAT WILL YOU CALL YOUR SECOND SET OF DATA ";B$
00070 P.:P."WHAT IS THE VALUE FOR THE ";A$;:IN.X
00072 P."WHAT IS THE CORRESPONDING VALUE FOR THE ";B$;:IN.Y
00102 IF(X=0)*(Y=0)G.150
00105 N=N+1
00108 P."YOU HAVE ";N;" SET(S) OF DATA IN THE COMPUTER."
00110 R=R+X
00120 O=O+X*X
00125 E=E+Y*Y
00130 P=P+Y
00140 Q=Q+X*Y
00145 G.70

```

```
00150 M=(N*Q-R*P)/(N*O-R*R)
00160 B=(O*P-R*Q)/(N*O-R*R)
00169 CLS
00170 P."Y =";M;"X + (";B;)"
00180 P."IF THE ";A$;" IS/ARE "":IN.A
00185 IFA=12345G.300
00190 P."THE ";B$;" IS/ARE ";A*M+B
00195 P.
00200 G.180
00300 W=W+1
00310 U=W/2-INT(W/2)
00320 IFU=0G.150
00330 IFU<>0G.400
00400 M=(N*Q-R*P)/(N*E-P*P)
00410 B=(E*R-P*Q)/(N*E-P*P)
00430 CLS
00440 P."X =";M;"Y + (";B;)"
00450 P."IF THE ";B$;" IS/ARE "":IN.A
00455 IFA=12345G.300
00460 P."THE ";A$;" IS/ARE ";A*M+B
00470 P.
00480 G.450
01000 P.:IN."PRESS ENTER TO TURN THE PAGE ";A$
01001 CLS:RET.
```

Y = MX + B LEVEL II

```

00002 REM AUTHOR CLOAD STAFF (RF)
00003 REM BYTES LEFT 13360/1072
00005 CLS
00008 R=0:Q=0:S=0:O=0:P=0
00009 W=0:E=0
00010 PRINT"THIS PROGRAM WILL TAKE TWO SETS OF DATA AND REDUCE THEM TO AN
00012 PRINT"EQUATION FOR A LINE THAT CAN BE USED TO ESTIMATE OTHER VALUES.
00013 PRINT:INPUT"DO YOU WISH INSTRUCTIONS ";X$
00014 N=0:IFLEFT$(X$,1)="N"THEN66
00015 PRINT:PRINT"LETS TAKE AN EXAMPLE.
00016 PRINT"YOUR SPEEDOMETER WAS JUST CALIBRATED BY AAA. YOU NOTICE THAT
00018 PRINT"THEY STOP AT 55 MPH. YOU NEED TO KNOW HOW FAST YOUR GOING WHE
00020 PRINT"YOUR SPEEDOMETER SAYS 100 MPH, NEGITIVE PERSPIRATION.
00021 GOSUB1000
00022 PRINT"THE DATA THEY GAVE YOU IS AS FOLLOWS:
00024 PRINT"SPEEDOMETER          ACTUAL SPEED"
00026 PRINT"   30";TAB(23);"   30"
00027 PRINT"   35";TAB(23);"   34"
00028 PRINT"   40";TAB(23);"   38"
00030 PRINT"   45";TAB(23);"   43"
00032 PRINT"   50";TAB(23);"   47"
00034 PRINT"   55";TAB(23);"   52"
00036 PRINT:PRINT"THE COMPUTER WILL ASK THE NAMES OF THE TWO SETS OF DATA.
00037 PRINT"YOU WILL HAVE YOUR OWN NAMES. IN THE EXAMPLE USED HERE
00038 PRINT"THE FIRST NAME IS SPEEDOMETER; THE SECOND IS ACTUAL SPEED.
00040 PRINT"THEN THE COMPUTER WILL ASK FOR THE DATA, ALTERNATELY. WHEN YOU
00042 PRINT"HAVE PLACED ALL THE DATA YOU HAVE IN THE COMPUTER ENTER ZERO
00043 GOSUB1000
00044 PRINT"FOR BOTH SPEEDOMETER AND ACTUAL SPEED, THIS STARTS THE
00046 PRINT"CALCULATING. THE COMPUTER WILL THEN SHOW YOU THE EQUATION FOR
00048 PRINT"THE CONVERSION. IT WILL ASK YOU FOR A VALUE FOR THE SPEED-
00050 PRINT"OMETER AND GIVE YOU THE CALCULATED VALUE OF ACTUAL SPEED.
00052 PRINT"IF YOU WOULD LIKE THE INFORMATION IN REVERSE; THAT IS,
00054 PRINT"WHAT WILL YOUR SPEEDOMETER SAY WHEN YOUR GOING 55 OR EVEN
00056 PRINT"100 MPH ACTUAL SPEED, ENTER 12345. THIS WILL TRIGGER THE
00058 PRINT"COMPUTER TO ASK THE QUESTION THE OTHER WAY. AFTER ALL,
00060 PRINT"A MAN NEEDS A CAR THAT WILL GO 100 MPH TO GET TO WORK,
00062 PRINT"DOESN'T HE?
00064 GOSUB1000
00065 CLS
00066 INPUT"WHAT WILL YOU CALL YOUR FIRST SET OF DATA ";A$
00068 INPUT"WHAT WILL YOU CALL YOUR SECOND SET OF DATA ";B$
00070 PRINT:PRINT"WHAT IS THE VALUE FOR THE ";A$;:INPUTX
00072 PRINT"WHAT IS THE CORRESPONDING VALUE FOR THE ";B$;:INPUTY
00102 IF(X=0)*(Y=0)GOTO150
00105 N=N+1
00108 PRINT"YOU HAVE ";N;" SET(S) OF DATA IN THE COMPUTER."
00110 R=R+X
00120 O=O+X*X
00125 E=E+Y*Y
00130 P=P+Y
00140 Q=Q+X*Y
00145 GOTO70

```

```
00150 M=(N*Q-R*P)/(N*O-R*R)
00160 B=(O*P-R*Q)/(N*O-R*R)
00169 CLS
00170 PRINT"Y =";M;"X + (";B;)"
00180 PRINT"IF THE ";A$;" IS/ARE ";;INPUTA
00185 IFA=12345GOTO300
00190 PRINT"THE ";B$;" IS/ARE ";;A*M+B
00195 PRINT
00200 GOTO180
00300 W=W+1
00310 U=W/2-INT(W/2)
00320 IFU=0GOTO150
00330 IFU<>0GOTO400
00400 M=(N*Q-R*P)/(N*E-P*P)
00410 B=(E*R-P*Q)/(N*E-P*P)
00430 CLS
00440 PRINT"X =";M;"Y + (";B;)"
00450 PRINT"IF THE ";B$;" IS/ARE ";;INPUTA
00455 IFA=12345GOTO300
00460 PRINT"THE ";A$;" IS/ARE ";;A*M+B
00470 PRINT
00480 GOTO450
01000 PRINT:INPUT"PRESS ENTER TO TURN THE PAGE ";A$
01001 CLS:RETURN
```

BREAK LEVEL I

```

00500 IFN=7.2135E-5T.600
00510 N=7.2135E-5:CLS:P."THE GAME OF BREAK":P.:M=0:G=0
00520 P."BREAK THE WINDOW, THAT IS ..."
00530 P.:IN."DO YOU WANT INSTRUCTIONS";A:IFA<>NT.GOS.900
00540 CLS:F.I=0T023:S.(127,I):N.I
00550 T=RND(5):P.AT512,"";
00560 F.X=1T063:R.(X-1,11):S.(X,11):F.I=1TOT:N.I:N.X
00570 F.X=64T0126:R.(X-1,11):R.(X,11):F.I=1TOT:N.I:N.X
00580 P."TOO LATE - THE BRICK JUST BROKE THE WINDOW"
00590 F.I=1T02000:N.I
00600 S=INT((X-64)*1.61+.5)
00610 IFS>100T.S=0:G.630
00620 S.(X,11)
00630 P.:P."YOUR SCORE FOR THIS THROW IS";S;"POINTS"
00640 G=G+S:M=M+1
00650 P.:P."YOUR AVERAGE IS";G/M;"POINTS"
00660 P.:IN."DO YOU WANT TO TRY AGAIN";A
00670 IFA<>NT.540
00680 CLS:IN."DOES ANYONE ELSE WANT TO PLAY";A
00690 IFA<>NT.510
00700 P."O.K. - BYE FOR NOW":END
00900 CLS
00901 P.
00905 P."I WILL DRAW A WINDOW ON THE RIGHT SIDE OF THE SCREEN, AND THEN"
00910 P."I'LL LAUNCH A BRICK FROM THE LEFT SIDE. AT THE EXACT TIME THAT"
00920 P."THE WINDOW IS HIT, PRESS THE 'BREAK' BUTTON (AT THE TOP RIGHT"
00930 P."CORNER OF THE KEYBOARD). THE COMPUTER WILL SAY SOMETHING LIKE"
00940 P." 'BREAK AT 570' OR SOME SUCH TRASH. IGNORE IT AND TYPE IN 'RUN'"
00950 P."AND HIT THE 'ENTER' KEY. (AFTER ALL, YOU JUST BROKE THE WINDOW"
00960 P."WHAT ELSE DO YOU DO BUT RUN ?)"
00970 P.:P."THE ONLY REAL DIFFICULTY IS THAT THE BIONIC BRICK SPINS AT"
00980 P."SUPER-SPEED AND TURNS INVISIBLE HALFWAY ACROSS THE SCREEN."
00990 P.:F.I=1T01500:N.I
01000 IN."PRESS 'ENTER' TO START";A$
01010 RET.

```

BREAK LEVEL II

```
00500 IFN=7.2135E-5THEN600
00510 N=7.0135E-5:CLS:PRINT"THE GAME OF BREAK":PRINT:M=0:G=0
00520 PRINT"BREAK THE WINDOW, THAT IS ..."
00530 PRINT:INPUT"DO YOU WANT INSTRUCTIONS";A$:IFLEFT$(A$,1)<>"N"GOSUB900
00540 CLS:FORI=0TO23:SET(127,I):NEXTI
00550 T=RND(5):PRINT@512,"";
00560 FORX=1TO63:RESET(X-1,11):SET(X,11):FORI=1TOT:NEXTI:NEXTX
00570 FORX=64TO126:RESET(X-1,11):RESET(X,11):FORI=1TOT:NEXTI:BS=INKEY$:IF$
NEXTX
00575 IFX<>127GOTO600
00580 PRINT"TOO LATE - THE BRICK JUST BROKE THE WINDOW"
00590 FORI=1TO2000:NEXTI
00600 S=INT((X-64)*1.61+.5)
✓ 00610 IFS>100THENS=0:GOTO630
00620 SET(X,11)
00630 PRINT:PRINT"YOUR SCORE FOR THIS THROW IS";S;"POINTS"
00640 G=G+S:M=M+1
00650 PRINT:PRINT"YOUR AVERAGE IS";G/M;"POINTS"
00660 PRINT:INPUT"DO YOU WANT TO TRY AGAIN";A$
00670 IFLEFT$(A$,1)<>"N"THEN540
00680 CLS:INPUT"DOES ANYONE ELSE WANT TO PLAY";A$
00690 IFLEFT$(A$,1)<>"N"THEN510
00700 PRINT"O.K. - BYE FOR NOW":END
00900 CLS
00901 PRINT
00905 PRINT"I WILL DRAW A WINDOW ON THE RIGHT SIDE OF THE SCREEN, AND THEN
00910 PRINT"I'LL LAUNCH A BRICK FROM THE LEFT SIDE. AT THE EXACT TIME THAT
00920 PRINT"THE WINDOW IS HIT, PRESS ANY KEY (AND WAIT TO ACCEPT YOUR"
00930 PRINT"PUNISHMENT...)."
00970 PRINT:PRINT"THE ONLY REAL DIFFICULTY IS THAT THE BIONIC BRICK SPINS
00980 PRINT"SUPER-SPEED AND TURNS INVISIBLE HALFWAY ACROSS THE SCREEN."
00990 PRINT:FORI=1TO1500:NEXTI
01000 INPUT"PRESS 'ENTER' TO START";A$
01010 RETURN
```

PINBALL LEVEL I

```

00100 REM TRS-80 PINBALL BY PAUL KAFOURE
01000 REM INSTRUCTIONS
01100 CLS: PRINT "WELCOME TO TRS-80 PINBALL": PRINT ""
01105 INPUT "DO YOU WANT INSTRUCTIONS";E
01110 IF E = N GOTO 2000
01120 P."A TARGET AREA WILL BE DRAWN ACROSS THE TOP OF THE SCREEN."
01130 P."POINTS ARE SCORED BY HITTING THE TARGET AREA WITH A"
01140 P."BALL BOUNCED OFF THE PADDLE. MOVE THE PADDLE LEFT BY"
01150 P."HITTING THE BACKSPACE (]) KEY. MOVE IT RIGHT BY HITTING"
01160 P."THE SPACE BAR. USE THE BACKSPACE TO STOP A PADDLE WHICH"
01170 P."IS MOVING RIGHT AND THE SPACE KEY TO STOP A PADDLE MOVING"
01180 P."LEFT. THERE WILL BE 10 BALLS WITH BONUSES FOR HIGH SCORES."
01900 INPUT "HIT ENTER WHEN YOU'RE READY";A$
02000 REM INITIALIZE
02010 A = 44 :REM Y CO-ORDINATE OF PADDLE
02020 B = 10 :REM MAXIMUM NUMBER OF BALLS
02030 C = 65 :REM CENTER OF PADDLE AT START
02040 D = 0 :REM MOVEMENT OF PADDLE (0 = STOP, 1 = RIGHT,
02050 REM -1 = LEFT)
02060 E = 0 :REM INPUT ANSWER TO YES/NO QUESTIONS
02070 F = 39 :REM LEFT MOVE/STOP SET POINT X CO-ORDINATE
02080 G = 42 :REM RIGHT MOVE/STOP SET POINT X CO-ORDINATE
02090 H = 60 :REM HORIZONTAL X CO-ORDINATE OF BALL AT START
02100 I = 0 :REM USED IN LOOPS
02110 J = 0 :REM USED IN LOOPS
02120 K = 2 :REM HORIZONTAL ADJUSTMENT OF BALL (+RIGHT,-LEFT)
02130 L = 2 :REM VERTICAL ADJUSTMENT OF BALL (+=DOWN,-=UP)
02140 M = 20 :REM CURSOR POSITION BETWEEN TWO SET POINTS
02150 N = 2 :REM "NO" VALUE FOR YES/NO REPLIES
02160 P = 20 :REM PADDLE WIDTH
02170 Q = 10 :REM END OF PADDLE
02180 R = 7 :REM USED AS ARGUMENT IN RANDOM FUNCTION
02190 S = 0 :REM SCORE
02200 T = 10 :REM USED IN LOOP TO MOVE PADDLE BEFORE NEW BALL
02210 U = 1 :REM BALL COUNTER
02220 V = 31 :REM VERTICAL Y CO-ORDINATE OF BALL AT START
02230 W = 0 :REM USED AS COUNTER WITH T IN NEW BALL LOOP
02240 X = 0 :REM X CO-ORDINATE IN MOVING PADDLE
02250 Y = 1 :REM "YES" VALUE FOR YES/NO REPLIES
02260 Z = 0 :REM Y CO-ORDINATE IN MOVING PADDLE
03000 REM DRAW PATTERN
03100 CLS
03150 FOR J = 4 TO 28
03200 FOR I = 0 TO 127
03220 SET (I,J): NEXT I: NEXT J
03300 FOR I = 55 TO 55 + P
03310 SET (I,A)
03320 NEXT I
03400 FOR J = 30 TO 47

```

```

03410 SET(0,J): SET(127,J)
03420 NEXT J
04000 REM NEXT BALL
04100 FOR U = U TO B
04150 PRINT AT 30, "SCORE ";S;: PRINT AT 50, "BALL ";U;
04160 PRINT AT M, "": SET (F,Y): SET (G,Y)
04190 SET (H,V): SET (H+1,V)
05000 REM MOVE BALL
05050 RESET(H,V): RESET(H+1,V)
05070 IF V > 44 GOTO 6950
05100 H = H + K: V = V + L
05150 IF H < 2 THEN H = 2: K = -K + 1
05160 IF H > 124 THEN H = 124: K = -K - 1
05180 IF V < 9 THEN V = 9: L = -L
05200 SET(H,V): SET(H+1,V)
05210 IF V < G GOTO 5270
05250 IF(V>42)*(H<=C+Q)*(H>=C-Q-1) L=-L: K=INT((C-H)/3)
05270 IF V > 29 GOTO 5900
05300 IF (POINT(H,V-1)=0) * (POINT(H+1,V-1)=0) GOTO 5900
05350 S = S + 10
05400 FOR I = H-1 TO H+2: FOR J = V-4 TO V-1
05410 RESET (I,J): N.J:N.I
05440 IF (K = 0) + (L > 0) THEN K = -K + RND(3) - 2
05450 L = ABS(L)
05900 REM END OF BALL MOVEMENT
06000 REM MOVE PADDLE
06100 IF POINT(G,Y) = 0 THEN D = D + 1: GOTO 6120
06110 IF POINT(F,Y) = 0 THEN D = D - 1
06120 PRINT AT M, "": SET(F,Y): SET(G,Y)
06150 IF D = 0 GOTO 5000
06200 D = D / ABS(D)
06250 X = C + 11*D: Z = C - 10*D
06800 SET(X,A): RESET(Z,A): SET(X+D,A): RESET(Z+D,A)
06810 SET(X+D+D,A): RESET(Z+D+D,A)
06840 C = C + D + D + D
06850 IF C = 113 THEN D = -1: GOTO 6250
06860 IF C = 14 THEN D = 1: GOTO 6250
06900 G.5000
06950 IF W < T THEN W = W + 1: V = 45: GOTO 6000
06970 W = 0: V = 31
06980 NEXT U: REM NEXT BALL
06990 IF(B=10)*(S>499)B=B+INT(S/500): G.4000
07000 PRINT AT 15, "END OF GAME SCORE ";S;
09000 REM NEW GAME
09100 PRINT AT 770, " ";
09110 INPUT " DO YOU WANT ANOTHER GAME";E
09115 IF E = N PRINT " OK, BYE": GOTO 9990
09140 GOTO 1000
09990 GOTO 9990

```

PINBALL LEVEL II

```

00100  REM TRS-80 PINBALL BY PAUL KAFOURE
01000  REM INSTRUCTIONS
01100  CLS: PRINT "WELCOME TO TRS-80 PINBALL": PRINT ""
01105  INPUT "DO YOU WANT INSTRUCTIONS";E$
01110  IFLEFT$(E$,1) = "N" GOTO 2000
01120  PRINT"A TARGET AREA WILL BE DRAWN ACROSS THE TOP OF THE SCREEN."
01130  PRINT"POINTS ARE SCORED BY HITTING THE TARGET AREA WITH A"
01140  PRINT"BALL BOUNCED OFF THE PADDLE. MOVE THE PADDLE LEFT BY"
01150  PRINT"HITTING THE BACKSPACE (]) KEY. MOVE IT RIGHT BY HITTING"
01160  PRINT"THE SPACE BAR. USE THE BACKSPACE TO STOP A PADDLE WHICH"
01170  PRINT"IS MOVING RIGHT AND THE SPACE KEY TO STOP A PADDLE MOVING"
01180  PRINT"LEFT. THERE WILL BE 10 BALLS WITH BONUSES FOR HIGH SCORES."
01900  INPUT "HIT ENTER WHEN YOU'RE READY";A$
02000  REM INITIALIZE
02010  A = 44 :REM Y CO-ORDINATE OF PADDLE
02020  B = 10 :REM MAXIMUM NUMBER OF BALLS
02030  C = 65 :REM CENTER OF PADDLE AT START
02040  D = 0 :REM MOVEMENT OF PADDLE (0 = STOP, 1 = RIGHT,
02050  REM -1 = LEFT)
02060  E = 0 :REM INPUT ANSWER TO YES/NO QUESTIONS
02070  F = 39 :REM LEFT MOVE/STOP SET POINT X CO-ORDINATE
02080  G = 42 :REM RIGHT MOVE/STOP SET POINT X CO-ORDINATE
02090  H = 60 :REM HORIZONTAL X CO-ORDINATE OF BALL AT START
02100  I = 0 :REM USED IN LOOPS
02110  J = 0 :REM USED IN LOOPS
02120  K = 2 :REM HORIZONTAL ADJUSTMENT OF BALL (+RIGHT,-LEFT)
02130  L = 2 :REM VERTICAL ADJUSTMENT OF BALL (+=DOWN, -=UP)
02140  M = 20 :REM CURSOR POSITION BETWEEN TWO SET POINTS
02150  N = 2 :REM "NO" VALUE FOR YES/NO REPLIES
02160  P = 20 :REM PADDLE WIDTH
02170  Q = 10 :REM END OF PADDLE
02180  R = 7 :REM USED AS ARGUMENT IN RANDOM FUNCTION
02190  S = 0 :REM SCORE
02200  T = 10 :REM USED IN LOOP TO MOVE PADDLE BEFORE NEW BALL
02210  U = 1 :REM BALL COUNTER
02220  V = 31 :REM VERTICAL Y CO-ORDINATE OF BALL AT START
02230  W = 0 :REM USED AS COUNTER WITH T IN NEW BALL LOOP
02240  X = 0 :REM X CO-ORDINATE IN MOVING PADDLE
02250  Y = 1 :REM "YES" VALUE FOR YES/NO REPLIES
02260  Z = 0 :REM Y CO-ORDINATE IN MOVING PADDLE
03000  REM DRAW PATTERN
03100  CLS
03150  FOR J = 4 TO 28
03200  FOR I = 0 TO 127
03220  SET (I,J): NEXT I: NEXT J
03300  FOR I = 55 TO 55 + P
03310  SET (I,A)
03320  NEXT I
03400  FOR J = 30 TO 47
03410  SET(0,J): SET(127,J)
03420  NEXT J

```

```

04000 REM NEXT BALL
04100 FOR U = U TO B
04150 PRINT@ 30, "SCORE ";S;: PRINT@ 50, "BALL ";U;
04160 PRINT@M, " ";: SET (F,Y): SET (G,Y)
04190 SET (H,V): SET (H+1,V)
05000 REM MOVE BALL
05050 RESET(H,V): RESET(H+1,V)
05070 IF V > 44 GOTO 6950
05100 H = H + K: V = V + L
05150 IF H < 2 THEN H = 2: K = -K + 1
05160 IF H > 124 THEN H = 124: K = -K - 1
05180 IF V < 9 THEN V = 9: L = -L
05200 SET(H,V): SET(H+1,V)
05210 IF V < G GOTO 5270
05250 IFV>42ANDH<=C+QANDH>=C-Q-1L=-L: K=INT((C-H)/3)
05270 IF V > 29 GOTO 5900
05300 IF POINT(H,V-1)=0ANDPOINT(H+1,V-1)=0 GOTO 5900
05350 S = S + 10
05400 FOR I = H-1 TO H+2: FOR J = V-4 TO V-1
05410 RESET (I,J): NEXTJ:NEXTI
05440 IF (K = 0) + (L > 0) THEN K = -K + RND(3) - 2
05450 L = ABS(L)
05900 REM END OF BALL MOVEMENT
06000 REM MOVE PADDLE
06050 A$ = INKEY$: IF A$ = "" THEN 6120
06070 IF A$ = " " THEN D = D + 1 :ELSE D = D - 1
06120 PRINT@ M, " ";: SET (F,Y): SET (G,Y)
06150 IF D = 0 GOTO 5000
06200 D = D / ABS(D)
06250 X = C + 11*D: Z = C - 10*D
06800 SET(X,A): RESET(Z,A): SET(X+D,A): RESET(Z+D,A)
06810 SET(X+D+D,A): RESET(Z+D+D,A)
06840 C = C + D + D + D
06850 IF C = 113 THEN D = -1: GOTO 6250
06860 IF C = 14 THEN D = 1: GOTO 6250
06900 GOTO5000
06950 IF W < T THEN W = W + 1: V = 45: GOTO 6000
06970 W = 0: V = 31
06980 NEXT U: REM NEXT BALL
06990 IFB=10ANDS>499B=B+INT(S/500): GOTO4000
07000 PRINT@ 15, "END OF GAME SCORE ";S;
09000 REM NEW GAME
09100 PRINT@ 770, " ";
09110 INPUT " DO YOU WANT ANOTHER GAME";E$
09115 IF LEFT$(E$,1) = "N" PRINT " OK, BYE": GOTO 9990
09140 GOTO 1000
09990 GOTO 9990

```

JUKEBOX LEVEL I

```

00001 U=.6:G.310
00010 F.K=1TO12*L
00011 A=12-12+1
00012 N.K:RET.
00015 F.K=1TO13*L
00016 A=A-1+10
00017 N.K:RET.
00020 F.K=1TO14*L
00021 A=A-A+123456
00022 N.K:RET.
00025 F.K=1TO15*L
00026 A=A-A+12
00027 N.K:RET.
00030 F.K=1TO16*L
00031 A=1+1
00032 N.K:RET.
00040 F.K=1TO17*L
00041 A=A+111
00042 N.K:RET.
00045 F.K=1TO18*L
00046 A=00000000
00047 N.K:RET.
00050 F.K=1TO19*L
00051 A=A-A
00052 N.K:RET.
00055 F.K=1TO20*L
00056 A=1111
00057 N.K:RET.
00060 F.K=1TO21*L
00061 A=11:N.K:RET.
00065 F.K=1TO22*L:A=1:N.K:RET.
00070 F.K=1TO23*L
00071 A=A
00072 N.K:RET.
00110 F.K=1TO24*L:A$="":REM
00112 N.K:RET.
00115 F.K=1TO26*L
00116
00117 A$=COUNTERPOINT
00120 F.K=1TO28*L
00121 A$=CLOAD *** MAGAZINE
00122 N.K:RET.
00130 F.K=1TO32*L
00131 A$=E
00132 N.K:RET.
00140 F.K=1TO34*L:A$=
00141 N.K:RET.
00310 V=U+U:W=V+V:X=W+W:Y=X+X:S=U+V:T=V+W:Q=W+X
00320 C.:P.AT970;"(PLACE AM-RADIO NEAR SPACE-BAR ON KEYBOARD)";
00400 P.AT84;"TRS-80 JUKEBOX
00410 P.:P."1) MICHAEL ROW THE BOAT ASHORE
00420 P."2) MARINES HYMN":P."3) CLEMENTINE":P."4) OH SUSANNAH

```

```

00430 P-"5) SILENT NIGHT":P."6) THE SOUND OF SILENCE
00490 P.:I."NUMBER";A:P.AT640
00495 IF(A<1)+(A>6)T.320
00500 P-AT160+64*A;"] ] ] NOW PLAYING";
00510 ONAGOS.999,2000,3000,4000,5000,6000:G.320
00999 F-R=1T02
01010 L=W:GOS.10:GOS.30:L=T:GOS.50:L=V:GOS.30:GOS.50:L=T:GOS.60
01020 L=X:GOS.50:L=W:GOS.30:GOS.50:L=Y:GOS.60:L=X:GOS.50
01030 L=V:GOS.30:L=T:GOS.50:GOS.50:L=V:GOS.30:GOS.40:L=T:GOS.30
01040 L=X:GOS.20:L=W:GOS.10:GOS.20:L=X:GOS.30:GOS.20:GOS.10:N.R
01050 L=V:GOS.15:L=T:GOS.40:GOS.55:L=V:GOS.40:GOS.55:L=T:GOS.65
01060 L=X:GOS.55:L=W:GOS.115:GOS.110:L=Y:GOS.65:L=X:GOS.55
01100 L=V:GOS.40:L=V+W:GOS.55:GOS.55:L=V:GOS.40:GOS.45
01110 L=T:GOS.40:L=X:GOS.25:L=W:GOS.15:GOS.25
01120 GOS.40:GOS.55:L=V:GOS.65:L=T:GOS.110:L=X:GOS.115:RET.
02000 F-R=1T04:ONRGOS.2010,2010,2050,2010:N.R:RET.
02010 L=S:GOS.10:L=U:GOS.30:L=W:F.I=1T04:GOS.50:N.I.
02020 L=T:GOS.50:L=V:GOS.110:L=W:GOS.50
02030 L=S:GOS.30:L=U:GOS.40:L=W:GOS.50:GOS.50:L=V:GOS.40:L=T
02040 GOS.20:L=W+X:GOS.10:RET.
02050 L=S:GOS.110:L=U:GOS.70:L=W:GOS.60:GOS.40:GOS.60:GOS.110
02060 L=T:GOS.50:L=V:GOS.30:L=W:GOS.50
02070 L=S:GOS.110:L=U:GOS.70:L=W:GOS.60:GOS.40
02080 L=V:GOS.60:L=T:GOS.110:L=W+X:GOS.50:RET.
03000 F-R=1T02:L=S:GOS.40:L=U:GOS.40:L=W:GOS.40:GOS.10:L=S:GOS.60
03020 L=U:GOS.60:L=W:GOS.60:GOS.40:L=S:GOS.40:L=U:GOS.60
03030 L=T:GOS.110:L=V:GOS.110:L=S:GOS.65:L=U:GOS.60:L=X:GOS.50
03040 L=S:GOS.50:L=U:GOS.60:L=W:GOS.65:GOS.65:L=S:GOS.60:L=U
03050 GOS.50:L=W:GOS.60:GOS.40:L=S:GOS.40:L=U:GOS.60:L=T:GOS.50
03060 L=V:GOS.10:L=S:GOS.30:L=U:GOS.50:L=X:GOS.40:N.R:RET.
04000 F-R=1T07:ONRGOS.4010,4040,4010,4050,4060,4010,4050:N,R:RET.
04010 L=U:GOS.50:GOS.60:L=V:GOS.70:GOS.120
04020 L=S:GOS.120:L=U:GOS.130:L=V:GOS.120:GOS.70
04030 L=S:GOS.50:L=U:GOS.60:L=V:GOS.70:GOS.70
04035 L=(U+V)/2:GOS.60:L=V:RET.
04040 GOS.50:L=T:GOS.60:RET.
04050 GOS.60:L=X:GOS.50:RET.
04060 L=W:GOS.110:GOS.110:L=V:GOS.130:L=W:GOS.130:L=V:F.N=1T05
04070 ONNGOS.130,120,120,70,50:N.N:L=T:GOS.60:RET.
05000 F-R=1T02:L=T:GOS.50:L=V:GOS.60:L=W:GOS.50:L=Q:GOS.30:N.R
05020 L=X:GOS.120:L=W:GOS.120:L=Q:GOS.70:L=X:GOS.110:L=W
05030 GOS.110:L=Q:GOS.50:F.R=1T02:L=X:GOS.60:L=W:GOS.60:L=T
05040 GOS.110:L=V:GOS.70:L=W:GOS.60:L=T:GOS.50:L=V:GOS.60:L=W
05050 GOS.50:L=Q:GOS.30:N.R:L=X:GOS.120:L=W:GOS.120:L=T:GOS.140
05060 L=V:GOS.120:L=W:GOS.70:L=Q:GOS.110:GOS.130:L=W:GOS.110
05070 GOS.50:GOS.30:L=T:GOS.50:L=V:GOS.40:L=W:GOS.20
05080 L=Q+X:GOS.10:RET.
06000 F.P=1T02:L=V:F.N=1T06:ONNGOS.20,20,40,40,60,60:N.N:L=Y:GOS.50
06030 L=V:F.N=1T08:ONNGOS.10,10,10,30,30,50,50,40:N.N:L=Y-V
06040 GOS.20:F.R=1T02:L=V:F.N=1T07:ONNGOS.40,40,40,60,60,110,110
06050 N.N:L=W:GOS.120:L=V:GOS.120:L=Q:GOS.110:N.R
06060 L=V:GOS.40:GOS.40:GOS.120:L=T+W:GOS.120:L=V:GOS.120:GOS.130
06070 GOS.140:L=T:GOS.140:L=V:GOS.130:L=T:GOS.120:L=Q:GOS.110
06080 L=V:GOS.120:GOS.110:L=Y:GOS.60:L=V:GOS.40:GOS.40:GOS.40:L=Q
06090 GOS.110:L=V:GOS.30:GOS.40:L=Y:GOS.20:N.P:RET.

```

```
00004 DEFINTA-T:DEFSNGL:GOTO310
00005 JAMES GARON:920 W. ROMNEYA #6 ANAHEIM, CA 92801
00006 FORK=1TO11.5*L
00007 ::::REM
00008 ::::REM
00009 NEXT:RETURN
00010 FORK=1TO12*L
00011 ::::REM
00012 :::REM//
00013 NEXT:RETURN
00015 FORK=1TO13*L
00016 :::REM//
00017 :::REM///
00018 NEXT:RETURN
00020 FORK=1TO14*L
00021 :::REM/////
00022 :::REM/////
00023 NEXT:RETURN
00025 FORK=1TO15*L
00026 :::REM/////
00027 :::REM
00028 NEXT:RETURN
00030 FORK=1TO16*L
00031 :::REM////
00032 :::REM////
00033 NEXT:RETURN
00040 FORK=1TO17*L
00041 :::REM
00042 :::REM
00043 NEXT:RETURN
00045 FORK=1TO18*L
00046 :REM////
00047 :::REM////
00048 NEXT:RETURN
00050 FORK=1TO19*L
00051 :REM
00052 :::REM
00053 NEXT:RETURN
00055 FORK=1TO20*L
00056 :REM/////
00057 :REM/////
00058 NEXT:RETURN
00060 FORK=1TO21*L
00061 :REM
00062 :REM
00063 NEXT:RETURN
00065 FORK=1TO22*L
00066 REM////////
00067 :REM////
00068 NEXT:RETURN
00070 FORK=1TO23*L
00071 :REM
00072 REM
00073 NEXT:RETURN
00110 FORK=1TO24*L::::REM//////////
00112 NEXT:RETURN
00115 FORK=1TO26*L::::REM////
00116 NEXT:RETURN
00120 FORK=1TO28*L::::REM//////////
00121 NEXT:RETURN
```

```

00130 FORK=1TO32*L: REM//////////
00131 NEXT:RETURN
00140 FORK=1TO34*L: REM//////////
00141 NEXT:RETURN
00310 U=1.4:V=U+U:W=V+V:X=W+W:Y=X+X:S=U+V:T=V+W:Q=W+X
00320 CLS:PRINT@910,"PLACE A-M RADIO NEAR KEYBOARD":PRINT@20,"TRS-80 JUKF
:PRINT
00410 PRINT:PRINT"1) MICHAEL ROW THE BOAT ASHORE
00420 PRINT"2) MARINES HYMN":PRINT"3) CLEMENTINE":PRINT"4) OH SUSANNAH
00430 PRINT"5) SILENT NIGHT":PRINT"6) THE SOUND OF SILENCE
00500 PRINT:INPUT"NUMBER";A:IFA<1ORA>6THEN320:ELSEPRINT@639,
:PRINT@160+64*A,"[ ] ] NOW PLAYING";
00510 ONAGOSUB999,2000,3000,4000,5000,6000:GOTO320
00999 FORR=1TO2:L=W:GOSUB10:GOSUB30:L=T:GOSUB50:L=V:GOSUB30:GOSUB50:L=T
:GOSUB60
01020 L=X:GOSUB50:L=W:GOSUB30:GOSUB50:L=Y:GOSUB60:L=X:GOSUB50
01030 L=V:GOSUB30:L=T:GOSUB50:GOSUB50:L=V:GOSUB30:GOSUB40:L=T:GOSUB30
01040 L=X:GOSUB20:L=W:GOSUB10:GOSUB20:L=X:GOSUB30:GOSUB20:GOSUB10:NEXTR
01050 L=V:GOSUB15:L=T:GOSUB40:GOSUB55:L=V:GOSUB40:GOSUB55:L=T:GOSUB65
01060 L=X:GOSUB55:L=W:GOSUB115:GOSUB110:L=Y:GOSUB65:L=X:GOSUB55
01100 L=V:GOSUB40:L=V+W:GOSUB55:GOSUB55:L=V:GOSUB40:GOSUB45
01110 L=T:GOSUB40:L=X:GOSUB25:L=W:GOSUB15:GOSUB25
01120 GOSUB40:GOSUB55:L=V:GOSUB65:L=T:GOSUB110:L=X:GOSUB115:RETURN
02000 FORR=1TO4:ONRGOSUB2010,2010,2050,2010:NEXTR:REM
02010 L=S:GOSUB10:L=U:GOSUB30:L=W:FORI=1TO4:GOSUB50:NEXTI
02020 L=T:GOSUB50:L=V:GOSUB110:L=W:GOSUB50
02030 L=S:GOSUB30:L=U:GOSUB40:L=W:GOSUB50:GOSUB50:L=V:GOSUB40:L=T
02040 GOSUB20:L=W+X:GOSUB10:RETURN
02050 L=S:GOSUB110:L=U:GOSUB70:L=W:GOSUB60:GOSUB40:GOSUB60:GOSUB110
02060 L=T:GOSUB50:L=V:GOSUB30:L=W:GOSUB50
02070 L=S:GOSUB110:L=U:GOSUB70:L=W:GOSUB60:GOSUB40
02080 L=V:GOSUB60:L=T:GOSUB110:L=W+X:GOSUB50:RETURN
03000 FORR=1TO2:L=S:GOSUB40:L=U:GOSUB40:L=W:GOSUB40:GOSUB10:L=S:GOSUB60
03020 L=U:GOSUB60:L=W:GOSUB60:GOSUB40:L=S:GOSUB40:L=U:GOSUB60
03030 L=T:GOSUB110:L=V:GOSUB110:L=S:GOSUB65:L=U:GOSUB60:L=X:GOSUB50
03040 L=S:GOSUB50:L=U:GOSUB60:L=W:GOSUB65:GOSUB65:L=S:GOSUB60:L=U
03050 GOSUB50:L=W:GOSUB60:GOSUB40:L=S:GOSUB40:L=U:GOSUB60:L=T:GOSUB50
03060 L=V:GOSUB10:L=S:GOSUB30:L=U:GOSUB50:L=X:GOSUB40:NEXTR:RETURN
04000 FORR=1TO7:ONRGOSUB4010,4040,4010,4050,4060,4010,4050:NEXTR:RETURN
04010 L=U:GOSUB50:GOSUB60:L=V:GOSUB70:GOSUB120
04020 L=S:GOSUB120:L=U:GOSUB130:L=V:GOSUB120:GOSUB70
04030 L=S:GOSUB50:L=U:GOSUB60:L=V:GOSUB70:GOSUB70
04035 L=(U+V)/2:GOSUB60:L=V:RETURN
04040 GOSUB50:L=T:GOSUB60:RETURN
04050 GOSUB60:L=X:GOSUB50:RETURN
04060 L=W:GOSUB110:GOSUB110:L=V:GOSUB130:L=W:GOSUB130:L=V:FORN=1TO5
04070 ONNGOSUB130,120,120,70,50:NEXTN:L=T:GOSUB60:RETURN
05000 FORR=1TO2:L=T:GOSUB50:L=V:GOSUB60:L=W:GOSUB50:L=Q:GOSUB30:NEXTR
05020 L=X:GOSUB120:L=W:GOSUB120:L=Q:GOSUB70:L=X:GOSUB110:L=W
05030 GOSUB110:L=Q:GOSUB50:FORN=1TO2:L=X:GOSUB60:L=W:GOSUB60:L=T
05040 GOSUB110:L=V:GOSUB70:L=W:GOSUB60:L=T:GOSUB50:L=V:GOSUB60:L=W
05050 GOSUB50:L=Q:GOSUB30:NEXTR:L=X:GOSUB120:L=W:GOSUB120:L=T:GOSUB110
05060 L=V:GOSUB120:L=W:GOSUB70:L=Q:GOSUB110:GOSUB130:L=W:GOSUB110
05070 GOSUB50:GOSUB30:L=T:GOSUB50:L=V:GOSUB40:L=W:GOSUB20
05080 L=Q+X:GOSUB10:RETURN
06000 L=V:FORN=1TO6:ONNGOSUB20,20,40,40,60,60:NEXTN:L=Y:GOSUB50
06030 L=V:FORN=1TO8:ONNGOSUB10,10,10,30,30,50,50,40:NEXTN:L=Y-V
06040 GOSUB20:FORN=1TO2:L=V:FORN=1TO7:ONNGOSUB40,40,40,60,60,110,110
06050 NEXTN:L=W:GOSUB120:L=V:GOSUB120:L=Q:GOSUB110:NEXTR
06060 L=V:GOSUB40:GOSUB40:GOSUB120:L=T+W:GOSUB120:L=V:GOSUB120:GOSUB130
06070 GOSUB140:L=T:GOSUB140:L=V:GOSUB130:L=T:GOSUB120:L=Q:GOSUB110
06080 L=V:GOSUB120:GOSUB110:L=Y:GOSUB60:L=V:GOSUB40:GOSUB40:GOSUB40:L=Q
06090 GOSUB110:L=V:GOSUB30:GOSUB40:L=Y:GOSUB20:RETURN

```