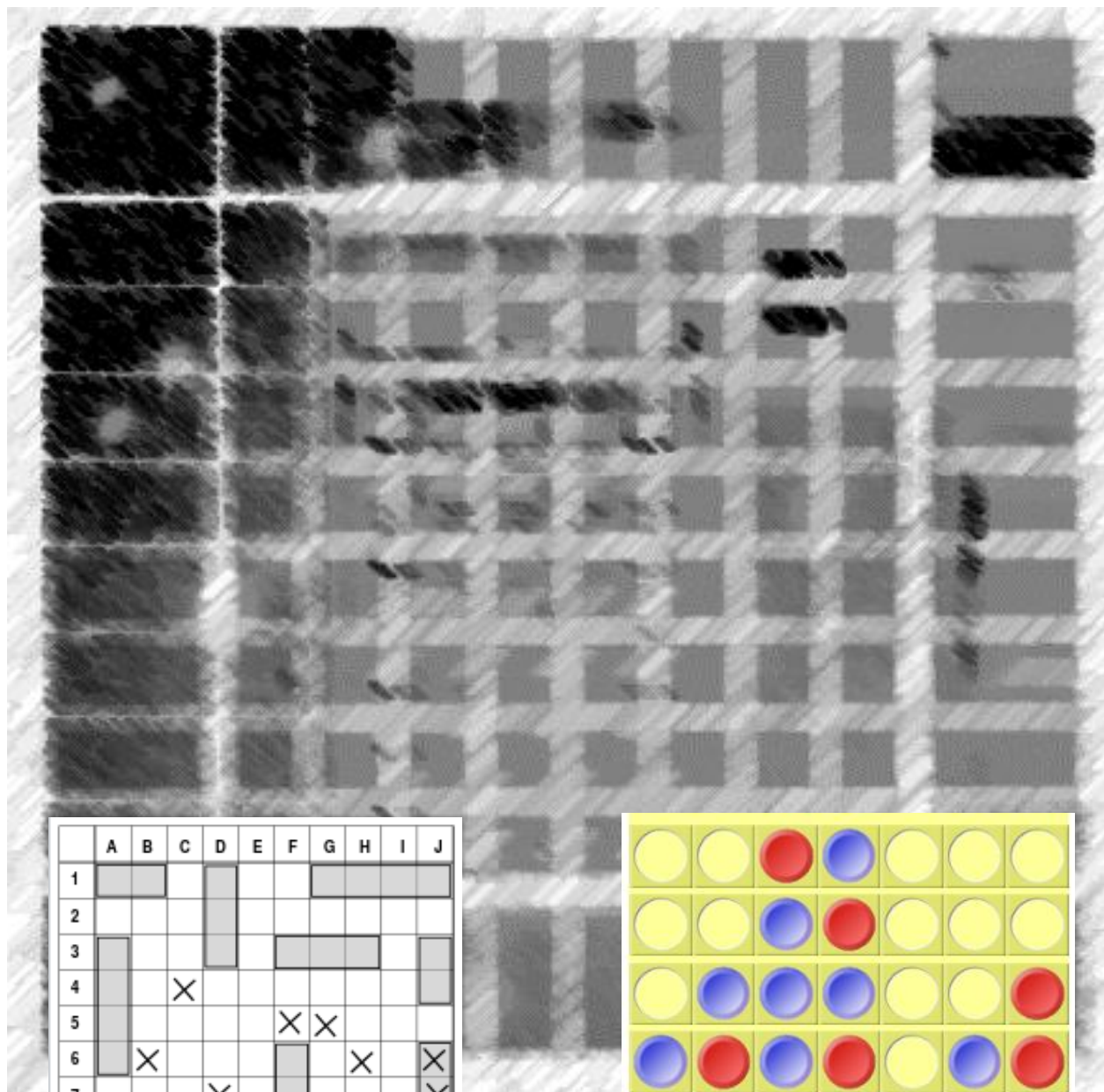
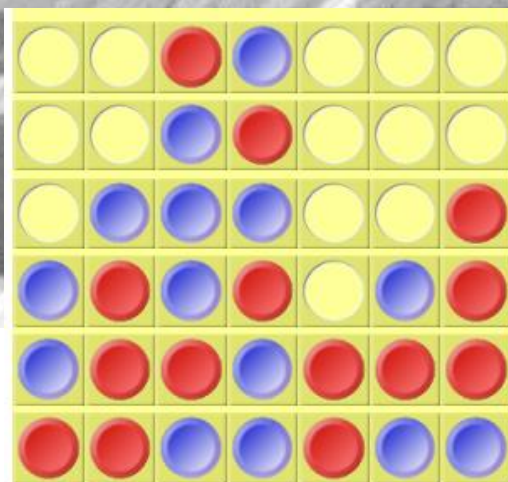


VINTAGE GAMES - VOLUME 2

COLLECTION FOR THE HP-41



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4			X							
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										



Compiled by Ángel M. Martín

June 2019

This compilation revision 1.2.2

Copyright © 2017-19 Ángel Martin

Published under the *GNU* software license agreement.

Original authors retain all copyrights and should be mentioned in writing by any part utilizing this material. No commercial usage of any kind is allowed.

Screen captures taken from V41, Windows-based emulator developed by Warren Furlow.
See www.hp41.org

Introduction.

This compilation includes a large amount of information on the subject of Games for the HP-41. Most of the games described here (and then some more!) are included in one of the Games Modules available in the CL Library; you can refer to the CL Modules Reference section for a relationship of the specific programs of interest.

The sources of the material included are very diverse: HP Museum, hp41.org Archive, PPC Calculator Journal, Data File Issues, Prisma Magazine, HP User program Library, Swap Disks (that huge heap of mostly undocumented stuff...), and finally different individual contributors' web sites. Collecting all this material hasn't been trivial, and necessarily introduces small inconsistencies in form and structure – but nevertheless the descriptions should be sufficient for a working implementation of the games.

The compilation is divided into three main sections, plus a final CL module reference as detailed below. You can use the hyperlinks to go directly to your area of interest, or use the index in the next pages to access the individual program.

1. [MCODE Games](#)
2. [FOCAL Games](#)
3. Adventure Games
4. CL Modules Reference

With the exception of the Adventure games (a class on their own for obvious reasons), there's no category, subject, topic or other criteria used to structure the compilation (Brain teasers, Casino, Puzzles, Board, Mazes, Simulations, etc). Broadly speaking, the games are structured in a "from more simple to more complex" sequence. This of course is not the same as "from worse to better", as this is largely a personal choice. Some of the most enjoyable games are small in size but require quick reflexes; and conversely some of the more elaborate simulations are a tad too long and the player may become desinterested after a few runs.

The way original authors documented their programs varies immensely, some are really minimalistic in the sketches, whereas others use a more verbose description of the game instructions with prolific program details.

So all in all, here you have a comprehensive representation of the Games on the HP-41 platform that can provide many days of enjoyment to the user. At the very least, it should contribute by providing easy access to a ton of documentation only available using obscure sources difficult to locate.

Table of Contents (in alphabetical order)

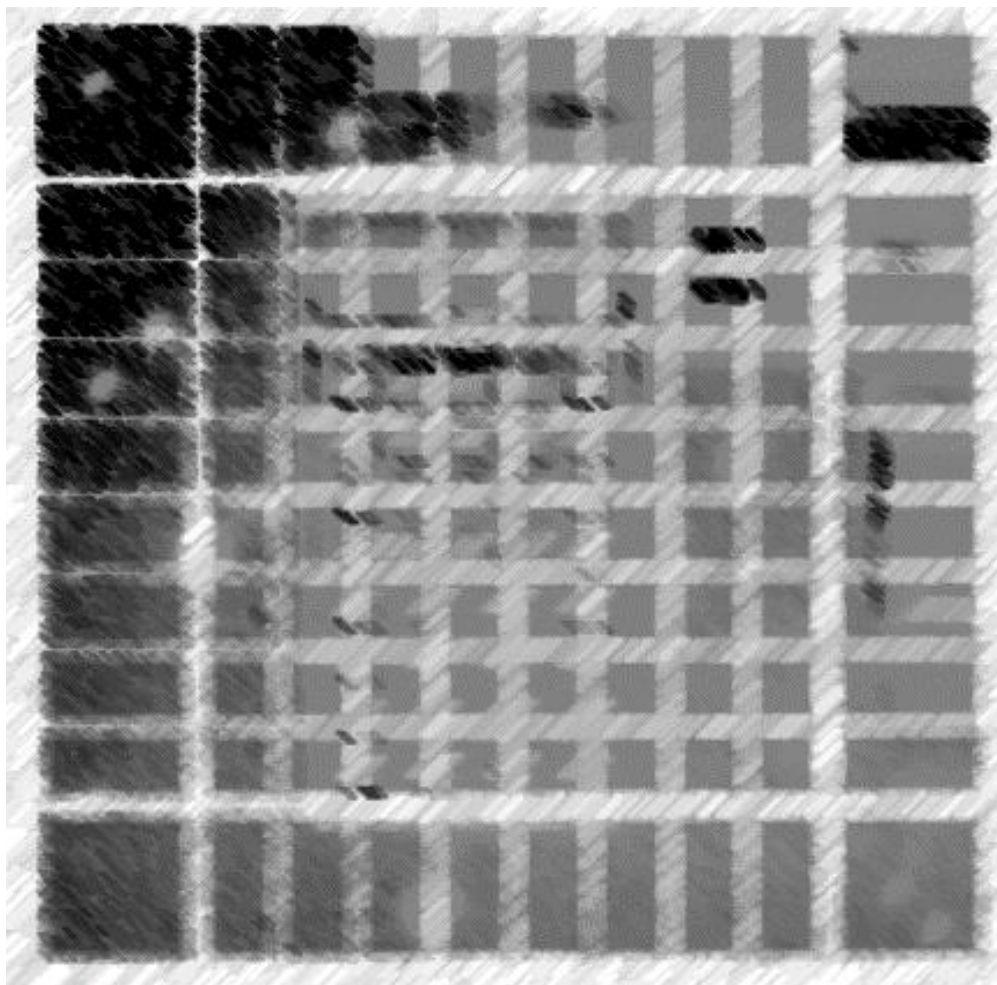
Part II : FOCAL Games

- 1 [3D Tic-Tac-Toe](#) – HP Co.; Games Solutions Books
- 2 [3-Ways Game](#) - L. Stein; PPC V9N4 p58
- 3 [5x5 Mini-Chess](#) – Valentín Albillo; PPCCJ V8N6 p66
- 4 [American Roulette](#) – JM Baillard; Web resource
- 5 [Black Jack](#) – Whodunit; Swap Disks
- 6 [BlackJack \(Twenty-One\)](#) – HP Co.; Standard Apps.
- 7 [Blackjack Card Counter](#) - Richard Baker; PPCCJ V7N7 p6
- 8 [Black Jack for the HP-41](#) – JM Baillard; Web resource
- 9 [Car Race](#) – Martin Meyer, Prisma 3/89 p29
- 10 [Checkers 3.0](#) – Valentín Albillo; PPCCJ V9N2 p20
- 11 [Catch the Goufers](#) – Thomas Fange; PPCCJ V8N6 p44
- 12 [Code Crack](#) – HP Co.; Games Solutions Books
- 13 [Connect Four](#) - Kai Schröder, Web resource
- 14 [Craps](#) – HP Co.; Games Pac
- 15 [Craps.v2](#) – Samuel & Spencer Hartmann; PPCCJ V4N13 p27
- 16 [Cybernetic Hexapawn](#) – Marco de Vries; PPCCJ V8N1 p26
- 17 [Digit SWAP \(Reversal\)](#) - Mark Power; DataFile V8N5 p15
- 18 [Domino](#) – Whodunit; Swap disks
- 19 [Eighth Passenger](#)– ‘Archilog’; MoHP Forum
- 20 [Fight Gun Duel](#) - Ruys Dirk; DataFile V1N3 p26
- 21 [Flip-Flop](#) – HP Co.; Games Solutions Books
- 22 [FOURS, Virtually Connected](#) – David Kipling; DataFile V2N4 p5
- 23 [Fruit Machine](#)– Brian Ward; DataFile V6N6 p39
- 24 [Galaxis Game](#) – ‘Heimchr’, MoHP Forum
- 25 [Gambler](#) - Neil G. Jarman - DataFile V3N2 p20
- 26 [Gork \(Foxhole Grenades\)](#) - Philip T. Frohme; PPCCJ V12N8 p11
- 27 [Grand Master Mind](#) – Tom Rice; PPCCJ V12N3 p10
- 28 [Hangman \(Word Guessing\)](#) – HP Co.; Standard Apps
- 29 [Hangman](#) – HP Co.; Games Pac
- 30 [Hangman \(SP\)](#) – William C. Wickes – PPCCJ V7N4 p31
- 31 [Hangman \(French\)](#) – Whodunit; Swap Disks
- 32 [Hangman \(w/ Subroutines\)](#) – John Rausch; PPCCJ V7N2 p43
- 33 [Hangman w/ X-Functions](#) - Julian Perry – DataFile V1N1 p5

- 34 [Health Check](#) – Whodunit; Swap Disks
- 35 [Herbie the Hippo](#) – Tomas Rodke, hp41.org
- 36 [Hexapawn](#)– HP Co. Games Solutions Books
- 37 [Hexapawn v2](#)– P. Hamer /E. Hartingsveldt; PPCCJ V7N4 p31
- 38 [High Rollers, FOCAL Version](#)– Randal C. Gibson; PPCCJ V6N7 p47
- 39 [Hi-Lo, a Game for Children](#) – Philip T. Frohme; PPCCJ V12N5 p25
- 40 [Integer Choice](#) – Jakub Tatarkiewicz; PPCCJ V7N9 p24
- 41 [Inverse MasterMind](#) – Kai Schröder; Web Source
- 42 [Kibur \(alpha order\)](#) – Robert Pulluard; L'Ord. de Poche N3 p43
- 43 [Labyrinth \(French\)](#) – Whodunit; Swap disks
- 44 [Lunar Lander](#)– Mark Gessner; PPCCJ V12N3 p26
- 45 [Mah-Jong score keeping](#)– Frans de Vries; DataFile V7N6 p25
- 46 [Mancala \(a Wari Game\)](#)–‘brianddk’; MoHP Forum
- 47 [Master Mind w/ Timer](#) – Julian Perry; DataFile V2N1 p26
- 48 [Master Mind \(The Sequel\)](#) – Peter Gatenby; DataFile V5N7 p12
- 49 [Master Mind v2](#) - Wouter Peters, MoHP Library
- 50 [Master Mind v3](#) - Kai Schröder, Web Source
- 51 [Master Mind \(w/ Colors\)](#) – Whodunit; Swap Disks
- 52 [Master Mind \(w/ Printer\)](#) – Whodunit; Swap Disks
- 53 [Matrix Game](#)– Kai Schröder; Web resource
- 54 [Maze Construction & Play](#) – Layne K. Johnson; UPL #00663C
- 55 [Mazes](#) – JM Baillard; Web resource
- 56 [Mazes w/ X-Memory](#) – Erik Christensen; PPCCJ V10N5 p30
- 57 [Naval Battle](#) – Javier Chamorro Pagani; Electro1 Mag. N1 p106
- 58 [Navy War](#) – Burkhard Oerttel; Sammlung Books
- 59 [NFL American Football](#) – Whodunit; Swap Disks
- 60 [NIM-41](#)– Bob Laughton; Australian Tech Notes V6 p11
- 61 [Orbital Lander](#)–HP Co.; MoHP Library
- 62 [Othello \(Reversi\)](#) – Valentín Albillo; PPCCJ V8N3 p14
- 63 [Petals around the Rose](#) – Edward M. Keefe; UPL #00479C
- 64 [Pilot Flying](#) – Whodunit; Swap Disks
- 65 [Pinball Machine](#)– HP Co.; Games Pac
- 66 [Pinball Wizard](#) – Craig Pearce; Games Solutions Books
- 67 [Planet Lander](#)– HP Co. Games Solutions Books
- 68 [Planet Lander v2](#) – Mark Gessner, PPCCJ V11N9 p39
- 69 [Planet Lander v3 \(French\)](#) – Whodunit; Swap Disks
- 70 [Planet Lander v4](#) – Whodunit; MoHP Disks

- 71 [Poker](#) – JM Baillard; Web resource
- 72 [Robot Trap](#) – HP Co. Games Solutions Books
- 73 [Rummy score keeping](#) - Wolfgang Pawlowsky; Prisma 87.4.12
- 74 [Scatter](#) – HP Co.; Games Solutions Books
- 75 [Sea Battle](#)– Mark Cracknell; Data File V8N6 p9
- 76 [Secret Number](#) – Tom Cadwallader; PPCCJ V11N7 p25
- 77 [Simon](#) – HP Co.; Games Solution Books
- 78 [Simon – Reloaded](#) – Whodunit; MoHP Disks
- 79 [Skunk](#), v1, - Russ Gilbert; MoHP Library
- 80 [Skunk](#), v2 - Gene Wright; Web resource
- 81 [Snakes & Ladders](#) – JM Baillard; Web resource
- 82 [Space Invaders](#)– Ramón Cererols; Boletín Pont-Reyes N1 p26
- 83 [Space Wars Interactive](#) - Roger M. Stenerson – UPL #00655C
- 84 [Sphinx](#) – JM Baillard; Web resource
- 85 [Star Riders](#) – George Ruppert; PPCCJ V11N1 p14
- 86 [Step Game](#) – George Sandoval – UPL #00363C
- 87 [Sub Hunt](#) – HP Co.; Games Pac
- 88 [Sub Hunt, v1](#). – Charles Campbell; PPC V7N4 p15
- 89 [Sub Hunt, v2](#) - James R. Merrill; PPCCJ V8N2 p17
- 90 [Sub-Hunt, v3](#) - Brian Steel; DataFile V2N3 p7
- 91 [Submarine Hunt, v4](#) – Gary Goodman; UPL #02864C
- 92 [Sub Hunt, v5](#) – Whodunit; MoHP Disks
- 93 [Sudoku Grid Generation](#) – JM Baillard; Web resource
- 94 [Sudoku Solver \(FOCAL\)](#) – JM Baillard; Web resource
- 95 [Super Bagels](#) – HP Co.; Games Pac
- 96 [Super Detective](#)– Tom Langland; PPCCJ V12N8 p2
- 97 [Super Master Mind](#) – Whodunit, Swap Disks
- 98 [Target, a War Game](#)– Mark Gessner, PPCCJ V11N9 p38
- 99 [Tic-Tac-Toe](#) – N. Michael Johnson; UPL #00948C
- 100 [Truck Routes](#) – Kenneth Sharp; Games Solutions Book
- 101 [True Battleship](#) – Luis Gasperini; UPL #00796C
- 102 [Wari](#)- HP Co., Games Solution Books
- 103 [Word Game](#)– Mark Gessner; PPCCJ V12N11 p20
- 104 [XOR Game](#) – Dejan Ristanovic; DataFile V2N1 p18
- 105 [Yahtzee](#), - Gene Wright; PPCCJ V12N5 p39
- 106 [Yams](#), a Yahtzee Game – JM Baillard; Web resource
- 107 [Z'Car](#) – An action game – Cary E. Reinstein; PPCCJ V11N1 p20

Part II – FOCAL GAMES



Herbie the Hippo (A Children's Play)

Thomas W. Rodke – <http://www.hp41.org/LibView.cfm?Command=View&ItemID=988>

Here's a minimalistic amusement to start off the collection – from the author of the Monopoly Game program himself!

Herbie is a gloton Hippopotamus. Just feed him beans [B] or caviar [C] and watch him do his stuff! (no R/S is needed, auto entry).

Every other time you feed him caviar he does a backflip!

After 10 feedings he is full and goes to sleep (calculator automatically turns off).

Wake him up again (by turning on the HP-41C) and he is hungry again!

Program listing:

01*LBL "HERBIE"	30 PSE	59 "...PRRRP..."
02 CF 27	31 FC?C 23	60 AVIEW
03 SF 11	32 GTO 04	61 TONE 0
04 CF 01	33 ASTO 02	62 PSE
05 " HERBIE"	34 AOFF	63 ISG 00
06 AVIEW	35 XEQ 01	64 GTO 00
07 PSE	36 FC? 01	65 GTO 03
08 " THE"	37 GTO 02	66*LBL 05
09 AVIEW	38 1	67 " HUP HUP HUP,"
10 PSE	39 ST+ 01	68 AVIEW
11 "HIPPOPOTAMUS"	40 RCL 01	69 PSE
12 AVIEW	41 2	70 BEEP
13 PSE	42 MOD	71 "***BACKFLIP***"
14 ,01	43 X=0?	72 AVIEW
15 STO 00	44 XEQ 05	73 BEEP
16 0	45 "NBF"	74 BEEP
17 STO 01	46 AVIEW	75 PSE
18 " BEANS OR"	47 PSE	76 ISG 00
19 AVIEW	48 ISG 00	77 GTO 00
20 PSE	49 GTO 00	78*LBL 03
21 " CAVIAR ?"	50 GTO 03	79 " I AM FULL, THA"
22 AVIEW	51*LBL 01	80 ""NKS."
23 PSE	52 "C"	81 AVIEW
24*LBL 00	53 ASTO X	82 PSE
25 CF 01	54 RCL 02	83 OFF
26 " B OR C ?"	55 X=Y?	84 GTO "HERBIE"
27 AVIEW	56 SF 01	85 END
28 AON	57 RTN	
29*LBL 04	58*LBL 02	

Integer-Choice Game

Jakub Tatartkiewicz - PPCCJ V7N9 p24; (November 1980)

Reference Martin Gardener, "Mathematical Games" Scientific American, III/75.

Two persons choose a number (eg. 1-5). Whose smaller scores a point unless it is smaller by 1 – in this case the other player scores two points. Equal numbers give no scoring. The program goes automatically after XEQ "ICG"

Size: 002

Program Registers: 25

Program listing:

9:13AM 05/23	27 X=0?	54 GTO 00
01*LBL "ICG"	28 GTO 18	55*LBL 01
02 FIX 0	29 X<0?	56 1
03 CF 29	30 GTO 19	57 RTN
04 CLX	31 1	58*LBL 02
05 STO 01	32 X=Y?	59*LBL 03
06 CF 22	33 GTO 20	60*LBL 04
07*LBL 00	34 ST+ 01	61*LBL 05
08 RCL 00	35 GTO 18	62*LBL 06
09 RNG	36*LBL 20	63 2
10 STO 00	37 -2	64 RTN
11 16	38 ST+ 01	65*LBL 07
12 *	39 GTO 18	66*LBL 08
13 1	40*LBL 19	67*LBL 09
14 +	41 -1	68*LBL 10
15 XEQ IND X	42 X=Y?	69 3
16 "YOUR NUMBER?"	43 GTO 21	70 RTN
17 AVIEW	44 ST+ 01	71*LBL 11
18*LBL 22	45 GTO 18	72*LBL 12
19 PSE	46*LBL 21	73*LBL 13
20 FC?C 22	47 2	74*LBL 14
21 GTO 22	48 ST+ 01	75*LBL 15
22 "MINE IS "	49*LBL 18	76 4
23 ARCL Y	50 "POINTS "	77 RTN
24 AVIEW	51 ARCL 01	78*LBL 16
25 PSE	52 AVIEW	79 5
26 -	53 PSE	80 END

Secret Number

[Tom Cadwallader - PPCCJ V11N7 p25; \(August 1984\)](#)

SECRET NUMBER ("SN") is an exercise in stack manipulation and in stack analysis. It also can be used to decide who is going to buy lunch. The following listing of the HP-41 program was produced by using an HP-71B with Text Editor and ThinkJet printer. Therefore, the HP-41's "lazy tee" has been replaced with a "+" in append alpha string. (e.g. "+0"). XRM "FF" in the Standard is called to play a tune while providing a delay between display messages at the end of the game. It can be replaced with PSEs and/or TONEs.

Also the TIME Module is required in order to create a "random" seed and thus a "secret number". The seed is "massaged" with a commonly used modifier.

Start the game by XEQ'ing, "SN". "L: 000 H: 501" will be displayed. The first player enters a guess between 1 and 500 (e.g. 123) and presses R/S. Presuming that the secret number is greater than the guess, "L: 123 H: 501" will be displayed. The second player enters a guess between the new limits (e.g. 456) and presses R/S. Presuming that the secret number is less than the guess, "L: 123 H: 456" will be displayed. Play continues until some unlucky player sees "* YOU LOSE *" and hears the "music". If the secret number were 289, the final display would be "xxx 289 xxx". Guesses outside or the limits are trapped.

Tom Cadwallader (3502)
1204 26th Avenue SW
Great Falls, MT 59404

01*LBL "SN"	22 XEQ 03	43 X<=Y?	64 "* YOU LOSE *"
02 CLST	23 X<>Y	44 BEEP	65 AVIEW
03 CF 22	24 PROMPT	45 X<>Y	66 CLA
04 CF 27	25 FC?C 22	46 X<Y?	67 TONE 9
05 CF 29	26 0	47 X<>Y	68 TONE 8
06 FIX 0	27 INT	48 RDN	69 TONE 9
07 TIME	28 R^	49 X<>Y	70 BEEP
08 RN#	29 X=Y?	50 X<> Z	71 RDN
09 500	30 GTO 04	51 GTO 01	72 X<>Y
10 STO Z	31 X<Y?	52*LBL 03	73 XEQ 03
11 *	32 GTO 02	53 "" "	74 X<>Y
12 INT	33 RDN	54 100	75 XEQ 03
13 1	34 X<=Y?	55 X>Y?	76 X<> Z
14 ST+ Z	35 BEEP	56 ""0"	77 XEQ 03
15 +	36 X>Y?	57 SQRT	78 AVIEW
16 X<> Z	37 X<>Y	58 X>Y?	79 SF 29
17*LBL 01	38 RDN	59 ""0"	80 FIX 2
18 "L:"	39 GTO 01	60 RDN	81 CLST
19 XEQ 03	40*LBL 02	61 ARCL X	82 END
20 "" H:"	41 X<>Y	62 RTN	
21 X<>Y	42 R^	63*LBL 04	

HI-LO Game for Children

[Philip T. Frohme - PPCCJ V12N5 p25 ; \(May 1985\)](#)

This is a version of the game in which the user must make repeated guesses at a hidden number between 1 and 1000 inclusive. Unlike other Hi-Low games, this one provides the user with much more data as to his/her current status during the guessing process and after the hidden number has been found.

Although this is primarily a game for children, much can be learned about the behavior of random numbers by playing it. For instance, if guesses are halved each time to trap the hidden number, the average number of guesses to find the number will be around nine (9). The challenge is to reduce this average by taking calculated guesses as to what value the number might be.

Execute "HI-LO" and enter a seed (0 to 1) at the prompt. The display returns with 1-(1)-1000. The one (1) represents the lower boundary and the one thousand (1000) represents the upper boundary. These numbers will change as wrong guesses are made. This prevents the user from forgetting what numbers he/she has already chosen. The number in parenthesis is the number of the current attempt.

All the user has to do is enter a guess. The program will automatically resume (using flag 22) and display whether the guess was higher or lower than the hidden number. The display returns with the new boundaries; the higher or lower being the user's last guess. When the hidden number is found, the message --SUCCESS-- is displayed along with the "BEEP". The number of of guesses is displayed ("nn GUESSES"), the number of low guesses ("nn LOW"), the number of high guesses (nn HIGH), and then the number of games played with the average number of guesses per game (nn AV- mm). The challenge is to keep this average as low as possible for the amount of games played.

The message --WRONNNNG-- is displayed when a guess is made outside of the boundaries. No penalty in the way of extra turns is given; only the opportunity to retry with a correct number. Some users (and children) have trouble pushing the next digit within the time required.

Philip T. Frohme (9660)
11317 Amboy Lane
St. Louis, MO 63136-6102

[Program listing:](#)

01*LBL "HI-LO"

02 CLRG	41 GTO 07	81 PSE
03 CF 21	42 RCL Y	82 CLA
04 "SEED=?"	43 RCL 07	83 RCL 08
05 XEQ 02	44 X=Y?	84 RCL 09
06 STO 00	45 GTO 04	85 /
07*LBL 00	46 X>Y?	86 ARCL 09
08 ,	47 GTO 05	87 FIX 2
09 STO 01	48 GTO 06	88 " AV= "
10 STO 02	49*LBL 02	89 ARCL X
11 STO 03	50 AVIEW	90 AVIEW
12 STO 04	51 CF 22	91 PSE
13 E3	52 TONE 9	92 GTO 00
14 STO 05	53*LBL 03	93*LBL 05
15 CF 29	54 PSE	94 RCL Y
16 FIX 0	55 FC?C 22	95 STO 04
17 .	56 GTO 03	96 E
18 DATE	57 RTN	97 ST+ 02
19 E3	58*LBL 04	98 " LOW"
20 *	59 "--SUCCESS--"	99 AVIEW
21 INT	60 AVIEW	100 TONE 0
22 STO 07	61 BEEP	101 GTO 01
23 CLX	62 PSE	102*LBL 06
24*LBL 01	63 E	103 RCL Y
25 E	64 ST+ 09	104 STO 05
26 ST+ 01	65 "**** "	105 E
27 ST+ 08	66 ARCL Y	106 ST+ 03
28 CLA	67 " `***"	107 " HIGH"
29 ARCL 04	68 AVIEW	108 AVIEW
30 "(-"	69 PSE	109 TONE 9
31 ARCL 01	70 CLA	110 GTO 01
32 " ")"	71 ARCL 01	111*LBL 07
33 ARCL 05	72 " GUESSES"	112 "--WRONNNNG--"
34 XEQ 02	73 AVIEW	113 AVIEW
35 RCL 04	74 PSE	114 PSE
36 X>Y?	75 CLA	115 E
37 GTO 07	76 ARCL 02	116 ST- 01
38 X<>Y	77 " LOW, "	117 GTO 01
39 RCL 05	78 ARCL 03	118 END
40 X<Y?	79 " HI"	
	80 AVIEW	

Digit SWAP. (Classic Calculator Game)

Mark Power - DataFile V8N5 p15 ; (Jul/Aug 1989)

The following game is one of the classic 'Old Chestnuts' for programmable calculators. In the form given it will run on any HP41 (no modules, no synthetics, no mcode). It should be very easy to re-work it for the HPI5C, or take the idea and write a version for one of the new machines.

The object of the game is to rearrange 8 numbers to form the number 12345678. To do this you make moves by specifying a number in the range 1 to 7. This number is the 'gap' between two of the digits around which the two halves of the number are rotated.

Well, as that doesn't make it clear, here's an example:-

- Clear register 06 which is used to randomize the number. (You don't need to do this normally, it's just for the purposes of this example)
- XEQ'SWAP'
- After a while the screen shows – 68472315
- Try a move, say '4' 4 R/S
- The screen then shows – 76843152
- Now 17' maybe 7 R/S
- The screen shows – 57684312
- Eventually you should make a move which results in the number 12345678 being produced, in this case the screen will show the number of moves - 24 MOVES
- To run the program again press - R/S

Note, if you specify a move of 1, then the left hand digit stays where it is and just those on the right rotate. Similarly if 7 is specified, the number on the right hand end stays put.

Enough of the waffle - have a go yourself. Once you have a technique for solving the problem, you should aim for a solution within 6 or 7 moves.

For those of you rewriting the program for other machines, the ALPHA bit in LBL 04 is just to tell the user how many goes they took. You can just end the program after line 86 and forget the other bits. The CF 25 on line 09 is just for those HP41 users who may have ALPHA DATA in register 06 (used for the random numbers to jumble up the initial number). Registers used are 00 to 06 inclusive.

Anyone care to write a MCODE version with multiple rotation patterns and variable number of digits? (*)

Ed's note: See "REVERSE" in the MCODE games section,

Program listing:

01*LBL "SWAP"

02 12345678	34 GTO 02	67 INT
03 STO 02	35 8	68 LASTX
04 STO 03	36 X<=Y?	69 FRC
05 FIX 0	37 GTO 02	70 8
06 CF 29	38 X<>Y	71 RCL 00
07 5	39 XEQ 03	72 -
08 STO 05	40 GTO 02	73 10^X
09 SF 25	41*LBL 03	74 *
10*LBL 01	42 8	75 +
11 RCL 06	43 X<>Y	76 RCL 04
12 PI	44 -	77 +
13 +	45 STO 00	78 RCL 00
14 X^2	46 RCL 03	79 10^X
15 FRC	47 RCL 00	80 *
16 STO 06	48 10^X	81 STO 03
17 7	49 /	82 RTN
18 *	50 INT	83*LBL 04
19 E	51 STO 01	84 RCL 05
20 +	52 LASTX	85 E
21 INT	53 FRC	86 -
22 XEQ 03	54 E1	87 CLA
23 DSE 05	55 *	88 ARCL X
24 GTO 01	56 FRC	89 "" MOVE"
25*LBL 02	57 LASTX	90 2
26 E	58 INT	91 X<=Y?
27 ST+ 05	59 RCL 00	92 ""S"
28 RCL 02	60 10^X	93 AVIEW
29 RCL 03	61 /	94 FIX 4
30 X=Y?	62 +	95 SF 29
31 GTO 04	63 STO 04	96 CLX
32 STOP	64 RCL 01	97 END
33 X<=0?	65 E1	
	66 /	

HP-41 CX Word Game

Mark Gessner - PPCCJ V12N11 p20 (November 1985)

HP-41CX Gamers will appreciate this improved version of a popular word game currently available for the HP-41C/CV. The CX version plays much faster than the C/CV game, and requires fewer keystrokes, while still fitting on one magnetic card. Besides all these advantages, it's simply more fun.

SUMMARY OF GAME PLAY:

Player 1 keys XEQ "WG", and keys in a difficult-to-guess word at the prompt "WORD?". He then presses [R/S] and immediately hands the machine to player 2. Player 2 sees "LETTER?" at which he presses a letter representing his best guess at a letter in the word keyed in by player 1. The computer will then display a string of nulls indicating unguessed character positions in the word. Any correctly guessed characters will appear in the proper location in the string. Player 2 continues guessing letters until the entire word is filled in. At that time, a musical sequence will play, followed by the score for the game. Scoring is based on the number of guesses per letter in the word or phrase, and the lowest score wins.

DETAILS OF GAME PLAY:

Time: The number of seconds Player 2 will have available to guess each letter is equal to the number of letters in the word or phrase being guessed. For example, a twelve-letter word will give twelve seconds on each guess, while a three-letter word will give only three seconds for each guess. The penalty for taking longer than the number of seconds allotted for each guess will be the addition of one guess to your score, without actually guessing anything.

Words: The limits on what words are fair game are: 12 characters maximum, 1 character minimum. Only those characters which can be entered via normal keyboard means are legal. That is, any shifted or unshifted ALPHA character is OK, but strings which can be created only through XTOA and the like are absolutely out! The guesses are entered into the computer each time by the GETKEYX function, and there will be no way for a player to use XTOA to enter a guess. As for what types of words or phrases are fair game, i.e., proper nouns, abbreviations, etc., -that is entirely up to the players involved.

Tones: The computer signals its readiness to accept a keystroke guess by a TONE 87, a high-pitched, medium duration tone, while the acceptance of a valid guess keystroke sequence will be signaled by a TONE 89, a very short duration, high pitched tone.

Scores: The best possible score is 8, but this is rarely achieved in practice. (To achieve an 8, the word must consist of twelve identical characters, and player 2 must guess the correct character on the first guess.) A score of 100 is perfect for a word or phrase with all unique letters, but the scores can be lower if all letters are not unique.

Display: The display during game play is as follows: [*]-----

The first character is a "boxed star" and can be ignored. The remaining "overbar" or null characters represent unguessed character positions, and these will be replaced during game play with the characters they "cover" as the characters are correctly guessed. The display shown is as for an 11-character word which has had none of its characters correctly guessed.

TECHNICAL DETAILS:

REQUIRED EQUIPMENT: HP-41CX, 2 PLAYERS
 MINIMUM DATA SIZE: 33 REGS
 PROGRAM SIZE: 223 BYTES (32 REGS , 1 CARD)
 FLAGS USED: 05,25,29

Notes on program listing: the right brace character has been used in ALPHA strings to denote the ALPHA APPEND character, ASCII 127, and "#" tests for "not equal."

The game was originally modified by Doug Swanson, A fellow CX owner, and non-(yet)-PPC member, but the final speed / GETKEYX / size reductions / flag-keeping (fun stuff) were done by me, in my spare time. Have a good time.

Mark D. Gessner (11922)
 603 Southwest Parkway 74
 College Station, TX 77840-4752

Program listing:

01*LBL "WG"	31 GTO 07	61 STO IND Z	91 GTO 01
02 CLRG	32 .9	62 RTN	92 BEEP
03 RCLFLAG	33 STO 28	63*LBL 04	93 TONE 9
04 STO 27	34*LBL 01	64 CLA	94 TONE 7
05*LBL 71	35 RCL 00	65 2	95 RCL 28
06 FIX 0	36 STO 30	66 XTOA	96 INT
07 CF 29	37 FS? 05	67 13	97 RCL 00
08 "WORD?"	38 "LETTER?"	68 RCL 00	98 /
09 AON	39 FS?C 05	69 12	99 E2
10 PROMPT	40 AVIEW	70 +	100 *
11 SF 05	41 RCL 00	71*LBL 05	101 "SCORE "
12 ALENG	42 CHS	72 RCL IND X	102 ARCL X
13 STO 00	43 TONE 7	73 XTOA	103 AVIEW
14 2	44 GETKEY	74 RDN	104 PSE
15 XTOA	45 TONE 9	75 X=Y?	105 RCL 27
16 - E	46 STO 29	76 GTO 06	106 STOFLAG
17 AROT	47 ISG 28	77 DSE X	107 "AGAIN? Y/N"
18 ASTO 25	48*LBL 02	78 GTO 05	108 TONE 9
19 ASHF	49 RCL 30	79*LBL 06	109 AVIEW
20 ASTO 26	50 RCL 29	80 AVIEW	110 GETKEY
21 CLA	51 RCL IND Y	81 ASTO 31	111 SF 25
22 ARCL 25	52 X=Y?	82 ASHF	112 XEQ IND X
23 ARCL 26	53 XEQ 03	83 ASTO 32	113 CF 25
24 ATOX	54 DSE 30	84 32	114 CLA
25 R^	55 GTO 02	85 RCL 26	115 AOFF
26*LBL 07	56 GTO 04	86 X#NN?	116 CLST
27 ATOX	57*LBL 03	87 GTO 01	117 END
28 STO IND Y	58 12	88 31	
29 RDN	59 ST+ T	89 RCL 25	
30 DSE X	60 RDN	90 X#NN?	

High Rollers (FOCAL version).

Randal C. Gibson – PPCCJ V6N7 p47 (October 1979)

High Rollers is an old HP-41 game that first appeared in the PPC Journal back in August 1979. It is based on the Bonus Round of a popular TV game show of the same timeframe. An example game is presented below along with the source code to be keyed in. Enjoy!

The Game: You are presented with the numbers 1-9 in a list. A roll of two dice occurs and you attempt to choose a combination of numbers out of the 1-9 list that add up to the total of the two dice. If you can do so, the numbers chosen disappear from the list and another pair of dice are rolled. This continues until you either exactly remove all the 1-9 numbers or you cannot put together a combination of numbers that total your dice roll.

Pretty simple? So it sounds. It's actually very hard to do. I've played well over 1000 rounds of this game and my average percent of the time that I've won is around 16%.

Other rules: 1) You may not choose a number from the list more than once. (If you get a 1 and 3 on the two dice that are rolled, you may not choose 2 and 2 as the numbers to be removed from the list). 2) If the two dice you roll are the same, you are given a "doubles" chip. This doubles chip is used as insurance should you get a roll that you cannot find a correct combination in the list. This will be clearer from the game example below.

Once the game ends, you are shown your cumulative game score and the game starts over automatically (it assumes that you want to keep playing. ;-)).

Here's the sample game. SIZE 014, Bytes = 217 (Fits on 1 mag card, if anyone still uses those!) The left side shows the display and what you type in. The right side provides comments.

Start by XEQ "HR"

SEED?

Let's use a seed of 0.611940299

0.611940299 ENTER

1:1:123456789

You've rolled two 1's. Your bonus chip is indicated by the "1" between the two colons.

2 R/S

2 is equal to the two 1's you rolled and will be removed from the list.

5,5:2:1 3456789

You've rolled two 5's. Another bonus chip (aren't you lucky?). Notice the two from last time has disappeared. Only 8 more numbers to go!

64 R/S

6 and 4 equal 10 (the sum of the two 5's just rolled). You've now got 2 bonus attempts in case you can't find a proper combination.

2,12:1 3 5 789 You've rolled a 2 and 1. Notice again the 6 and 4 have disappeared. Let's try something wrong.

9 R/S 9 is not a proper entry for rolling a 2 and a 1. It should return the display with no changes.

2,12:1 3 5 789 It does. Let's choose 3 this time.

3 R/S
5,6:2:1 5 789 We've rolled a 5 and 6. No combination exists that will work. To "pass" on this roll and indicate we have no move, enter a zero (or decimal point).

0 R/S
4,2:1:1 5 789 Notice the 2 bonus chips have dropped to 1. We used one last time. Enter a 1 and 5 to use up the 6 (4 and 2) just rolled.

15 R/S
2,6:1 789 A 2 and 6. Let's use up the 8.

8 R/S
2,2:2: 7 9 Doubles, but this does us no good.

0 R/S
3,4:1 7 9 Almost there! Let's use up the 7.

7 R/S
3,4:1 9 Another 7. We have to use up our last bonus insurance chip!

0 R/S
3,6: 9 Great! We've won!

9 R/S
1/1 100.00% Our record so far is pretty good!

1,3: 123456789 The next game has started....

Here's the source code for the program!

A few lines below might need some explanation. Line 18 is append comma. Line 20 is append colon. Anywhere X NE 0? or X NE Y? is found they are X not equal to zero or X not equal to Y. Line 31 is append colon. Lines 33, 38 and 109 are append space (1 space only). Line 107 is append slash (the divide symbol). RDN is roll down, and RUP is roll up. Everything else should be self-explanatory!

Program listing:

01	LBL "HR"	48	X=0?	95	RCL IND X
02	CF 29	49	GTO 07	96	X#0?
03	FIX 0	50	ENTER	97	GTO 03
04	"SEED?"	51	ENTER	98	RDN
05	PROMPT	52	0	99	DSE X
06	STO 00	53	X<>Y	100	GTO 10
07	LBL 01	54	ENTER	101	ISG 14
08	9	55	LOG	102	LBL 11
09	LBL 02	56	INT	103	E
10	STO IND X	57	10^X	104	ST+ 15
11	DSE X	58	/	105	CLA
12	GTO 02	59	LBL 08	106	ARCL 14
13	LBL 03	60	INT	107	>"/"
14	XEQ 04	61	RCL IND X	108	ARCL 15
15	CLA	62	X=0?	109	>" "
16	ARCL X	63	GTO 06	110	RCL 14
17	XEQ 04	64	RDN	111	RCL 15
18	>" "	65	LASTX	112	/
19	ARCL X	66	RDN	113	E2
20	>": "	67	+	114	*
21	X#Y?	68	RUP	115	FIX 2
22	GTO 12	69	FRC	116	ARCL X
23	ISG 13	70	E1	117	AVIEW
24	LBL 12	71	*	118	FIX 0
25	+	72	X#0?	119	CLX
26	STO 12	73	GTO 08	120	STO 13
27	RCL 13	74	RDN	121	GTO 01
28	X#0?	75	RCL 12	122	LBL 07
29	ARCL X	76	X#Y?	123	RCL 13
30	X#0?	77	GTO 06	124	X=0?
31	->": "	78	RDN	125	GTO 11
32	X=0?	79	RDN	126	E
33	>" "	80	LOG	127	ST- 13
34	1.009	81	INT	128	GTO 03
35	LBL 05	82	10^X	129	LBL 04
36	RCL IND X	83	/	130	RCL 00
37	X=0?	84	LBL 09	131	997
38	->" "	85	0	132	*
39	X#0?	86	STO IND Y	133	FRC
40	ARCL X	87	RDN	134	STO 00
41	RDN	88	FRC	135	6
42	ISG X	89	E1	136	*
43	GTO 05	90	*	137	E
44	LBL 06	91	X#0?	138	+
45	PROMPT	92	GTO 09	139	INT
46	FC?C 22	93	9	140	END
47	GTO 06	94	LBL 10		

Simon – Memory Game

HP Co. – Games Solutions Books

This game exercises your memory by presenting longer and longer sequences of random numbers. You try to remember and key in each sequence. Flag settings may be varied to change the difficulty.

Example: use a seed of pi for the random number generation to duplicate this game.

<u>Keystrokes</u>	<u>Display</u>	
XEQ "SIZE" 004, PI, STO 00		
XEQ "SIMON"	HOW MANY?	
3, R/S	1	(in sequence)
	NUMBERS?	
1, R/S	YES: 1	
	9 . 2	(in sequence)
	NUMBERS?	
9, R/S	NO: 92, NOT 9	
	3 . 4 . 6	(in sequence)
	NUMBERS?	
346, R/S	YES: 346	
	YOU MISSED 1	

Note 1: You can set flag 0 (SF 00) to use longer and longer pieces of the same sequence. This version of the game is easier for young children.

Note 2: You can clear flag 26 (CF 26) to suppress the tone and make the sequences pass more quickly. Some people find them easier to memorize this way

Note 3: You can start with a sequence longer than 1 digit by keying in a number of the form $100a+b$, where "a" is one less than the length of the first sequence you want and "b" is the maximum length. For example, 2006 would yield sequences of length 3, 4, 5, and 6.

Program listing:

<see next page>



01*LBL "R"

02 FS? 00
03 RTN
04 RCL 00
05 9821
06 *
07 .211327
08 +
09 FRC
10 1.1111
11 *
12 FRC
13 STO 00
14 RTN

15*LBL "SIMON"

16 FIX 0
17 CF 29
18 CF 06
19 FS?C 00
20 SF 07
21 XEQ "R"
22 FS?C 07
23 SF 00
24 "HOW MANY?"
25 PROMPT
26 1 E3
27 /
28 1
29 +
30 STO 01
31 CLX
32 STO 02
33*LBL 10
34 RCL 01
35 INT
36 XEQ "R"

37 XEQ "TONES"
38 XEQ "?"
39 FS? 05
40 GTO "NO"
41 "YES: "
42 ARCL X
43 AVIEW
44 ISG 01
45 GTO 10
46 FS?C 06
47 GTO 01
48 BEEP
49 BEEP
50 "YOU WIN"

51 AVIEW

52 RTN

53*LBL "TONES"

54 RCL 01
55 INT
56 STO 03
57 RCL 00
58 FRC
59*LBL 03
60 10
61 *
62 INT
63 VIEW X
64 TONE IND X
65 LASTX
66 FRC
67 DSE 03
68 GTO 03
69 RTN

70*LBL "NO"

71 SF 06
72 TONE 2

73 TONE 0
74 "NO: "
75 ARCL X
76 ", NOT "
77 ARCL Y
78 AVIEW
79 PSE
80 1
81 ST+ 02
82 ISG 01
83 GTO 10
84*LBL 01
85 "YOU MISSED "
86 ARCL 02
87 CF 06
88 AVIEW
89 RTN
90*LBL "?"
91 "NUMBERS?"
92 PROMPT
93 CF 05
94 RCL 00
95 RCL 01
96 INT
97 10^X
98 *
99 INT
100 X=Y?
101 RTN
102 SF 05
103 RTN
104 END

Simon – Reloaded

Whodunit – MoHP Disks

A different version of unknown author, from the MoHP Disks...

With this version the player uses the local labels A-J for the digit input, instead of typing the actual digits. In that regard it more closely resembles the physical game.

Program listing:

1 <u>LBL "SIMON"</u>	31 ISG 00	61 <u>LBL B</u>
2 <u>LBL a</u>	32 PROMPT	62 1
3 2,001	33 <u>"CORRECT"</u>	63 GTO 02
4 STO 00	34 AVIEW	64 <u>LBL C</u>
5 <u>LBL 01</u>	35 GTO 01	65 2
6 RCL 00	36 <u>LBL 03</u>	66 GTO 02
7 ENTER^	37 <u>"YOU BLEW IT"</u>	67 <u>LBL D</u>
8 FRC	38 AVIEW	68 3
9 2,001	39 TONE 3	69 GTO 02
10 +	40 TONE 2	70 <u>LBL E</u>
11 STO 00	41 TONE 1	71 4
12 RCL 01	42 TONE 0	72 GTO 02
13 PI	43 <u>LBL b</u>	73 <u>LBL F</u>
14 +	44 RCL 00	74 5
15 E^X	45 FRC	75 GTO 02
16 FRC	46 STO 00	76 <u>LBL G</u>
17 STO 01	47 2	77 6
18 10	48 +	78 GTO 02
19 *	49 <u>"IT WAS..."</u>	79 <u>LBL H</u>
20 INT	50 AVIEW	80 7
21 STO IND Z	51 LBL 04	81 GTO 02
22 RCL 00	52 RCL IND X	82 <u>LBL I</u>
23 XEQ 04	53 TONE IND X	83 8
24 <u>"GO"</u>	54 X<>Y	84 GTO 02
25 PROMPT	55 ISG X	85 <u>LBL J</u>
26 <u>LBL 02</u>	56 GTO 04	86 9
27 TONE IND X	57 RTN	87 GTO 02
28 RCL IND 00	58 <u>LBL A</u>	88 END
29 X#Y?	59 0	
30 GTO 03	60 GTO 02	

Code Crack

HP Co. – Games Solutions Book

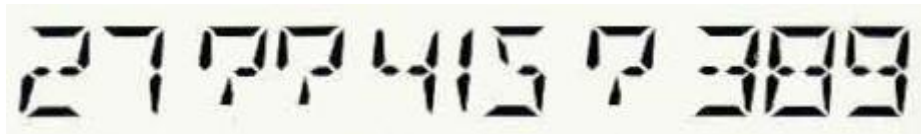
Yet another MasterMind in disguise... oh dear.

Program listing:

01*LBL "MM"	36 STO IND 13	72 FS? 05	108 RCL 12
02 FIX 0	37 1	73 GTO 11	109 15.01501
03 CF 29	38 ST+ IND Y	74 RDN	110 +
04 SF 21	39 DSE 13	75 RCL IND 13	111 STO 13
05 "NO. DIGITS?"	40 GTO 01	76 X=Y?	112 RTN
06 PROMPT	41 "GUESS"	77 XEQ 03	113*LBL 04
07 STO 12	42 ARCL 12	78*LBL 11	114 SF 27
08 "NO. IN	43 AVIEW	79 DSE 13	115 "OK, "
CODE?"	44*LBL 14	80 GTO 02	116 ARCL 14
09 PROMPT	45 CF 05	81 FS? 05	117 " TRIES"
10 STO 11	46 STO 00	82 GTO 12	118 AVIEW
11 "SEED?"	47 1	83 RCL 12	119 RTN
12 PROMPT	48 ST+ 14	84 RCL 22	120*LBL 12
13 STO 10	49 0	85 X=Y?	121 CLA
14*LBL C	50 STO 22	86 GTO 04	122 ARCL 00
15 CF 27	51 STO 23	87 9	123 >" "
16 9	52*LBL 10	88 STO 13	124 RCL 22
17 ENTER^	53 RCL 00	89*LBL 05	125 X=0?
18 0	54 STO 15	90 RCL IND 13	126 GTO 07
19 STO 14	55 XEQ 06	91 X>0?	127*LBL 08
20*LBL 00	56*LBL 02	92 ST+ 23	128 >"*"
21 STO IND Y	57 RCL 15	93 DSE 13	129 DSE X
22 DSE Y	58 10	94 GTO 05	130 GTO 08
23 GTO 00	59 /	95 RCL 12	131*LBL 07
24 XEQ 06	60 FRC	96 RCL 23	132 RCL 23
25*LBL 01	61 LASTX	97 -	133 X=0?
26 RCL 10	62 INT	98 RCL 22	134 GTO 13
27 997	63 STO 15	99 -	135*LBL 09
28 *	64 RDN	100 STO 23	136 "+"
29 FRC	65 10	101 SF 05	137 DSE X
30 STO 10	66 *	102 GTO 10	138 GTO 09
31 RCL 11	67 1	103*LBL 03	139*LBL 13
32 *	68 FC? 05	104 1	140 AVIEW
33 INT	69 ST- IND Y	105 ST+ 22	141 GTO 14
34 1	70 FS? 05	106 RTN	142 END
35 +	71 ST+ IND Y	107*LBL 06	

Super Bagels

HP Co. – Games Pac



The object of the game of Super Bagels is for the player to guess a number which the calculator has chosen. Clues are given after each guess to tell the player how close his guess is to the hidden number. To make the game more interesting, the hidden number can be specified by the player to be from 1 to 8 digits in length. The maximum digit value is also specified by the player. When the hidden number is finally entered, the number of guesses required to discover the hidden number is displayed.

A game where the number of digits is 4 and the maximum digit value is 5, is of moderate difficulty. In this game the player tries to guess a 4 digit number with each digit having any value from 0 to 5. Thus the minimum possible number is 0 (0000) and the maximum is 5555.

After each guess is entered and [R/S] is pressed, a clue is displayed indicating how well that number matched the hidden number. The guess appears in the left side of the display and the clue in the right side. The clue has the form "PLC-X VAL" where PLC is the number of digits of the guess that exactly match digits in the hidden number both in value and the placement. X VAL (extra values) is the number of digits which match in value but not in location. Digits are not counted twice: that is, digits counted in PLC are not counted again in X VAL digits .

For example, if the hidden number is 0025 1. a guess of 01234 would yield 01234 2- 1. This display means that two numbers (the 0 and the 2) match the hidden number exactly , but that one number (the 1) is out of place. When the guess finally matches the hidden number, the hidden number and the number of guesses is displayed. Next the calculator prompts with "SAME? Y/N". Pressing the letter Y followed by [R/S] enables the user to continue playing with his previous limits. N, [R/S] should be pressed to play a game with new limits.

If the user forgets the limits and the calculator expects a guess as the next input the user can get a review of the limits entering a negative number as a guess. The first time through the game there will be three reminder messages. The first message is "DIGS CAN DUP" . This message serves as a reminder that there can be duplicate digits. The second message immediately follows: "0 LEGAL" appears in the display to remind the user that 0 is a legal digit.

The third reminder message is displayed just before the first due. PLC=m, XVAL=n (where m and n are numbers) serves as a reminder that the first digit of the due is the number of digits of the guess which are correct in value and placement. XVAL= stands for extra values. These are numbers which are correct in value only.

Example 1:

Play a game with 4 digits, each in the range 0 through 5.

Keystrokes (SIZE >= 028) Display

XEQ "BAGELS"	SEED?	
75192.23, CHS, R/S	LENGTH?	
4, R/S	MAX DIG?	
5, R/S	DIGS CAN DUP	
R/S	0 LEGAL	
R/S	GUESS?	
3214, R/S	PLC=0 XVAL=2	
R/S	3214: 0-2	0 in the right places, 2 correct
0514, R/S	0514: 1-3	1 in the right places, 3 correct
0145, R/S	0145: 1-3	
0451, R/S	0451: 0-4	4 correct !
1540, R/S	YOU GOT IT.	
R/S	1540	
R/S	5 GUESSES	
R/S	SAME? Y.N	

Program listing:

01*LBL "BAGELS"	32 GTO d	62 FC? 05	93 XEQ 11
02 28	33 STO 27	63 GTO 00	94 RCL 23
03 XROM "SIZE?"	34 9	64 FS? 07	95 CF 06
04 FC?C 25	35 X<>Y	65 GTO 01	96 XEQ 11
05 PROMPT	36 X>Y?	66 CLX	97 XEQ e
06 SF 08	37 GTO d	67 RCL 24	98*LBL 02
07 XROM "INI"	38 RCL 20	68 1/X	99 RCL IND 10
08 STO 24	39 /	69 R-D	100 RCL 20
09 CF 05	40 ST+ 21	70 STO 24	101 ST- 10
10 10	41 FC? 08	71*LBL 00	102 CLX
11 STO 20	42 GTO b	72 SF 05	103 RCL IND 10
12*LBL a	43 "DIGS CAN	73 CLX	104 X#Y?
13 RCL 25	DUP"	74 .211327	105 GTO 03
14 "LENGTH?"	44 AVIEW	75 ENTER^	106 CHS
15 PROMPT	45 "0 LEGAL"	76 9821	107 X<>Y
16 ABS	46 AVIEW	77 RCL 24	108 CHS
17 INT	47*LBL b	78 *	109 X<>Y
18 X=0?	48 CLX	79 +	110 1
19 GTO d	49 STO 22	80 FRC	111 ST+ 19
20 8	50 SF 29	81 D-R	112 RDN
21 X<>Y	51 SF 09	82 1/X	113*LBL 03
22 X>Y?	52 "GUESS?"	83 STO 24	114 STO IND 10
23 GTO d	53 PROMPT	84*LBL 01	115 X<>Y
24 STO 21	54 CF 29	85 CF 07	116 RCL 20
25 STO 25	55 X<0?	86 RDN	117 ST+ 10
26 RCL 27	56 GTO 16	87 INT	118 RDN
27 "MAX DIG?"	57 CF 09	88 STO 23	119 STO IND 10
28 PROMPT	58*LBL c	89 RCL 24	120 DSE 10
29 ABS	59 0	90 1/X	121 GTO 02
30 INT	60 STO 19	91 R-D	122 XEQ e
31 X=0?	61 STO 26	92 SF 06	123*LBL 04

124 RCL IND 10	170 CF 08	216*LBL 10	262 INT
125 X<0?	171 "PLC="	217 AVIEW	263 STO IND 10
126 GTO 07	172 ARCL 19	218 SF 29	264 DSE 10
127 SF 07	173 "` XVAL="	219 AON	265 GTO 13
128 RCL 25	174 ARCL 26	220 "N"	266 RTN
129 STO 00	175 AVIEW	221 ASTO Y	267*LBL d
130*LBL 05	176 PSE	222 "SAME? Y/N"	268 TONE 4
131 CLX	177*LBL 07	223 PROMPT	269 "ILLEGAL NO."
132 RCL IND 00	178 XEQ 17	224 AOFF	270 AVIEW
133 X=Y?	179 ">:"	225 ASTO X	271 GTO a
134 GTO 06	180 FIX 0	226 X=Y?	272*LBL e
135 DSE 00	181 9	227 GTO a	273 RCL 25
136 GTO 05	182 RCL 25	228 GTO b	274 RCL 20
137 GTO 07	183 -	229*LBL 11	275 +
138*LBL 06	184 STO 00	230 RCL 21	276 .01
139 CHS	185*LBL A	231 FRC	277 +
140 STO IND 00	186 "" "	232 RCL 20	278 STO 10
141 1	187 DSE 00	233 1/X	279 RTN
142 ST+ 26	188 GTO A	234 +	280*LBL 16
143*LBL 07	189 ARCL 19	235 STO 00	281 CF 29
144 DSE 10	190 ">-"	236 CLX	282 CLA
145 GTO 04	191 ARCL 26	237 RCL 25	283 FIX 0
146 RCL 25	192 SF 29	238 STO 10	284 "LEN="
147 STO 10	193 PROMPT	239 RDN	285 ARCL 25
148 CLX	194 CF 29	240 FS? 06	286 "", MAX="
149*LBL 08	195 CLA	241 GTO 12	287 ARCL 27
150 STO IND 10	196 X<0?	242 R^	288 AVIEW
151 DSE 10	197 GTO 16	243 10^X	289 CLA
152 GTO 08	198 ADV	244 /	290 FS? 09
153 XEQ e	199 GTO c	245 ABS	291 GTO b
154 CLX	200*LBL 10	246 XEQ e	292 GTO 15
155*LBL 09	201 TONE 8	247 RDN	293*LBL 17
156 STO IND 10	202 TONE 9	248*LBL 12	294 CLA
157 DSE 10	203 "YOU GOT IT."	249 STO 09	295 RCL 23
158 GTO 09	204 AVIEW	250*LBL 13	296 RCL 25
159 FIX 0	205 XEQ 17	251 RCL 09	297 10^X
160 1	206 AVIEW	252 FRC	298 /
161 ST+ 22	207 FIX 0	253 RCL 20	299 FRC
162 FC? 07	208 CLA	254 *	300 FIX IND 25
163 GTO 10	209 ARCL 22	255 STO 09	301 " "
164 CLX	210 "" GUESS"	256 RCL 00	302 ARCL X
165 TONE 5	211 1	257 X<>Y	303 ASHF
166 TONE 5	212 RCL 22	258 FS? 06	304 RTN
167*LBL 15	213 X=Y?	259 *	305 END
168 FC? 08	214 GTO 10	260 1	
169 GTO 07	215 "`ES"	261 +	

Hangman w/ X-Functions.

Julian Perry – DataFile V1N1 p5 ; (July 1982)

This version of the popular Hangman game was written to take advantage of the improved ALPHA manipulation available in the Extended functions/Memory Module. The program requires 390 bytes of program memory and 9 data registers. Some synthetic instructions are used but these are only alpha text lines and could easily be omitted without too much alteration. This program is faster than most similar games due to the use of the XF/M with such functions as POSA, XTOA & AROT; there is also a special feature to allow a word list to be stored in extended memory.

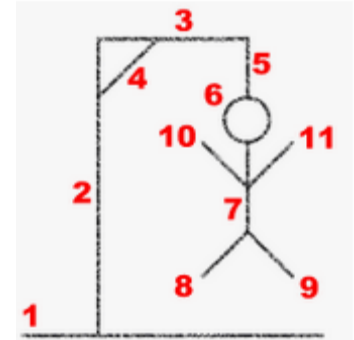
Instructions.

- 1) Load the program and set SIZE 009. You may wish to include the program lines 9 PSIZE at the beginning of the program.
- 2) XEQ "HANGMAN"
- 3) If you have a word list in extended memory (named "WORD-L") you will be prompted by "LIST? (Y/N)", if not go to step 5.
- 4) Press "Y" if you want the next word from the list to be used or press any other key if you want to enter a word manually. If "END OF FL" is displayed then the word list has been used up – go to step 2 to start again at the beginning of the list. If you pressed "Y" then go to step 6.
- 5) You will next be prompted to enter a word manually by "WORD?" Enter a word of up to 12 letters (including spaces) and press R/S
- 6) The next prompt will be a series of underscores (character 95), one for each letter in the word. Guess at a single letter and press R/S
- 7) Continue guessing at letters until you get all of them or until you are hung for taking too many attempts. If at any stage you want to know which letters you have guessed at but are not in the mystery word press the question mark ("?",) and then R/S, you will see a display of the wrong letters, but before them there will be some characters representing the gallows ("@") and a little man ("π") which will build up to be a complete "hangman" as you get more letters wrong. Once the gallows are complete and the man has all of its limbs then you are hung ("YOU'RE HUNG" displayed) - press R/S to see what the word was. You can go back and forth between the two displays of the wrong letters and the current correct letters by using the question mark.
- 8) After you have correctly guessed the word, or have been hung press R/S for a new game and go to step 3

If you want to have a word list it can be created as follows:

- a. Work out the total number of registers needed for the ASCII file (see page 24 of the XF/M manual) with one word stored per record – and enter this number into the X register
- b. Key in "WORD-L" into the ALPHA register
- c. XEQ "CRFLAS"
- d. To enter the words – key in the word to the ALPHA register and XEQ "APPREC" (see example on page 27 of the manual)

You will find that another file will automatically be created the first time you use a word from the list. This data file ("WORD-D") is one register long and contains a pointer to the next unused word from the list. If this file is deleted then the pointer is set to the first word in the list.



I hope you like the program... Have fun. Julian Perry

Program listing:

1 LBL "HANGMAN"	32 SAVEX	63 ASTO 05
2 SF 25	33 RCL 00	64 ASTO 06
3 "WORD-L"	34 CLA	65 32
4 ASTO 01	35 ARCL 01	66 XEQ 14
5 FLSIZE	36 SF 25	67 CLA
6 FC? 25	37 SEEKPTA	68 ASTO 07
7 GTO 02	38 SF 25	69 ASTO 08
8 "LIST ? (Y/N)"	39 GETREC	70 CLX
9 AVIEW	40 FS?C 25	71 STO 00
10 LBL 01	41 GTO 03	72 CF 05
11 GETKEY	42 CLA	73 LBL 20
12 X=0?	43 ARCL 02	74 FS?C 05
13 GTO 01	44 CLFL	75 GTO 04
14 71	45 1 E	76 SF 05
15 X#Y?	46 SEEKPT	77 CLA
16 GTO 02	47 LBL 02	78 ARCL 05
17 CLD	48 "WORD ?"	79 ARCL 06
18 SF 25	49 AON	80 GTO 05
19 1 E	50 PROMPT	81 LBL 04
20 "WORD-D"	51 12	82 " _ L C @"
21 ASTO 02	52 ALENG	83 ">"@`@---@--@--- "
22 CRFLD	53 X>Y?	84 RCL 00
23 CLX	54 GTO 02	85 2
24 SEEKPTA	55 LBL 03	86 *
25 GETX	56 ">" "	87 LASTX
26 STO 00	57 ASTO 01	88 +
27 1 E	58 ASTO 03	89 AROT
28 +	59 ASHF	90 ASHF
29 .	60 ASTO 02	91 ASHF
30 SEEKPT	61 ASTO 04	92 ASHF
31 X<>Y	62 " _ "	93 ">" ,"

94 ARCL 07	123 ALENG	152 RDN
95 ARCL 08	124 STO 00	153 RDN
96 LBL 05	125 ASTO 07	154 AROT
97 AON	126 ASHF	155 ASTO 03
98 CF 23	127 ASTO 08	156 ASHF
99 PROMPT	128 9	157 ASTO 04
100 63	129 X>Y?	158 CLA
101 ATOX	130 GTO 20	159 ARCL 05
102 FS? 23	131 " YOU'RE HUNG"	160 ARCL 06
103 X=Y?	132 TONE 01	161 X<>Y
104 GTO 20	133 TONE 00	162 AROT
105 AOFF	134 PROMPT	163 ATOX
106 CF 05	135 GTO 06	164 RCL 00
107 XEQ 14	136 LBL 14	165 XTOA
108 CLA	137 STO 00	166 R^
109 ARCL 05	138 CLA	167 AROT
110 ARCL 06	139 ARCL 03	168 X<>Y
111 95	140 ARCL 04	169 ASTO 05
112 POSA	141 POSA	170 ASHF
113 X<0?	142 X<0?	171 ASTO 06
114 BEEP	143 RTN	172 GTO 14
115 X<0?	144 SF 05	173 LBL 06
116 GTO 06	145 AROT	174 " WORD= "
117 CLA	146 11	175 ARCL 01
118 RCL 00	147 RCL Y	176 ARCL 02
119 FC?C 05	148 -	177 AVIEW
120 XTOA	149 ATOX	178 TONE 07
121 ARCL 07	150 33	179 END
122 ARCL 08	151 XTOA	

Note from the Editor: Since the original "HM" version written by Bill Wickes and published in his seminal book "Synthetic programming on the HP-41", multiple variations of the same subject have been made available by the user community. What follows are the program listings for Bill's original (a tour de force in synthetics), and another version (in French) from the Swap Disks, also using the extended functions set.

In fact, in a wild librarian vein of an utterly useless purpose, one's tempted to compile a dedicated ROM image with all the versions available for this classic game, perhaps also including all the Master Mind (and derivatives) programs ... to be continued.

Mastermind. Jean-Marc Baillard. <http://hp41programs.yolasite.com/mastermind.php>
 Mastermind. Wouter Peters. <https://www.hpmuseum.org/software/41/41master.htm>
 Mastermind. Kai Schroeder. http://www.achim-und-kai.de/kai/hp41cx/superhirn_var1_e.html
 Mastermind. DataFile
 Mastermind. PPCCJ
 Mastermind. Swap Disks

Hangman Game

HP Co. – Games Pac

This game is a version of the popular word game "hangman". The first player selects a word or phrase with as many as 12 characters in length and keys it into the calculator. The second player guesses various characters until he completes the word or gets hanged. The second player gets 7 wrong guesses before he is hanged.

After each guess the display shows the previous wrong guesses, next the number of wrong guesses left, then the correctly guessed characters in their appropriate places. If the player is "hanged" before the word is guessed, the word is displayed.



Program listing:

01*LBL "HANG"	32 GTO 13	63 RCL 12
02 18	33 RCL 17	64 STO 15
03 XROM "SIZE?"	34 X#Y?	65 RCL 14
04 FC?C 25	35 GTO 14	66 CF 06
05 PROMPT	36 RDN	67 CF 05
06 FIX 0	37 6	68*LBL 04
07 CF 29	38 RCL 12	69 6
08 AON	39 INT	70 /
09 "WORD?"	40 Y^X	71 FRC
10 STOP	41 ST+ 14	72 X=0?
11 ASTO 13	42*LBL 14	73 GTO 05
12 CF 23	43 X<> Z	74 ARCL IND 15
13 ASHF	44 ISG 12	75 GTO 06
14 .005	45 GTO 12	76*LBL 05
15 XEQ 01	46*LBL 13	77 SF 05
16 ARCL 13	47 RCL 12	78 "-"
17 6.011	48 1	79*LBL 06
18 XEQ 01	49 -	80 LASTX
19 7	50 INT	81 INT
20 STO 13	51 1 E3	82 ISG 15
21 CLX	52 /	83 GTO 04
22 STO 14	53 STO 12	84 FC?C 05
23 .011	54 SF 06	85 GTO 09
24 STO 12	55 CLA	86*LBL 07
25 CLA	56 ASTO 17	87 PROMPT
26 ASTO X	57*LBL 03	88 FC?C 23
27 " "	58 CLA	89 GTO 08
28 ASTO 17	59 ARCL 13	90 ASTO 16
29*LBL 12	60 " LEFT"	91 CLA
30 RCL IND 12	61 AVIEW	92 ARCL 17
31 X=Y?	62 CLA	93 AVIEW

94 RCL 12	115 ARCL 16	136 ASTO 14
95 STO 15	116 ASTO 17	137*LBL 02
96*LBL 10	117 DSE 13	138 " "
97 RCL IND 15	118 GTO 03	139 ARCL 14
98 RCL 16	119 XROM "BOOM"	140 ASTO 14
99 X#Y?	120 6	141 ASHF
100 GTO 11	121 RCL 15	142 RCL 15
101 SF 06	122 INT	143 12
102 6	123 Y^X	144 -
103 RCL 15	124 1	145 ASTO IND X
104 INT	125 -	146 ISG 15
105 Y^X	126 STO 14	147 GTO 02
106 ST+ 14	127 GTO 03	148 CLA
107 TONE 9	128*LBL 09	149 RTN
108*LBL 11	129 BEEP	150*LBL 08
109 ISG 15	130 AOFF	151 VIEW 17
110 GTO 10	131 SF 29	152 STOP
111 FS?C 06	132 PROMPT	153 GTO 07
112 GTO 03	133 GTO "HANG"	154 END
113 CLA	134*LBL 01	
114 ARCL 17	135 STO 15	

Hangman (Word Guessing)

HP Co. – Standard Apps.

This program is a version of the word game "hangman". The first player makes up a six-character word and gives it to the calculator. The second player guesses various letters until he has completed the word. After each guess, the calculator displays all correctly guessed characters in their appropriate places. When the entire word has been guessed, the number of guesses is displayed.

Instructions:

1. Set status and key in the program
2. Begin running the program XEQ "WORDS"
3. First player: Key in your word <six chars>, R/S
4. Second player: Guess a Char. <any char>, R/S
5. Repeat step 4 to guess more characters. When word is complete, you will see "DONE, WORD IS <word>" and "YOU TOOK <nn> GUESSES"

Example:

Hide "HP-41C" and then guess it.

Keystrokes:

XEQ ALPHA WORDS ALPHA
HP-41C R/S

A R/S

P R/S

C R/S

H R/S

4 R/S

1 R/S

- R/S

Display:

KEY IN WORD
LETTER?

(Notice that the
program stops in
ALPHA mode.)

LETTER?

P

LETTER?

P C

LETTER?

HP C

LETTER?

HP 4 C

LETTER?

HP 41C

LETTER?

DONE

WORD IS <HP-41C>

YOU TOOK 7 GUESSES

Program listing:

01*LBL "WORDS"

02 "KEY IN WORD"

03 AON

04 PROMPT

05 ASTO 08

06 6

07 XEQ "DESPEL"

08 ,9

09 STO 17

10 " "

11 ASTO 09

12 16,01

13 XEQ "DESPEL"

14*LBL "LTTR"

15 CLA

16 ASTO 09

17 "LETTER?"

18 AON

19 PROMPT

20 ASTO 10

21 ISG 17

22 1,006

23 STO 18

24*LBL 06

25 " "

26 ASTO Y

27 RCL 18

28 10

29 +

30 CLA

31 ARCL IND X

32 RDN

33 ASTO X

34 X#Y?

35 GTO 00

36 CLA

37 ARCL 10

38 ASTO Y

39 CLA

40 ARCL IND 18

41 ASTO X

42 X=Y?

43 GTO 00

44 " "

45 ASTO X

46*LBL 00

47 CLA

48 ARCL 09

49 ARCL X

50 ASTO 09

51 AVIEW

52 10

53 RCL 18

54 +

55 CLA

56 ARCL Y

57 ASTO IND X

58 ISG 18

59 GTO 06

60 CLA

61 ARCL 08

62 ASTO Y

63 CLA

64 ARCL 09

65 ASTO X

66 X=Y?

67 GTO 00

68 PSE

69 PSE

70 GTO "LTTR"

71*LBL 00

72 "DONE"

73 AVIEW

74 "WORD IS <"

75 ARCL 09

76 ">"

77 AVIEW

78 PSE

79 PSE

80 RCL 17

81 INT

82 "YOU TOOK "

83 ARCL X

84 "` GUESSES"

85 AVIEW

86 RTN

87*LBL "DESPEL"

88 STO 07

89 ASTO 00

90*LBL 07

91 " "

92 ARCL 00

93 ASTO 00

94 ASHF

95 ASTO IND 07

96 DSE 07

97 GTO 07

98 RTN

99 END

Hangman (w/ SP Gallows)

William C. Wickes – PPCCJ V7N4 p31 (May 1980)

As well as being an amusing elaboration of various word guessing games, this hangman program serves as an illustration of the use of alpha manipulation techniques using synthetic function techniques: modified versions of "SUB" and "ISO" are found in lines 122-164 and lines 170-181 respectively.

User instructions: XEQ "HM". At the prompt "**WORD?**", one player enters a word of one to nine letters, then R/S. At the tone, the HP-41C display will show a number of dashes equal to the number of dashes in the mystery word. The second player then enters a guess letter, R/S. The guess letter will remain in the display until the next tone. At this point, if the letter was contained in the word, all occurrences of the letter will be displayed in the dashed word. The second player then guesses again, and so on.

If the guessed letter was not in the mystery word, the display will show the "gallows" @ starting construction with the base. For each subsequent guess, another piece of the gallows is added. After the fifth wrong guess, the display begins building the man - next to the gallows, starting with the head T • After the ninth wrong guess, the man is complete, and the tenth wrong guess will result in "hanging", whereupon the mystery word is displayed. If the second player gets all of the letters before he is hung, the ending sequence will show the number of wrong guesses.

Program listing:

01*LBL "HM"	22 "___"	43 TONE 9	64 ISG 06
02 ,	23 ASTO 03	44 CLD	65 GTO 02
03 STO d	24 ARCL 03	45 STOP	66 FS?C 19
04 ,009	25 ASTO 02	46 ASTO 05	67 GTO 01
05 STO 07	26 CLA	47 RCL d	68 ISG 07
06 FIX 0	27 ASTO 04	48 AVIEW	69 GTO 06
07 SF 26	28 1,009	49 STO d	70 "ARRRRGGH..."
08 "WORD?"	29 STO 08	50*LBL 02	71 AVIEW
09 AON	30 STO 06	51 FS? IND 06	72 TONE 0
10 STOP	31 SF 19	52 GTO 04	73 TONE 0
11 " " "	32 " "	53 RCL 05	74 PSE
12 ASTO 00	33 ASTO 05	54 CLA	75 "WORD IS: "
13 ASHF	34 GTO 02	55 ARCL 00	76 ARCL 00
14 ASTO X	35*LBL 01	56 ARCL 01	77 ARCL 01
15 CLA	36 1,009	57 RCL 06	78 AOFF
16 ARCL X	37 STO 06	58 INT	79 PROMPT
17 "^^^^"	38 CLA	59 XEQ 08	80 GTO "HM"
18 RCL \	39 ARCL 02	60 ASTO X	81*LBL 06
19 CLA	40 ARCL 03	61 X=Y?	82 "-----"
20 STO [41 " " "	62 XEQ 03	83 E1
21 ASTO 01	42 ARCL 04	63*LBL 04	84 RCL 07

85 INT	110 INT	136 CLX	162 STO [
86 -	111 ARCL X	137 FIX 4	163 CLX
87 XEQ 08	112 "`WRONG."	138 ARCL X	164 X<> ^
88 ASTO X	113 AOFF	139 CLX	165 STO \
89 RCL 07	114 PROMPT	140 X<> \	166 ASTO 02
90 INT	115*LBL 05	141 STO [167 ASHF
91	116 SF IND 06	142 CLX	168 ASTO 03
"@@@@@CL_"	117 SF 19	143 X<>]	169 RTN
92 XEQ 08	118 RCL 06	144 STO \	170*LBL 08
93 ARCL Y	119 INT	145 X<> T	171 E1
94 ASTO 04	120 CLA	146 X<>]	172 -
95 GTO 01	121 ARCL 02	147 LASTX	173 CHS
96*LBL 03	122 ARCL 03	148 X<> ^	174 X<> d
97 ISG 08	123 E1	149 X<> T	175 SCI IND d
98 GTO 05	124 -	150 9	176 ARCL d
99 " **DONE**"	125 CHS	151 -	177 X<> d
100 AVIEW	126 RCL d	152 CHS	178 CLX
101 TONE 3	127 SCI IND Y	153 FIX 0	179 X<>]
102 TONE 4	128 ARCL Y	154 RND	180 "`^"
103 TONE 5	129 RCL ^	155 CF 29	181 X<>]
104 TONE 8	130 STO L	156 10^X	182 CLA
105 TONE 7	131 CLX	157 ARCL X	183 STO [
106 TONE 8	132 X<>]	158 R^	184 END
107 CLA	133 "`^"	159 STO d	
108 PSE	134 X<> T	160 CLX	
109 RCL 07	135 X<>]	161 X<>]	

Hangman (French version)

Whodunit – Swap Disks

And here's the French version for you (the shortest of the three):

01*LBL "HANGX"	41 ARCL 13	81*LBL 04	121 X=Y?
02 AON	42 ARCL 14	82 PSE	122 GTO 08
03 "-----"	43 ALENG	83 FC? 23	123 X<>Y
04 ASTO 10	44 E	84 GTO 04	124 RCL \
05 32	45 -	85 RCL d	125 RCL [
06 STO 11	46 STO 12	86 AVIEW	126 GTO 02
07 1,008	47 X<> L	87 STO d	127*LBL 07
08 95	48 CHS	88 RDN	128 X<> Z
09*LBL 00	49 AROT	89 ATOX	129 RCL \
10 STO IND Y	50 ATOX	90 X<> Z	130 RCL [
11 ISG Y	51 STO 00	91 STO \	131 CLA
12 GTO 00	52 ATOX	92 X<>Y	132 ARCL 10
13 "-----"	53 STO IND L	93 STO [133 ATOX
14 RCL [54 X<> Z	94 RDN	134 ASTO 10
15 X<> d	55 XTOA	95 RDN	135 X=0?
16 "MOT : "	56 XTOA	96 ALENG	136 GTO 09
17 AVIEW	57 AROT	97 X<>Y	137 STO 11
18 CLA	58 ,	98*LBL 05	138 RDN
19 STOP	59 RCL \	99 POSA	139 GTO 02
20 X<> d	60 RCL [100 X<0?	140*LBL 08
21 ALENG	61*LBL 02	101 GTO 06	141 "GAGNE..."
22 X#0?	62 RCL 12	102 AROT	142 TONE 9
23 GTO 01	63 SIGN	103 E	143 GTO 10
24 TIME	64 RDN	104 ST+ T	144*LBL 09
25 FRC	65 CLA	105 +	145 "PENDU..."
26 R-D	66*LBL 03	106 -	146 E
27 FRC	67 RCL IND L	107 XTOA	147 STO 11
28 E6	68 XTOA	108 ATOX	148 TONE 0
29 *	69 RDN	109 STO IND Y	149*LBL 10
30 66	70 DSE L	110 SF 05	150 AVIEW
31 MOD	71 GTO 03	111 GTO 05	151 PSE
32 INT	72 RCL 00	112*LBL 06	152 CLA
33 "PA"	73 XTOA	113 FC?C 05	153 ARCL 13
34 SEEKPTA	74 RDN	114 GTO 07	154 ARCL 14
35 GETREC	75 " " "	115 RDN	155 " " "
36*LBL 01	76 RCL 11	116 AROT	156 RCL 11
37 ASTO 13	77 XTOA	117 CLX	157 XTOA
38 ASHF	78 RDN	118 RCL 12	158 CLD
39 ASTO 14	79 AVIEW	119 E	159 END
40 CLA	80 CF 23	120 -	

Hangman w/ Subroutines

[John Raush – PPCCJ V7N2 p43 ; \(March 1980\)](#)

At first glance you would think that word games like HANGMAN would be easy to write on an alphanumeric calculator. Hardly! Our friends from Corvallis did not have character manipulation in mind when they designed the HP-41C. For the most part, the alphanumeric capabilities of the HP-41C are intended for putting and labeling of numeric output. Games like HANGMAN or more practical applications like a hexadecimal calculator have to go through all kinds of awkward data movements in and out of the ALPHA register to accomplish their tasks. My first reaction was to not write any of the word games for the HP-41C. After two months had passed I could no longer resist. There is no longer such a thing as a word encoder since letters can be stored as letters. There is a need to store the words in compact strings so they can be written on a data card. Efforts to write a "foolproof" program for building word cards resulted in lengthy routines that are not really necessary. Therefore, the very short routine "BW" (Build Words) is what I ended up with. It simply prompts you to enter 16 strings of 6 characters each and then puts the strings onto a data card. As you enter the words, they must be separated with a ":" and the last word must be followed by a ".". The following example should clarify the use of this program:

<u>Keystrokes</u>	<u>Display</u>
XEQ "BW"	1?
THOU:S, R/S	2?
HALT:M, R/S	3?
IND:TH, R/S	4?
INE:OW, R/S	5?
N:BUSI, R/S	6?
NESS. , R/S	7?
R/S	CARD

Note that there is no requirement that all 16 registers be used. Pressing R/S without entering any alpha data terminates the build process. All register contents beyond the "." will be ignored by the extract routine.

As was done with the HP-67/ 97, a subroutine was written to extract the words from the strings. The routine for the HP-41C turned out to be better than I had expected. The routine is designed to permit words with up to 15 letters. Typically, word games use less than the maximum (HANGMAN has a maximum of 10 letters). The register usage by the "EW" (extract word) routine is as follows :

R00 – Letter count	R16 – Loop control and indirect use
R01 – Individual letters	R17 – word list starts
...	...
R15 – not used	R32 – end of word list

Note that word programs that use words with less than 15 letters can use the registers that would normally be occupied by letters. Registers beyond the last letter in a word are unaltered.

The first time the "EW" routine is executed flag 05 must be set. Flag 05 is used by the routine to indicate an empty word list and will request a data card when it is set. When the "EW" routine has set flag 05 your program should not clear it if the routine is to be executed again .

You may also note that the word separator (":") and word list terminator (".") can easily be changed by altering steps 20 and 24 . It is also a fairly simple matter to alter both the "BW" and "EW" routines to use a shorter word list if required. No subroutine calls are made by the "EW" routine .

The "EW" routine is designed to handle word lists where the last word in the list need not be complete. The first part of the word could be on one card and the second part on another card . I don't recommend that this technique be used for most word games since you must always begin with the first word in a list and it is usually better to have a number of different word lists to choose from. Keep in mind that an "endless" word list is possible though.

The HANGMAN program started out with a fairly simple routine to keep track of guessed letters by maintaining a fractional string of 1's for each letter in the word. This number is created by steps 10-18.

As each letter is guessed, its corresponding 1 is changed to a 0. This makes it a simple matter to test for completion. The main loop in the program that compares a guessed letter to each letter in a word starts at line 74. The matter of what kind of wrong guess indication went from one idea to another when I finally decided to leave it up to the individual user. The program allows for six wrong guesses. If a seventh wrong guess it made, the word is displayed. the player is considered "hanged" and the next word is extracted.

The six indicators for wrong guesses must be stored in registers 13-38 as six alphabetic characters prior to execution. By storing "1" - "6" in the registers a wrong guess count will be maintained. However, those of you with printer or some other means of obtaining the display characters created by the BLDSPC function will probably want to store a set of displays that is ideally suited for HANGMAN. A short routine "SC" is provided to build a set of these characters in registers 13-38. It's a good idea to write these registers to a data card since a printer is not always at hand.

A guess of "*" will result in a "premature" hanging . This allows for giving up should you have the inclination. Once the HANGMAN program has been started it is not necessary to press any keys except letters to be guessed.

The HANGMAN program requires one memory module and a card reader. However, it is very easy to modify the programs to run on a basic HP-41C. Two modifications are required so that, the HANGMAN program and "EW" routine will both fit in storage. The first is to modify the "BW" (Build Word) routine to use only to register for the word list instead of 16. Change line 5 from 1.016 to 1.01. It is not necessary to change the "EW" routine. Next change lines 86-91 in the HANGMAN program to "|-B" and ARCL X. This provides a wrong guess count instead of obtaining the indicators from registers 13-38. If you want to play around a little more it is possible to pick up a few bytes here and there (for example, R15 is not used). The first place that might get attention is the use of six registers to hold the wrong guess indicators. Be my guest!

I had a working version that stored all six characters in one register but the program lines required to extract the proper character took more than overall storage than the current version. It also took more time and the program required three cards. I should also point out that labels 00 and 02 are intentionally local even though they are out of the 112 byte range of some GTO's near the end of the program. A rapid GTO is not needed in these points (in fact, a delay is preferred) and less storage is used..

Program listing:

01 LBL "HM" ;Main	29 ASTO X	57 RCL IND 13	84 X=Y?
02 FIX 0	30 AVIEW	58 RCL 14	85 GTO 08
03 CF 23	31 "*" ;1 asterisk	59 X=Y?	86 32
04 CF 29	32 ASTO Y	60 GTO 05	87 +
05 SF 05	33 X=Y?	61 >"*" ; one	88 LASTX
06 LBL 00	34 GTO 08	asterisk	89 >" " ;Append
07 XEQ "EW"	35 STO 14	62 GTO 06	space
08 CLX	36 RCL 00	63 LBL 05	90 X#Y?
09 STO 12	37 1 E3	64 SF 06	91 ARCL IND Y
10 9	38 /	65 1	92 GTO 01
11 1/X	39 1	66 ST- 11	93 LBL 07
12 10	40 +	67 ARCL IND 13	94 BEEP
13 RCL 00	41 STO 13	68 LBL 06	95 GTO 00
14 -	42 CF 06	69 AVIEW	96 LBL 08
15 10^X	43 CLA	70 ISG 13	97 CLA
16 *	44 LBL 03	71 GTO 03	98 RCL 00
17 FRC	45 10	72 RCL 11	99 0
18 STO 11	46 ST* 11	73 X=0?	100 LBL 09
19 CLA	47 RCL 11	74 GTO 07	101 X=Y?
20 ARCL 00	48 INT	75 RCL 00	102 GTO 10
21 >" LTRS"	49 X<>Y	76 10^X	103 1
22 LBL 01	50 /	77 /	104 +
23 AON	51 FRC	78 STO 11	105 ARCL IND X
24 AVIEW	52 X#0?	79 1	106 GTO 09
25 LBL 02	53 GTO 04	80 FC?C 06	107 LBL 10
26 PSE	54 ARCL IND 13	81 ST+ 12	108 AVIEW
27 FC?C 23	55 GTO 06	82 7	109 PSE
28 GTO 02	56 LBL 04	83 RCL 12	110 "HANGED"

111 AVIEW	20 X#Y?	34 GTO 00	38 AON
112 GTO 00	21 GTO 02	35 LBL 02	39 "DONE"
113 END	22 ISG Z	36 17.032	40 PSE
<u>01 LBL "BC"</u>	23 GTO 00	37 STO 16	41 AOFF
02 96	24 LBL 01	38 CLD	42 END
03 XEQ 01	25 RCL 00	39 RDTAX	<u>01 LBL "WSET2"</u>
04 STO 33	26 WDTAX	40 GTO 00	02 AON
05 6	27 RTN	41 LBL 03	03 "LOADING
06 XEQ 01	28 LBL 02	42 SF 05	REGS"
07 STO 34	29 ">6"	43 END	04 PSE
08 4	30 AVIEW	<u>01 LBL "WSET1"</u>	05 AOFF
09 XEQ 01	31 PSE	02 AON	06 "EQUINO"
10 STO 35	32 CLA	03 "LOADING	07 ASTO 00
11 5	33 GTO 00	REGS"	08 "X:POLY"
12 XEQ 01	34 END	04 PSE	09 ASTO 01
13 STO 36	<u>01 LBL "EW"</u>	05 AOFF	10 "NOMIAL"
14 1	02 CLX	06 "CASTLE"	11 ASTO 02
15 XEQ 01	03 STO 00	07 ASTO 00	12 ":STIPU"
16 STO 37	04 FS?C 05	08 ":FIRET"	13 ASTO 03
17 64	05 GTO 02	09 ASTO 01	14 "LATE:L"
18 XEQ 01	06 LBL 00	10 "RUCK:K"	15 ASTO 04
19 STO 38	07 " " ;5 spaces	11 ASTO 02	16 "ATTICE"
20 RTN	08 ARCL IND 16	12 "HAKI:C"	17 ASTO 05
21 LBL 01	09 ASTO X	13 ASTO 03	18 ":SYMME"
22 0	10 ASHF	14 "HIPMUN"	19 ASTO 06
23 X<>Y	11 ASTO IND 16	15 ASTO 04	20 "TRY:KN"
24 BLDSPEC	12 " " ;1 space	16 "K:SAVV"	21 ASTO 07
25 END	13 ARCL X	17 ASTO 05	22 "OTHOLE"
<u>01 LBL "BW"</u>	14 ASHF	18 "Y:AMID"	23 ASTO 08
02 FIX 0	15 ASTO X	19 ASTO 06	24 ":PURSU"
03 CF 23	16 CLA	20 "SHIPS:"	25 ASTO 09
04 CF 29	17 ASTO Y	21 ASTO 07	26 "IT:TOR"
05 1.016	18 X=Y?	22 "ARCTIC"	27 ASTO 10
06 STO 00	19 GTO 01	23 ASTO 08	28 "US:REF"
07 STO Z	20 ". ";1 period	24 ":JOURN"	29 ASTO 11
08 AON	21 ASTO Y	25 ASTO 09	30 "EREE:B"
09 CLA	22 X=Y?	26 "AL:DIV"	31 ASTO 12
10 ASTO Y	23 GTO 03	27 ASTO 10	32 "UZZARD"
11 LBL 00	24 " ";1 colon	28 "ISION:"	33 ASTO 13
12 ARCL Z	25 ASTO Y	29 ASTO 11	34 ":CHEWA"
13 ">?"	26 X=Y?	30 "FROSTB"	35 ASTO 14
14 PROMPT	27 RTN	31 ASTO 12	36 "BLE."
15 FC?C 23	28 ISG 00	32 "ITE:PI"	37 ASTO 15
16 GTO 01	29 .	33 ASTO 13	38 AON
17 ASTO IND Z	30 STO IND 00	34 "LLBOX:"	39 "DONE"
18 ASHF	31 GTO 00	35 ASTO 14	40 PSE
19 ASTO X	32 LBL 01	36 "BULLY."	41 AOFF
	33 ISG 16	37 ASTO 15	42 END

Master Mind (w/ Timer)

Julian Perry – DataFile V2N1 p26 (Jan/Feb 1983)

HP-41CX MASTERMIND

This short program allows you to play 9 digit mastermind on a 41C with X-Functions and Time Module. The program is very fast and is only 223 bytes long. The program listing includes both time module and X-functions instructions but these can easily be removed (see below).

INSTRUCTIONS

- 1.) XEQ "MMIND"
- 2.) You will then be prompted by "DIGITS?, 1-9" Enter the code length then press R/S.
- 3.) Next you will see "MAX. VALUE ?" Enter the maximum value of each digit in the code then press R/S.
- 4.) You will then be prompted by "GUESS ?" Enter your guess containing the digits from 0 to the maximum value, and then press R/S. If you key in a guess which is too long the first digits will be lost, and if you do not key in enough the guess will be padded with zeros on the left hand side.
- 5.) The program will then return your guess followed by a dash and two other digits (eg. "12345-2.1") The first digit indicates the number of correctly placed digits in your guess and the second indicates the number of digits in your guess that are in the code, but are in the wrong place (corresponding to black and white key pegs).
- 6.) Enter your new guess and press R/S. (Don't take too long because if you have got the time module you are being timed).
- 7.) Keep guessing until you get all of the digits in the right place, when this happens you will be told how many attempts you made and if you press R/S you will see how much think time you used.

NOTES:

If you do not have an X-functions module then delete lines 2 and 3, and change lines 96-99 to: 96 "A8CD", 97 ARCL 12, 98 ASHF.

If you do not have a time module delete lines 5, 6, 7, 26, 27, 28, 29, 36, 38 and 113-119. Also insert a few lines to prompt for a seed for the random number generator.

Line 35 is TONE p (120)	Line 40 is STO N
Line 57 is RCL M	Line 62 is STO M
Lines 68 and 70 are NOPs (text 0)	
Line 75 is DSE N)	Line 47 is STO M

Program listing:

1 <u>LBL "MMIND"</u>	42 10^X	83 ,
2 16	43 /	84 LBL 09
3 PSIZE	44 FRC	85 RCL IND Y
4 CLRG	45 STO 12	86 ST- IND Z
5 CLX	46 RCL 15	87 X>0?
6 STOPSW	47 STO M	88 CLX
7 SETSW	48 ENTER^	89 +
8 "DIGITS? 1-9"	49 LBL 10	90 ISG X
9 PROMPT	50 RCL Z	91 GTO 09
10 ABS	51 E1	92 E1
11 INT	52 *	93 /
12 9	53 FRC	94 ST+ 10
13 X>Y?	54 LASTX	95 FIX IND 13
14 RDN	55 INT	96 CLA
15 STO 13	56 RCL 14	97 ARCL 12
16 "MAX. VALUE?"	57 RCL M	98 ATOX
17 PROMPT	58 R-D	99 ATOX
18 ABS	59 FRC	100 >"-"
19 INT	60 *	101 FIX 1
20 0	61 FRC	102 ARCL 10
21 X>Y?	62 STO M	103 AVIEW
22 RDN	63 X<> L	104 RCL 10
23 E	64 INT	105 RCL 13
24 +	65 X#Y?	106 X#Y?
25 STO 14	66 GTO 03	107 GTO 01
26 TIME	67 DSE IND Y	108 PSE
27 *	68 ""	109 FIX 0
28 FRC	69 ISG IND X	110 CLA
29 STO 15	70 ""	111 ARCL 11
30 ADV	71 ,9	112 >" ATTEMPTS"
31 ADV	72 ST- 10	113 PROMPT
32 "GUESS?"	73 RDN	114 RCLSW
33 AVIEW	74 LBL 03	115 E2
34 LBL 01	75 DSE N	116 *
35 TONE 4	76 GTO 10	117 FIX 2
36 RUNSW	77 RCL 14	118 "TIME="
37 STOP	78 E	119 ATIME24
38 STOPSW	79 ST+ 11	120 AVIEW
39 RCL 13	80 -	121 END
40 STO N	81 E3	
41 STO 10	82 /	

Master Mind (The Sequel)

Peter Gatenby – DataFile V5N7 p12 (November 1986)

Another mastermind program may seem unnecessary so soon after Julian Perry's (V2N1 P26) -- I just happen to think that mine is better! MINDD was composed independently of Julian's MMIND though there are inevitably some similarities.

To use MINDD as listed here proceed as follows:

XEQ "MINDD"

At the prompt "**SEED?**" enter a positive number less than 1 and R/S.

On the prompt "**DIGITS?**" enter an integer, 1 through 10 and R/S. This number determines the number of digits in the sequence to be guessed.

On the prompt "**GUESS?**" enter a sequence of digits and R/S. Your guess will be AVIEW'd while it is compared with the target sequence. The score for your guess will be displayed as (guess)*a.b - where a is the number of guess digits which match target digits for magnitude and position and b is the number of the remainder which match for magnitude only. RIS and try another guess.

When an exact match is achieved you will hear BEEP and see:

"(guess) WINS IN (number of guesses)".R/S, see:"DIGITS?", and have another go.

Example.

Type	Display
XEQ MINDD	SEED?
0.1; R/S	DIGITS?
3; R/S	GUESS?
123; R/S	123
	123*0.2
R/S; 214; R/S	214*0.1
R/S; 315; R/S	315
	315*1.0
R/S; 636; R/S	636
R/S	636*0.1
R/S; 787; R/S	787
	787*0.1
R/S; 372; R/S	372
	312 WINS IN 6
RIS	DIGITS?

Program Remarks:

- The EF/EM module is needed. Registers 00 to d+10 and flags 05 to d+4 are used, where "d" is the number of digits. Flags 00 to 04 cannot be used because the annunciators would allow cheating.

- Line 02 calls the subroutine at lines 142-7 which sets flags 26, 28, 29, establishes FIX 0 and DEG modes and clears all other flags. Line 144 is hex F7,00,00,00,20,00,80,00
- Line 21 calls the random number generator at lines 133-41 which is modified from the routine RNDM of the Standard ROM. Any other would do which delivers to X a positive number less than 1 and does not interfere with registers 01 to 20, line 07 might need modification.
- Line 102 includes initial and terminal spaces.
- A false score of 1.0 is returned for a guess which scores 10×0.1 .
- Unlike Julian's program there is no control of the maximum value of target digits.
- If you inadvertently R/S at "GUESS?" without having entered a guess you will see the ALPHA DATA error message. You can rectify the situation by GTO 22; R/S to get back to "GUESS?".

Ed's note: Obviously you can integrate the subroutines into the main body of the code, since they're only called once. That would save two FAT entries... and chances are your ROM already has a random number generator anyway.

Program listing:

1 LBL "MINDD"	26 ASTO IND 15	51 X#0?
2 XEQ "FLGS"	27 DSE 15	52 GTO 10
3 RCLFLAG	28 GTO 01	53 ARCL X
4 STO 19	29 LBL 22	54 RDN
5 "SEED?"	30 0	55 FRC
6 PROMPT	31 STO 18	56 DSE 15
7 STO 00	32 RCL 20	57 GTO 02
8 LBL 21	33 STO 15	58 GTO 11
9 "DIGITS?"	34 RCL 19	59 LBL 10
10 PROMPT	35 RCLFLAG	60 RDN
11 STO 17	36 "GUESS?"	61 RCL 15
12 4,004	37 TONE 8	62 INT
13 +	38 PROMPT	63 5
14 STO 20	39 ISG 16	64 -
15 STO 15	40 LBL 00	65 10^X
16 0	41 CLA	66 *
17 STO 16	42 CF 29	67 ARCL X
18 CF 29	43 RCL 17	68 LBL 11
19 LBL 01	44 10^X	69 AVIEW
20 CLA	45 /	70 ASTO 01
21 XEQ "RDM"	46 LBL 02	71 ASTO 03
22 E1	47 E1	72 ASHF
23 +	48 *	73 ASTO 02
24 INT	49 ENTER^	74 ASTO 04
25 ARCL X	50 INT	75 RCL 20

76 STO 15	101 ARCL 04	126 ARCL 04
77 CLA	102 >"WINS IN "	127 >"*"
78 ARCL 01	103 ARCL 16	128 FIX 1
79 ARCL 02	104 BEEP	129 ARCL 18
80 LBL 03	105 PROMPT	130 TONE 9
81 RCL 15	106 GTO 21	131 PROMPT
82 POSA	107 LBL 06	132 GTO 22
83 X#0?	108 RCL 20	133 END
84 GTO 04	109 STO 15	
85 ATOX	110 LBL 07	1 <u>LBL "RDM"</u>
86 SF IND 15	111 FS? IND 15	2 RCL 00
87 ISG 18	112 GTO 09	3 9821
88 LBL 00	113 RCL IND 15	4 *
89 GTO 05	114 POSA	5 ,2211327
90 LBL 04	115 X<0?	6 +
91 E	116 GTO 09	7 FRC
92 AROT	117 AROT	8 STO 00
93 LBL 05	118 ATOX	9 RTN
94 DSE 15	119 ,1	10 <u>LBL "FLGS"</u>
95 GTO 03	120 ST+ 18	11 F7,00,00,00,20,00,8
96 RCL 17	121 LBL 09	0,00
97 RCL 18	122 DSE 15	12 RCL M
98 X#Y?	123 GTO 07	13 STO d
99 GTO 06	124 CLA	14 END
100 ARCL 03	125 ARCL 03	

Master Mind (w/ Colors)

Whodunit – Swap Disks

Maybe the question here is who hasn't written a Master Mind program at some point in time? Beware! : The idea of a ROM solely dedicated to Master Mind is slowly getting stronger ;-)

Well, for completion sake – here are two more from the French collection in the Swap Disks. The first one has a twist: like the real-life game, it speaks of colors instead of numbers. It's also quite wordy, which should add to the gaming experience - assuming *vous parlez français, mais naturellement ;-)*

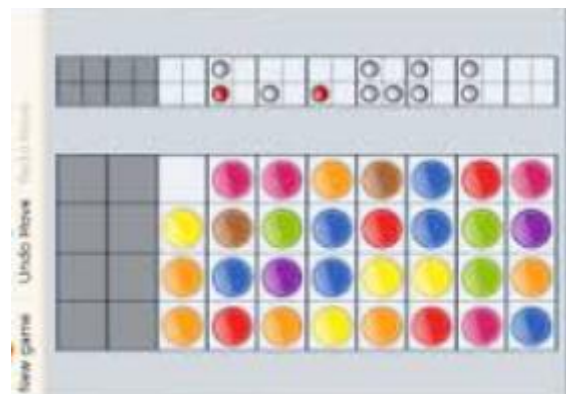
Both versions use the peripheral printer, supposedly for a neat presentation of the results - unfortunately it's not documented of course...

Program listing:

1 LBL "MIND"	32 -	64 PROMPT
2 CF 21	33 "LES COULEURS SO"	65 CF 10
3 CF 00	34 >"NT LES CH"	66 "*MASTERMIND*"
4 SF 27	35 AVIEW	67 SF 12
5 FIX 00	36 >"IFFRES 0 A "	68 AVIEW
6 CF 29	37 ARCL X	69 CF 12
7 RCL 00	38 AVIEW	70 " "
8 SIGN	39 PSE	71 ARCL 01
9 X#0?	40 "OUI"	72 >" TROUS, "
10 LASTX	41 ASTO Y	73 ARCL 02
11 FRC	42 "LES CHIFFRES D'"	74 >" COULEURS"
12 "N? (0<N<1)"	43 >"UN"	75 FS? 55
13 X=0?	44 AVIEW	76 PRBUF
14 PROMPT	45 >"E COMBINAISO"	77 FS? 55
15 STO 00	46 AVIEW	78 PRA
16 "Nb DE TROUS? (2"	47 >"N DOIVENT-IL"	79 " POUVANT SE R"
17 >" A 9)"	48 AVIEW	80 >"EPETER"
18 PROMPT	49 >"S ETRE TOUS "	81 FS? 55
19 STO 01	50 AVIEW	82 FS? 00
20 8	51 >"DIFFERENTS ?"	83 FS? 30
21 +	52 AON	84 PRA
22 E3	53 PROMPT	85 " NE SE REPETA"
23 /	54 ASTO X	86 >"NT PAS"
24 9	55 X=Y?	87 FS? 55
25 +	56 SF 00	88 FC? 00
26 STO 03	57 AOFF	89 FS? 30
27 "Nb DE	58 LBL A	90 PRA
COULEURS?"	59 FS? 55	91 ADV
28 >" (2 A 10)"	60 SF 21	92 RCL 03
29 PROMPT	61 "PRENEZ DE QUOI "	93 ENTER
30 STO 02	62 >"ECRIRE"	94 LBL 00
31 E	63 FC? 55	95 ASTO IND Y

96 ISG Y	148 "SUSPENSE..."	200 LBL 08
97 GTO 00	149 AVIEW	201 RDN
98 LBL 27	150 RCL 01	202 RCL IND Z
99 RCL 00	151 10^X	203 X#Y?
100 9821	152 /	204 GTO 13
101 *	153 RCL 03	205 CLX
102 .211327	154 9.009	206 E1
103 +	155 +	207 ST+ Y
104 FRC	156 X<>Y	208 ST+ IND T
105 STO 00	157 CLA	209 ISG 08
106 TONE 06	158 LBL 04	210 "" ;F0
107 FS?C 10	159 E1	211 LBL 13
108 RTN	160 *	212 ISG T
109 RCL 02	161 FRC	213 GTO 08
110 *	162 LASTX	214 ISG L
111 INT	163 INT	215 GTO 07
112 FC? 00	164 STO IND Z	216 LBL 10
113 GTO 13	165 ARCL X	217 R^
114 RCL 03	166 RDN	218 R^
115 STO T	167 ISG Y	219 RCL IND X
116 LBL 03	168 GTO 04	220 E1
117 RDN	169 ASTO 04	221 X<=Y?
118 RCL IND Z	170 ASHF	222 ST- IND Z
119 X#Y?	171 ASTO 05	223 ISG Z
120 GTO 01	172 8	224 GTO 10
121 R^	173 RCL 03	225 RCL 07
122 R^	174 ST+ Y	226 ST- 08
123 GTO 27	175 ENTER	227 FS? 55
124 LBL 01	176 ENTER	228 GTO 11
125 ISG T	177 LBL 05	229 "ESSAI "
126 GTO 03	178 R^	230 ARCL 06
127 RDN	179 R^	231 >": "
128 LBL 13	180 ISG Y	232 ARCL 04
129 STO IND Y	181 "" ;F0	233 ARCL 05
130 X<>Y	182 RCL IND Y	234 TONE 05
131 ISG X	183 RCL IND Y	235 AVIEW
132 GTO 27	184 X=Y?	236 PSE
133 CLX	185 ISG 07	237 SF 10
134 STO 06	186 "" ;F0	238 "_____"
135 LBL 99	187 ISG Z	;"\1C"
136 ISG 06	188 GTO 05	239 ASTO X
137 "" ;F0	189 9.009	240 RCL 07
138 CLX	190 RCL 03	241 XEQ 27
139 STO 08	191 +	242 ST+ X
140 STO 07	192 X<> L	243 INT
141 CF 21	193 ENTER	244 SIGN
142 CLA	194 ENTER	245 "O"
143 ARCL 01	195 LBL 07	246 ASTO X
144 >"T"	196 RCL Z	247 CLA
145 ARCL 02	197 R^	248 AVIEW
146 >"C A VOUS"	198 RCL IND L	249 PSE
147 PROMPT	199 ENTER	250 RCL 08

251 X<> L	289 ARCL 04	327 AVIEW
252 X=0?	290 ARCL 05	328 XEQ 13
253 GTO 13	291 ACA	329 TONE 07
254 X<> L	292 19	330 TONE 06
255 R^	293 RCL 01	331 4
256 R^	294 ST+ X	332 LOG
257 SIGN	295 -	333 LOG
258 LBL 13	296 45	334 XEQ 13
259 X<> L	297 LBL 06	335 TONE 06
260 XEQ 13	298 ACCHR	336 RCL 06
261 R^	299 DSE Y	337 "EN "
262 R^	300 GTO 06	338 ARCL X
263 XEQ 13	301 8	339 >" ESSAI"
264 PSE	302 RCL 08	340 1
265 PSE	303 XEQ 13	341 -
266 GTO 01	304 0	342 X#0?
267 LBL 13	305 RCL 07	343 >"S"
268 X=0?	306 XEQ 13	344 AVIEW
269 RTN	307 PRBUF	345 STOP
270 LBL 02	308 LBL 01	346 LBL 13
271 ARCL Y	309 RCL 01	347 TONE 05
272 AVIEW	310 RCL 07	348 TONE 06
273 TONE 05	311 X=Y?	349 TONE 06
274 TONE 06	312 GTO 09	350 TONE 05
275 PSE	313 GTO 99	351 TONE 05
276 DSE X	314 LBL 13	352 TONE 07
277 GTO 02	315 X=0?	353 TONE 06
278 RTN	316 RTN	354 4
279 LBL 11	317 X<>Y	355 LOG
280 SF 21	318 LBL 14	356 LOG
281 ADV	319 ACCRH	357 TONE 06
282 CLA	320 DSE Y	358 TONE 07
283 9	321 GTO 14	359 TONE 06
284 RCL 06	322 RTN	360 TONE 05
285 X<=Y?	323 LBL 09	361 TONE 05
286 " "	324 ADV	362 END
287 ARCL X	325 SF 12	
288 >": "	326 " TROUVE"	



Master Mind (w/ Printer))

Whodunit – Swap Disks

Program listing:

1 LBL "MM"	42 RCL 07	85 ISG 08	123 XEQ 07
2 11	43 E1	86 GTO 02	124 RCL 10
3 PSIZE	44 X>Y?	87 RCL 10	125 E3
4 CLRG	45 " "	88 STO 08	126 /
5 LBL 05	46 ARCL Y	89 LBL 03	127 E
6 ADV	47 >" "	90 RCL IND 08	128 +
7 "_____"	48 ACA	91 FRC	129 STO 08
;"\04	49 ASTO X	92 ST- IND 08	130 PRBUF
\00\80\81"	50 "ESSAI "	93 X#0?	131 ISG 07
8 RCL M	51 ARCL X	94 GTO 04	132 GTO 01
9 "MOT?"	52 CF 21	95 LASTX	133 3
10 STO d	53 AON	96 POSA	134 SKPCHR
11 TONE 73	54 TONE 06	97 X<0?	135 CF 21
12 STOP	55 PROMPT	98 GTO 04	136 LBL 06
13 AOFF	56 AVIEW	99 AROT	137 "PERDU"
14 6	57 AOFF	100 ATOX	138 X=Y?
15 ALENG	58 ASTO X	101 SIGN	139 "GAGNE"
16 X>Y?	59 RCL 00	102 ST+ 09	140 AVIEW
17 GTO 05	60 X=Y?	103 XTOA	141 BEEP
18 ASTO 00	61 GTO 06	104 +	142 ASTO X
19 " "	62 RCL 10	105 CHS	143 SF 21
20 ARCL X	63 ALENG	106 AROT	144 SF 12
21 >"	64 X#Y?	107 LBL 04	145 CLA
LETTRES"	65 GTO 01	108 DSE 08	146 ARCL 00
22 AVIEW	66 SF 12	109 GTO 03	147 ACA
23 STO 10	67 SF 21	110 CF 12	148 FMT
24 E3	68 ACA	111 RCL 09	149 CF 12
25 /	69 6	112 INT	150 ACX
26 STO 07	70 -	113 "_____0a	151 ADV
27 ST+ 07	71 SKPCHR	<"	152 GTO 05
28 E	72 3	;"\10\F2\1	153 LBL 07
29 +	73 XROM	4\180a<"	154 RCL M
30 STO 08	29,23	114 X#0?	155 3
31 ST+ 07	74 LBL 02	115 XEQ 07	156 LBL 08
32 CLA	75 RCL IND 08	116 ,	157 SKPCOL
33 ARCL 00	76 ATOX	117 X<> 09	158 X<>Y
34 LBL 00	77 X#Y?	118 FRC	159 ACSPEC
35 ATOX	78 GTO 04	119 E1	160 X<>Y
36 STO IND Y	79 .1	120 *	161 DSE Z
37 X<>Y	80 ST+ 09	121 "_____<"	162 GTO 08
38 ISG X	81 ST+ IND 08	;"\10\F3\F	163 END
39 GTO 00	82 E	7\FF\FF\F	
40 LBL 01	83 LBL 04	<"	
41 CLA	84 XTOA	122 X#0?	

Super Master Mind

Whodunit – Swap Disks

And since there aren't two without three... here's yet another variation on the same theme, courtesy of our anonymous friends from the Swap Disks department...

Note that for whatever reason this version uses OUTA instead of the more common PRA functions, perhaps it was meant for a different peripheral (like the Video Interface?)... we'll never know.

Program listing:

1 LBL "VMIND"	36 INT	71 X=Y?
2 CLRG	37 STO 06	72 GTO 08
3 17	38 RCL 01	73 RDN
4 PSIZE	39 X<>Y	74 RCL 13
5 SF 17	40 X=Y?	75 X=0?
6 " * SUPER MA"	41 XEQ 00	76 GTO 10
7 OUTA	42 RCL 02	77 RDN
8 "STER MIND *"	43 RCL 06	78 CLA
9 CF 17	44 X=Y?	79 ARCL X
10 OUTA	45 XEQ 00	80 CF 17
11 CLRG	46 RCL 03	81 OUTA
12 CF 29	47 RCL 06	82 12
13 "X?"	48 X=Y?	83 RCL 13
14 PROMPT	49 XEQ 00	84 X=Y?
15 STO 01	50 RCL 04	85 XEQ 11
16 FIX 00	51 RCL 06	86 XEQ A
17 5	52 X=Y?	87 LBL A
18 STO 00	53 XEQ 00	88 SF 17
19 STO 11	54 RCL 05	89 "GUESS?"
20 8	55 RCL 06	90 FIX 00
21 STO 09	56 X=Y?	91 OUTA
22 "5 POSITIONS/8 C"	57 XEQ 00	92 PROMPT
23 ">OLORS"	58 RCL 06	93 CLA
24 OUTA	59 X=0?	94 ARCL X
25 XEQ 00	60 XEQ 00	95 ">" "
26 LBL 00	61 RCL 06	96 OUTA
27 8	62 STO IND 00	97 STO 09
28 RCL 01	63 DSE 00	98 STO 10
29 9821	64 GTO 00	99 LBL 02
30 *	65 LBL 01	100 XEQ 09
31 .211327	66 FIX 01	101 RCL IND 00
32 +	67 RCL 11	102 ABS
33 FRC	68 X<> 00	103 INT
34 STO 01	69 X<> 08	104 STO IND 00
35 *	70 RCL 11	105 X#Y?

106 GTO 03	134 LBL 06	162 STO 12
107 1	135 DSE 07	163 RTN
108 ST+ 08	136 GTO 05	164 LBL 10
109 CHS	137 LBL 07	165 " "
110 ST* IND 00	138 DSE 00	166 >" N.B"
111 LBL 03	139 GTO 04	167 OUTA
112 DSE 00	140 1	168 GTO A
113 GTO 02	141 ST+ 13	169 LBL 11
114 RCL 09	142 GTO 01	170 FIX 00
115 STO 10	143 LBL 08	171 "SOLUTION : "
116 RCL 11	144 FIX 00	172 RCL 01
117 STO 00	145 CLA	173 ABS
118 LBL 04	146 ARCL 09	174 ARCL X
119 RCL 11	147 >"*"	175 RCL 02
120 STO 07	148 ARCL 13	176 ABS
121 XEQ 09	149 >" GUESSES"	177 ARCL X
122 RCL IND 00	150 BEEP	178 RCL 03
123 X<0?	151 OUTA	179 ABS
124 GTO 07	152 OFF	180 ARCL X
125 LBL 05	153 LBL 09	181 RCL 04
126 RCL 12	154 RCL 10	182 ABS
127 RCL IND 07	155 INT	183 ARCL X
128 X#Y?	156 10	184 RCL 05
129 GTO 06	157 /	185 ABS
130 .1	158 STO 10	186 ARCL X
131 ST+ 08	159 FRC	187 OUTA
132 ST+ IND 07	160 10	188 OFF
133 GTO 07	161 *	189 END

Mastermind for the HP-41C

This program is by [Wouter Peters](#) and is used here by permission.

This program is supplied without representation or warranty of any kind. Wouter Peters and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

Mastermind is originally a kind of board game for two. The board of the game consist of a series of four-holed rows. One is covered to hide it from one player. The other rows have four small 'marker' holes at one side.

The first player puts together a row of four colored pegs. There are usually six colours available. Two or more pegs of the same color are allowed. This array is hidden from the other player.

The other player has to discover the colors and positions of the pegs by putting together reasoned guesses of usually four colored pegs (less than four pegs in one guess is allowed though, only the placed pegs will be evaluated). A guess is evaluated by the first player and is rewarded a white marker for every peg of the right color that is placed in a wrong position and a black marker for every peg of the right color that's in the right place. These marker pegs are plugged into the marker-holes. The aim is to find the right configuration in as few as possible guesses.

Mastermind - HP 41C as the composer/evaluator

The composition of a hidden row and evaluation of the guesses can be programmed into a HP 41C. The display can't show colours so we'll use figures instead. I've chosen to enter a guess as a four-digit number. This limits the number of 'colours' to 9. The figure '0' is reserved for an empty hole in a row (incomplete guess). **The number of colours** isn't hard coded though and all numbers of colours between 1-9 can be chosen (see Register use below). This is sufficient, as the classic Mastermind is played with pegs of six colours.

The HP can only show one row of pegs with markers so either we must use the printer or use pen and paper to represent the board to keep track of the earlier guesses. This program can be run on a HP 41 C with or without a printer connected.

The calculator will generate a hidden array of four figures randomly chosen within the range of possible 'colors' and then prompt the player for his or her first guess. A guess is entered as a four-digit number. It is then parsed and processed. The program will return the appropriate white (0) or black (*) markers or will beep and return 'RIGHT' on a right guess.

A few notes on the program listing:

1. I haven't found a suitable character on my PC for the 'lazy T' associated with APPEND. The few times it occurs in the listing below (in lines 101, 111, 119 and 132) I've typed it as 'It'. So these are APPEND <space>, APPEND '*', APPEND '0' and APPEND 'RIGHT'

2. A few Extended Functions are used: RCLFLAG, STOFLAG and REGMOVE. The first two are not really necessary. They have only a cosmetic function: to restore the numerical display format back to the state from before Mastermind was run (Mastermind needs a FIX 0 display). REGMOVE offers an easy way to copy registers 1-4 into 11-14 (Line 044, 045), but it can easily be rewritten to a series of RCL and STO statements to get the same result: a copy of Reg 1-4 into Reg 11-14.

Size= 21

Use of registers:

R00= Random seed
R01= code A
R02= code B
R03= code C
R04= code D
R05= guess A
R06= guess B
R07= guess C
R08= guess D
R09= # of right guessed places
R10= loop counter #1
R11= scratch A
R12= scratch B
R13= scratch C
R14= scratch D
R15= address pointer #1
R16= address pointer #2
R17= # of right guessed figures
R18= loop counter #2
R19= # of possible figures
R20= num.display format flags

How to play the game

1. Decide on the number of different figures you want to use and store this into register 19. To play in the classic setup: 6 STO 19
2. XEQ "MM" will generate an hidden array and prompts you for the first guess ("?" in the display)
3. enter for instance 1234 and press R/S . After your guess is parsed you'll see "1,2,3,4," scrolling through the display instead of the flying goose (a trick using a bug, the ignore error flag (25) and a flag operation on a nonexisting flag (any > 55), after evaluation of your guess it's marked, your marked guess displayed (for instance "1,2,3,4, *OO" one in a right place, two right figures in wrong places), a TONE is executed (no printer) or the output printed and the program STOPS. Now you can study your guess(es) deduct a new one enter it and press R/S again.
4. Repeat 3 until you've got it right. The calculator will BEEP and you get a display saying that you had it right (for instance: '6,4,3,2 RIGHT'). The program winds up (restores the original numerical display format) and halts at the END, so you can simply start another play by pressing R/S.

Program listing:

```

001 LBL "MM"
002 RCLFLAG
003 STO 20
004 FIX 0
005 4
006 STO 10
007 LBL 01
008 XEQ 11
009 RCL 19
010 *
011 INT
012 E
013 +
014 STO IND 10
015 DSE 10
016 GTO 01
017 CF 22
018 LBL A
019 FC? 22
020 ?
021 FC? 22
022 PROMPT
023 E4
024 /
025 STO L
026 XEQ 12
027 STO 05
028 XEQ 12
029 STO 06
030 XEQ 12
031 STO 07
032 XEQ 12
033 STO 08
034 CLA
035 ARCL 05
036 ARCL 06
037 ARCL 07
038 ARCL 08
039 FS? 55
040 CF 21
041 SF 25
042 AVIEW
043 SF 99
044 1,011004
045 REGMOVE
046 5
047 STO 15
048 11
049 STO 16
050 4
051 STO 10
052 STO 18
053 STO 17

054 STO -17
055 LBL 02
056 RCL IND 15
057 RCL IND 16
058 X=Y?
059 GTO 03
060 E
061 ST+ 16
062 DSE 10
063 GTO 02
064 GTO 04
065 LBL 03
066 ISG 17
067 STO X
068 PI
069 STO IND 16
070 LBL 04
071 E
072 ST+15
073 11
074 ST 16
075 4
076 STO 10
077 DSE 18
078 GTO 02
079 STO 09
080 ST- 09
081 RCL 01
082 RCL 05
083 X=Y?
084 ISG 09
085 STO X
086 RCL 02
087 RCL 06
088 X=Y?
089 ISG 09
090 STO X
091 RCL 03
092 RCL 07
093 X=Y?
094 ISG 09
095 STO X
096 RCL 04
097 RCL 08
098 X=Y?
100 STO X
101 "lt "
102 RCL 09
103 ST- 17
104 STO 10
105 X=0?
106 GTO 06
107 4

108 X=Y?
109 GTO 09
110 LBL 05
111 "lt*"
112 DSE 10
113 GTO 05
114 LBL 06
115 RCL 17
116 X=0?
117 GTO 08
118 LBL 07
119 "ltO"
120 DSE 17
121 GTO 07
122 LBL 08
123 FS? 55
124 SF 21
125 AVIEW
126 FC? 55
127 TONE 7
128 CF 22
129 STOP
130 GTO A
131 LBL 09
132 "ltRIGHT"
133 FS? 55
134 SF 21
135 AVIEW
136 BEEP
137 ADV
138 GTO 10
139 LBL 11
140 RCL 00
141 9821
142 *
143 .211327
144 +
145 FRC
146 STO 00
147 RTN
148 LBL 12
149 LASTX
150 FRC
151 10
152 *
153 INT
154 RTN
155 LBL 10
156 RCL 20
157 36,41
158 STOFAG
159 END

```

MasterMind, Variant 1, for HP-41CX

Kai Schröder, http://www.achim-und-kai.de/kai/hp41cx/superhirn_var1_e.html

Everybody should know "MasterMind", the logic game by Parker. One player provides a hidden code of colors or figures, and the other one's object is in as few as possible guesses to find out this sequence. The only hints are black and white markers. A black marker is given for a correct color (or figure) in the correct position, a white one determines a correct color (or figure) in an incorrect position.

In this program figures are used instead of colors, and the HP-41CX provides the hidden code and assesses the guess of the player. This has two advantages: first, you don't need a second willingly person, and second, the HP-41CX is guaranteed perfect in its assessment :-)

Example :

Given is a code consisting of four figures in the range from 0 through 5. Each figure is allowed to appear only once in the sequence. A possible flow of game could be as follows:

Try	Black	White	Code
			2 4 0 1
5	4	0	2 4 0 1
4	0	4	0 1 2 4
3	0	3	1 3 4 0
2	0	3	4 0 3 2
1	1	2	4 3 2 1

Course of Game :

On starting the program first, a seed for the random number generator must be entered. The player can choose between a code of 4, 5, or 6 figures. When "POSITIONS ?" is displayed the desired number of positions must be entered. The range of figures can be chosen from 0 through 9, but always starts with 0 (inclusive) upwards. When "0-.. ?" appears in the display the user has to enter the upper boundary (inclusive). Now the player is asked, whether the figures are allowed to appear only once in the code or several times, too. If the figures are allowed to appear only once "Y" must be pressed, otherwise "N". Now the number of permutations is calculated and displayed. While the hidden sequence is generated "MIXING" is shown in the display. After this "CODE READY" is displayed.

Shortly later "INPUT" is displayed, and the stopwatch starts running. Now it's the turn of the player to enter a code and to press R/S. The stopwatch pauses and the HP-41CX assesses the code. When this is accomplished the BEEP sounds and "RESULT:" is displayed. Pressing R/S first the number of black markers and then the number of the white ones are displayed. After this the prompt appears again and the next code is to be entered by the player.

On determining the exact code, the required tries and time are displayed. If the player did give up - I can't imagine this! ;-) - or runs out of time the hidden code can be revealed by

XEQ 40. A code must be entered for authorization purposes. If the code is correct the sequence is displayed, otherwise all data are removed.

You want to know the code ??? ;-) . . . Simply read the source code carefully, then you will see, which code must be entered! ;-)

Program listing:

```

01 LBL "GAME5"
002 "
MASTERMIND"
003 AVIEW
004 ,037
005 CLRGX
006 RCLFLAG
007 STO 08
008 " "
009 RCL [
010 STO d
011 ,
012 SETSW
013 SIGN
014 STO 12
015 2
016 STO 13
017 3
018 STO 14
019 4
020 STO 15
021 5
022 STO 16
023 11
024 STO 01
025 16
026 STO 32
027 17
028 STO 31
029 23
030 STO 10
031 48
032 STO 23
033 STO 24
034 STO 25
035 STO 26
036 STO 27
037 STO 28
038 LBL 18
039 CF 22
040 " RNG-
SEED : "
041 PROMPT
042 FC? 22
043 GTO 18
044 STO 09
045 LBL 07
046 CF 22
047 " 4-6
POS."
048 AVIEW
049 PSE
050 6
051
"POSITIONS ?"
052 PROMPT
053 FC? 22
054 GTO 07
055 INT
056 X>Y?
057 GTO 07
058 3
059 X<>Y?
060 X<=Y?
061 GTO 07
062 STO 33
063 ST+ 31
064 ST+ 32
065 DSE X
066 E3
067 /
068 STO 37
069 LBL 06
070 CF 22
071 "
FIGURES MAX."
072 "FROM 0-
9"
073 AVIEW
074 PSE
075 9
076 " 0-.. ?"
077 PROMPT
078 FC? 22
079 GTO 06
080 INT
081 ABS
082 X>Y?
083 GTO 06
084 2
085 X<>Y?
086 X<=Y?
087 GTO 06
088 STO 36
089 ISG X
090 ""
091 STO 34
092 LBL 17
093 CF 23
094 "
FIGURES"
095 AVIEW
096 PSE
097 " ONLY
ONCE "
098 " Y/N
:"
099 AON
100 PROMPT
101 AOFF
102 FC? 23
103 GTO 17
104 74
105 ATOX
106 X=Y?
107 SF 01
108 FS? 01
109 GTO 08
110 RCL 36
111 ISG X
112 ""
113 RCL 33
114 Y^X
115 GTO 10
116 LBL 08
117 ISG 36
118 ""
119 RCL 36
120 RCL 33
121 -
122 E3
123 /
124 ST+ 36
125 E
126 LBL 09
127 RCL 36
128 INT
129 *
130 DSE 36
131 GTO 09
132 LBL 10
133 "
POSSIBLE "
134 "
PERMUTATIONS:"
135 AVIEW
136 PSE
137 VIEW X
138 PSE
139 PSE
140 LBL 00
141 "
MIXING"
142 AVIEW
143 LBL 01
144 RCL 09
145 R-D
146 FRC
147 R-D
148 FRC
149 STO 09
150 E5
151 *
152 INT
153 RCL 34
154 MOD
155 FS? 01
156 GTO 03
157 LBL 05
158 ST+ IND
10
159 ISG 10
160 ""
161 ISG 37
162 GTO 01
163 GTO 04
164 LBL 03
165 5
166 +
167 FS? IND X
168 GTO 01
169 SF IND X
170 5
171 -
172 GTO 05
173 LBL 40
174 STOPSW
175 CLA
176 AON
177 STOP
178 AOFF
179 6
180 ALENG
181 X#Y?
182 GTO 41
183 ATOX
184 48
185 -
186 DATE
187 DOW
188 XY?
not equal
189 GTO 41
190 SF 00
191 "CODE : "
192 23
193 STO 35

```


194	STO 31	248	STO 07	300	RCL 33	355	BEEP
195	RCL 33	249	RCL IND	301	ST- 05	356	" RESULT
196	ST+ 31	01		302	RTN	:	"
197	GTO 42	250	AROT	303	LBL 20	357	AVIEW
198	LBL 43	251	RCL IND	304	E	358	STOP
199	AVIEW	00		305	ST+ 30	359	FIX 0
200	STOP	252	POSA	306	AROT	360	CF 29
201	GTO 41	253	STO 04	307	RCL IND	361	"BLACK :
202	LBL 04	254	X<0?	00		"	
203	CF 27	255	GTO 14	308	POSA	362	ARCL 02
204	" CODE	256	X=0?	309	STO 06	363	AVIEW
READY"		257	GTO 15	310	ST+ 30	364	PSE
205	AVIEW	258	RCL 04	311	ST+ 04	365	"WHITE :
206	PSE	259	STO 30	312	RCL 33	"	
207	17	260	AROT	313	DSE X	366	ARCL 03
208	STO 00	261	RCL 00	314	"" (NOP)	367	AVIEW
209	RCL 08	262	RCL 04	315	RCL 30	368	PSE
210	STOFLAG	263	+	316	X>Y?	369	CLA
211	LBL 31	264	STO 05	317	GTO 14	370	RCL 33
212	AON	265	RCL 32	318	RCL 06	371	RCL 02
213	TONE 7	266	RCL 05	319	AROT	372	X=Y?
214	RUNSW	267	X>Y?	320	RCL 00	373	GTO 10
215	" INPUT	268	XEQ 12	321	RCL 04	374	2,003
:	"	269	RCL IND	322	+	375	CLRGX
216	PROMPT	05		323	RCL 07	376	6
217	STOPSW	270	POSA	324	+	377	ST- 31
218	AOFF	271	X=0?	325	STO 05	378	GTO 31
219	17	272	GTO 20	326	RCL 32	379	LBL 10
220	STO 35	273	RCL 05	327	RCL 05	380	"TRIES :
221	LBL 13	274	12	328	X>Y?	"	
222	ATOX	275	-	329	XEQ 12	381	ARCL 29
223	STO IND	276	STO 05	330	RCL IND	382	AVIEW
35		277	FS? IND	05		383	PSE
224	ISG 35	05		331	POSA	384	FIX 4
225	"" (NOP)	278	GTO 20	332	X=0?	385	SF 29
226	RCL 31	279	SF IND 05	333	GTO 21	386	"
227	RCL 35	280	GTO 16	334	RCL 05	REQUIRED"	
228	X<Y?	281	LBL 14	335	12	387	AVIEW
229	GTO 13	282	ISG 00	336	-	388	PSE
230	6	283	""	337	STO 05	389	"TIME IN
231	ST+ 31	(NOP)		338	FS? IND	H.:"	
232	LBL 11	284	ISG 01	05		390	AVIEW
233	23	285	"" (NOP)	339	GTO 21	391	PSE
234	STO 35	286	RCL 32	340	SF IND 05	392	RCLSW
235	CLA	287	RCL 00	341	GTO 16	393	VIEW X
236	LBL 42	288	X>Y?	342	LBL 21	394	LBL 41
237	RCL IND	289	GTO 30	343	ISG 07	395	RCL 08
35		290	GTO 11	344	"" (NOP)	396	STOFLAG
238	XTOA	291	LBL 15	345	GTO 20	397	,037
239	ISG 35	292	ISG 02	346	LBL 30	398	CLRGX
240	"" (NOP)	293	""	347	ISG 29	399	CLST
241	RCL 31	(NOP)		348	"" (NOP)	400	SETSW
242	RCL 35	294	GTO 14	349	RCL 08	401	CLA
243	X<Y?	295	LBL 16	350	STOFLAG	402	END
244	GTO 42	296	ISG 03	351	17		
245	FS? 00	297	"" (NOP)	352	STO 00		
246	GTO 43	298	GTO 14	353	11		
247	SIGN	299	LBL 12	354	STO 01		

Inverse MasterMind, for HP-41CX

Kai Schröder, http://www.achim-und-kai.de/kai/hp41cx/superhirn_var1_e.html

MasterMind again - but in this variant the HP-41CX guesses the hidden code provided by the player! The code must consist of four characters out of a range from "A" through "F" (inclusive). (Of course, it's from a programming point of view no problem to extend the number of positions, but this would last too long!) Every character is allowed to appear **only once** in the code. The several time appearances of characters is no real problem, but in this case the number of permutations increases very quickly and - believe me :-) - it would be no fun to wait for the next proposed code! Therefore, in this program version these features have not been implemented.

Computing time is up to 20 min, if you have the TURBO alteration it's up to about 10 min, until the HP-41CX proposes the next code. To prevent battery voltage from decreasing too much, flag 49, battery voltage flag, is checked during program execution. If this happens program execution is terminated and the HP-41CX powers off itself. On power on "BATTERY" is displayed to indicate the low battery voltage. Program execution can't be continued.

Course of Game :

On starting the program, a seed for the random number generator has to be entered. Now the HP-41CX calculates a first guess. After the sound it is shown in the display. Pressing R/S, now "B,W : " appears in the display and the number of black and white markers, separated by decimal comma must be entered. Again, pressing R/S starts the computation of the next sequence. When the correct code is determined, the HP-41CX displays the number of tries.

Program listing:

001 <u>LBL</u>	024 MOD	049 RCL 01	074 /
<u>"GAME5b"</u>	025 FS? IND X	050 FS? IND X	075 STO 05
002	026 GTO 00	051 GTO 17	076 LBL 18
"MASTERMIND 2"	027 SF IND X	052 SF IND X	077 FS? 49
003 AVIEW	028 65	053 RCL 02	078 GTO 23
004 PSE	029 +	054 FS? IND X	079 RCL 05
005 ,034	030 XTOA	055 GTO 17	080 21
006 CLRGX	031 ISG 01	056 SF IND X	081 +
007 CLX	032 GTO 00	057 RCL 03	082 RCL IND X
008 X<>F	033 11	058 FS? IND X	083 STO 06
009 E	034 STO 04	059 GTO 17	084 CLA
010 " RNG-	035 ASTO IND	060 LBL 05	085 RCL 05
SEED : "	X	061 RCL 08	086 11
011 PROMPT	036 ,005	062 31	087 +
012 STO 00	037 STO 00	063 -	088 ARCL IND
013 ,003	038 STO 01	064 RCL IND X	X
014 STO 01	039 STO 02	065 65	089 XEQ 22
015 CLA	040 STO 03	066 +	090 LBL 01
016 LBL 00	041 XEQ 22	067 STO IND	091 RCL IND
017 RCL 00	042 GTO 16	08	08
018 R-D	043 LBL 15	068 ISG 08	092 POSA
019 FRC	044 XEQ 22	069 GTO 05	093 RCL 08
020 STO 00	045 CLX	070 RCL 04	094 31
021 E5	046 X<>F	071 11	095 -
022 *	047 RCL 00	072 -	096 INT
023 6	048 SF IND X	073 E3	097 X#Y?

098 GTO 02	138 CLA	176 BEEP	216 LBL 22
099 ISG 07	139 LBL 04	177 AVIEW	217 31,034
100 ""	140 RCL IND	178 STOP	218 STO 08
101 LBL 02	08	179 LBL 06	219 CLX
102 ISG 08	141 XTOA	180 TONE 8	220 STO 07
103 GTO 01	142 ISG 08	181 CF 22	221 RTN
104 RCL 06	143 GTO 04	182 " B,W :"	222 LBL 23
105 INT	144 ASTO IND	183 PROMPT	223 SF 11
106 RCL 07	04	184 FC? 22	224 OFF
107 X#Y?	145 GTO 16	185 GTO 06	225 "
108 GTO 17	146 LBL 17	186 STO 06	BATTERY"
109 XEQ 22	147 ISG 00	187 FRC	226 AVIEW
110 LBL 03	148 GTO 15	188 E1	227 STOP
111 RCL IND	149 FS? 49	189 *	228 LBL 19
08	150 GTO 23	190 RCL 06	229 E
112 POSA	151 ,005	191 +	230 RCL 04
113 X<0?	152 STO 00	192 INT	231 E1
114 GTO 02	153 ISG 01	193 2	232 -
115 RCL 08	154 GTO 15	194 X>Y?	233 CLA
116 31	155 ,005	195 GTO 07	234 FIX 0
117 -	156 STO 00	196 4	235 CF 29
118 INT	157 STO 01	197 RCL Z	236 ARCL X
119 X=Y?	158 ISG 02	198 X>Y?	237 " TR"
120 GTO 02	159 GTO 15	199 GTO 07	238 X>Y?
121 ISG 07	160 ,005	200 RCL 06	239 GTO 08
122 ""	161 STO 00	201 4	240 "↑Y"
123 LBL 02	162 STO 01	202 X=Y?	241 GTO 09
124 ISG 08	163 STO 02	203 GTO 19	242 LBL 08
125 GTO 03	164 ISG 03	204 RCL 04	243 "↑IES"
126 RCL 06	165 GTO 15	205 E1	244 LBL 09
127 FRC	166 LBL 16	206 +	245 AVIEW
128 E1	167 CLA	207 RCL Z	246 CLA
129 *	168 ARCL IND	208 STO IND Y	247 FIX 4
130 RCL 07	04	209 GTO 15	248 SF 29
131 X#Y?	169 32	210 LBL 07	249 ,034
132 GTO 17	170 XTOA	211 TONE 5	250 CLRGX
133 ISG 05	171 XTOA	212 "WRONG	251 CLX
134 GTO 18	172 XTOA	INPUT"	252 X<>F
135 ISG 04	173 XTOA	213 AVIEW	253 CLST
136 ""	174 4	214 PSE	254 END
137 XEQ 22	175 AROT	215 GTO 06	

Grand MasterMind (at last)

Tom Rice - PPCCJ V12N3 p10 (March 1985)

Here is an HP-41 program that plays the part of the codemaker for Grand Master Mind. The program is 1,048 bytes long, and does not contain synthetic instructions.

To play the game, execute GMM and enter a seed from 0 to 1 when prompted. The 41 will then find a four-position random code, with each position containing a one digit integer and one digit decimal.

Numbers 1 to 5 are used, but 0 can be used by deleting lines 84-85; you can also make the game more difficult (or easier) by increasing (or decreasing) the value in line 71. As the 41 prompts for each guess position, supply it with a number in the form X.Y.

The 41 will display "CLUE: ABC", where A is the number of correct integer-decimal pairs in the right position, B is the number of correct pairs in the wrong position, and C is the number of integers or decimals (only) in the right position. If a printer is used, each set of guesses along with its clue will be printed.

If you want this program recorded on magnetic cards, send five (5) cards (or \$2.59) and an SASE (or \$1.00) to:

Tom Rice (10921)
N. 6505 Suther 1 in
Spokane, WA 99208

Program Description:

This version of the popular game Master Mind has the complication of using pairs of colors and shapes. The object of the game is to correctly guess the four pairs in the hidden code in as few guesses as possible, using the clues given by the computer.

The computer gives three types of clues. A black (1, or multiple of 100) is given for a correct pair in the right position. A white (*, or a multiple of 10) is given for a correct pair in the wrong position. A blue (+, or a multiple of 1) is given for a correct integer or decimal (only) in the correct position. The computer compares in such a way that a maximum number of clues is given, following a strict one-to-one correspondence.

Necessary Accessories: 2 Memory modules, or CV/CX. Limits and Warnings: Flags 05-29 are used, but flags 11, 14, 23-25 will be cleared and 26 set if the program is aborted immediately after a clue is given. If a printer is used, you must execute (R/S) immediately after printing.

References:

The Official Master Mind Handbook, Ault, Leslie H., New York:Signet, 1976.
The Official Master Mind Puzzle Book, Ault, Leslie H., Ph.D., NewYork: Signet, 1978.

Further Discussion:

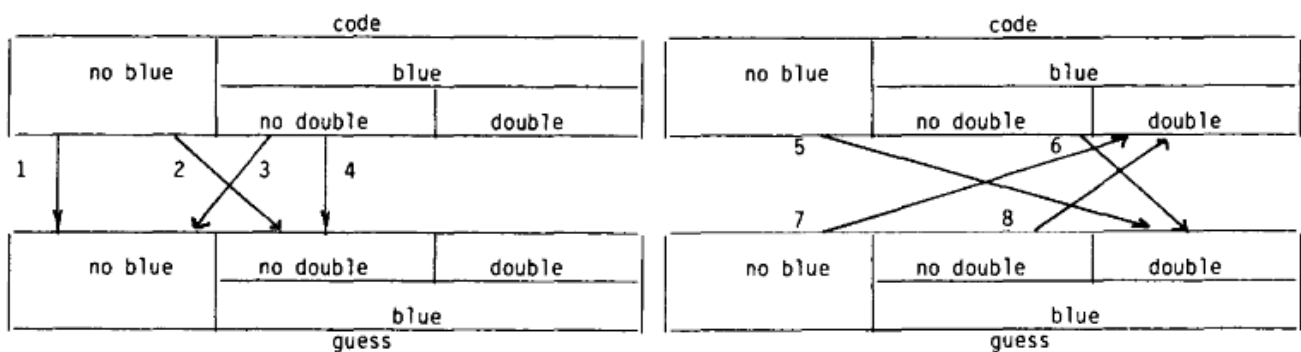
Only the digits 0 - 5 are valid for either the integer or decimal part of each guess position; the computer will not check for invalid guesses.

With 5 possible integer and decimal values, there are $(5 \times 5)^4 = 390,625$ permutations. If you wish to use zeros as blanks, delete lines 84 and 85. This will give $(6 \times 6)^4 = 1,679,616$ permutations. If you do not value your sanity, you may increase the value of the number in line 71 to further increase the number of permutations.

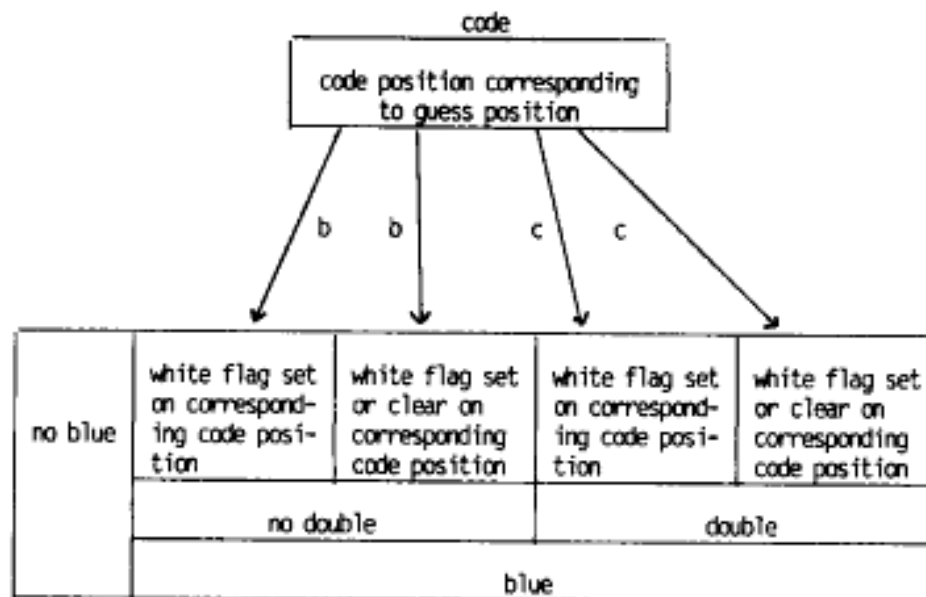
If you do not have a printer available, you will want to keep track of your guesses and clues on paper or on a Grand Master Mind game board. Any value can correspond to any color or shape, but one possible combination is given below.

<u>Number</u>	<u>Corresponding Color</u>	<u>Corresponding Shape</u>
1	Yellow	Triangle
2	Red	Square
3	Blue	Rectangle
4	Green	Hexagon
5	Black	Star

The computer goes through a rather involved routine in order to make the comparisons. First, the computer searches for blacks then possible blues, then doubles in both the code and the guess. The blue flag must be set in order for the double flag to be set. The double flag may actually be for a triple, quadruple, or two doubles. The computer will give as many blues as possible by making the comparisons for whites in the following order:



Comparisons 5 - 8 have sub-parts as shown below for comparisons 5 and 6 (subpart a matches to guess (or code) positions in which the corresponding code (or guess) position already has a white flag).



Comparison # 9 matches for whites in any combination not yet checked.

Keystroke Solution:

<u>Display</u>	<u>Input</u>	<u>Function</u>	<u>Comments</u>
		(XEQ) SIZE 018	
		(XEQ) "GMM"	
SEED ?	0.5	(R/S)	Generate random code
POSITION 1	1.2	(R/S)	Guess A, Position 1
POSITION 2	4.1	(R/S)	Guess A, Position 2
POSITION 3	5.4	(R/S)	Guess A, Position 3
POSITION 4	5.3	(R/S)	Guess A, Position 4
CLUE:111		(R/S)	1 black, 1 white, 1 blue
POSITION 1	4.1	(R/S)	Guess B, Position 1
POSITION 2	3.4	(R/S)	Guess B, Position 2
POSITION 3	4.1	(R/S)	Guess B, Position 3
POSITION 4	2.4	(R/S)	Guess B, Position 4
CLUE:11		(R/S)	1 white, 1 blue
POSITION 1	5.1	(R/S)	Guess C, Position 1
POSITION 2	5.1	(R/S)	Guess C, Position 2
POSITION 3	3.3	(R/S)	Guess C, Position 3
POSITION 4	4.2	(R/S)	Guess C, Position 4
CLUE:11		(R/S)	1 white, 1 blue
POSITION 1	5.1	(R/S)	Guess D, Position 1
POSITION 2	4.1	(R/S)	Guess D, Position 2
POSITION 3	5.3	(R/S)	Guess D, Position 3
POSITION 4	1.3	(R/S)	Guess D, Position 4
CLUE:121		(R/S)	1 black, 2 whites, 1 blue
POSITION 1	5.3	(R/S)	Guess E, Position 1
POSITION 2	4.1	(R/S)	Guess E, Position 2
POSITION 3	5.1	(R/S)	Guess E, Position 3
POSITION 4	1.5	(R/S)	Guess E, Position 4
CLUE:400		(R/S)	4 blacks; you got it!

Same problem with Printer (in MAN position)

					<u>Clue</u>	<u>Reason</u>
Code:	1.1	1.3	1.1	2.5		
Guess A:	2.1	4.3	1.3	1.1	***	The 1.1 in position 3 of the code matches the 1.1 in the guess, thus allowing a blue for position 1.
Guess B:	2.5	1.1	4.1	1.1	***	The 1.1's get two whites, which can be matched in any order. The 2.5 gets the other white.
Guess C:	2.5	1.1	6.1	1.3	***	This is similar to A. The 1.1 in position 1 of the code matches the 1.1 in the guess, so position 3 can get a blue. Note that the computer did not reject the 6.1.
Guess D:	1.1	1.3	1.1	2.5	####	Perfect match; four blacks.

Keystroke Solution:

<u>Display</u>	<u>Input</u>	<u>Function</u>	<u>Comments</u>
		(XEQ) SIZE 018	
		(XEQ) "GMH"	
SEED ?	0.7	(R/S)	Generate random code
POSITION 1	2.1	(R/S)	Guess A, Position 1
POSITION 2	4.3	(R/S)	Guess A, Position 2
POSITION 3	1.3	(R/S)	Guess A, Position 3
POSITION 4	1.1	(R/S)	Guess A, Position 4
			Printout: 2.1 4.3 1.3 1.1 ***
POSITION 1	2.5	(R/S)	Guess B, Position 1
POSITION 2	1.1	(R/S)	Guess B, Position 2
POSITION 3	4.1	(R/S)	Guess B, Position 3
POSITION 4	1.1	(R/S)	Guess B, Position 4
			Printout: 2.5 1.1 4.1 1.1 ***
POSITION 1	2.5	(R/S)	Guess C, Position 1
POSITION 2	1.1	(R/S)	Guess C, Position 2
POSITION 3	6.1	(R/S)	Guess C, Position 3
POSITION 4	1.3	(R/S)	Guess C, Position 4
			Printout: 2.5 1.1 6.1 1.3 ***
POSITION 1	1.1	(R/S)	Guess D, Position 1
POSITION 2	1.3	(R/S)	Guess D, Position 2
POSITION 3	1.1	(R/S)	Guess D, Position 3
POSITION 4	2.5	(R/S)	Guess D, Position 4
			Printout: 1.1 1.3 1.1 2.5 ####

Data Registers used:

00	Flag clearing counter	08	Code position 4
	Double flag setting counter	09	Guess position 1
	Primary position counter	10	Guess position 2
01	Double flag setting counter	11	Guess position 3
	Secondary position counter	12	Guess position 4
02	Tertiary position counter	13	Clue value
03	4 Or -4 code-guess register conversion	14	5.008 or 9.012 position counter value
04	14 (or 18) guess (or code) to blue flag conversion	15	9.012 or 5.008 position counter value
05	Code position 1	16	10, 11, 10, 11, 10, 11, 32 test label value
06	Code position 2		
07	Code position 3	17	3, 4, 3, 4 test label value

Flags Used

05 Black or white for code pos. #1	17 Double and blue for guess pos. #1
06 Black or white for code pos. #2	18 Double and blue for guess pos. #2
07 Black or white for code pos. #3	19 Double and blue for guess pos. #3
08 Black or white for code pos. #4	20 Double and blue for guess pos. #4
09 Black or white for guess pos. #1	21 Changed to set if printer is present
10 Black or white for guess pos. #2	22 (not used)
11 Black or white for guess pos. #3	23 Blue might be given for pos. #1
12 Black or white for guess pos. #4	24 Blue might be given for pos. #2
13 Double and blue for code pos. #1	25 Blue might be given for pos. #3
14 Double and blue for code pos. #2	26 Blue might be given for pos. #4
15 Double and blue for code pos. #3	27 (not used)
16 Double and blue for code pos. #4	28 Used primarily in test subroutines
29 Jump if no doubles are flagged (17-20) in guess	

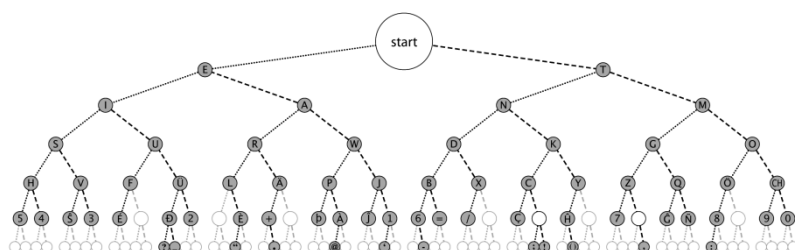
Program listing:

01 LBL "GMM"	11 ISG 00	21 GTO 16	31 STO 00
02 "SEED ?"	12 GTO 15	22 LBL 17	32 FIX 0
03 PROMPT	13 RCL 15	23 18	33 LBL 18
04 STO 04	14 STO 00	24 STO 03	34 CF IND 00
05 5.008	15 LBL 16	25 0	35 ISG 00
06 STO 15	16 XEQ 00	26 STO 13	36 GTO 18
07 STO 00	17 10	27 9.012	37 RCL 15
08 LBL 15	18 /	28 X<> 15	38 STO 00
09 XEQ 00	19 ST+ IND 00	29 STO 14	39 LBL 19
10 STO IND 00	20 ISG 00	30 5.029	40 RCL 00

41 8	93 E2	145 RCL 01	197 FS? IND 00
42 -	94 ST+ 13	146 8	198 GTO 31
43 "POSITION "	95 GTO 22	147 +	199 XEQ IND 16
44 ARCL X	96 LBL 01	148 SF IND X	200 FS?C 28
45 PROMPT	97 RCL 00	149 14	201 GTO 31
46 STO IND 00	98 RCL 03	150 RCL 03	202 RCL 15
47 ISG 00	99 +	151 X=Y?	203 STO 01
48 GTO 19	100 SF IND X	152 SF 29	204 LBL 29
49 RCL 14	101 LBL 22	153 GTO 25	205 RCL IND 00
50 STO 00	102 ISG 00	154 LBL 27	206 RCL IND 01
51 LBL 20	103 GTO 20	155 -4	207 X#Y?
52 RCL 00	104 5.007	156 STO 04	208 GTO 30
53 4	105 STO 00	157 10	209 FS? IND 01
54 +	106 XEQ 02	158 STO 16	210 GTO 30
55 RCL IND X	107 9.011	159 3	211 XEQ IND 17
56 RCL IND 00	108 STO 00	160 STO 17	212 FS?C 28
57 X=Y?	109 14	161 XEQ 05	213 GTO 30
58 GTO 21	110 STO 03	162 4	214 XEQ 08
59 -	111 XEQ 02	163 STO 17	215 GTO 31
60 FRC	112 GTO 27	164 XEQ 05	216 LBL 30
61 X=0?	113 LBL 02	165 11	217 ISG 01
62 GTO 01	114 RCL 00	166 STO 16	218 GTO 29
63 RCL IND Y	115 RCL 03	167 3	219 LBL 31
64 INT	116 +	168 STO 17	220 ISG 00
65 RCL IND 00	117 FC? IND X	169 XEQ 05	221 GTO 28
66 INT	118 GTO 25	170 4	222 RTN
67 X=Y?	119 RCL 00	171 STO 17	223 LBL 32
68 GTO 01	120 1.001	172 XEQ 05	224 RTN
69 GTO 22	121 +	173 GTO 33	225 LBL 33
70 LBL 00	122 STO 01	174 LBL 03	226 FC? 29
71 5	123 LBL 23	175 RCL 01	227 GTO 34
72 RCL 04	124 RCL IND 00	176 RCL 03	228 -4
73 9821	125 RCL IND 01	177 +	229 STO 04
74 *	126 X#Y?	178 FS? IND X	230 10
75 .21137	127 GTO 24	179 SF 28	231 STO 16
76 +	128 RCL 01	180 RTN	232 XEQ 35
77 FRC	129 RCL 03	181 LBL 04	233 11
78 STO 04	130 +	182 RCL 01	234 STO 16
79 10	131 FS? IND X	183 8	235 XEQ 35
80 *	132 GTO 26	184 +	236 LBL 34
81 INT	133 LBL 24	185 FS? IND X	237 X<> 14
82 X>Y?	134 ISG 01	186 SF 28	238 X<> 15
83 GTO 00	135 GTO 23	187 RCL 01	239 X<> 14
84 X=0?	136 LBL 25	188 RCL 03	240 18
85 GTO 00	137 ISG 00	189 +	241 STO 03
86 RTN	138 GTO 02	190 FC? IND X	242 4
87 LBL 21	139 RTN	191 SF 28	243 STO 04
88 SF IND 00	140 LBL 26	192 RTN	244 10
89 RCL 00	141 RCL 00	193 LBL 05	245 STO 16
90 4	142 8	194 RCL 14	246 XEQ 35
91 +	143 +	195 STO 00	247 11
92 SF IND X	144 SF IND X	196 LBL 28	248 STO 16

249 XEQ 35	301 FS?C 28	353 SF IND 00	405 SF IND 00
250 32	302 GTO 44	354 SF IND 01	406 RCL 15
251 STO 16	303 SF IND 00	355 10	407 STO 01
252 STO 17	304 RCL 15	356 ST+ 13	408 LBL 45
253 XEQ 05	305 STO 01	357 RTN	409 RCL IND 00
254 RCL 15	306 LBL 39	358 LBL 09	410 RCL IND 01
255 STO 00	307 RCL IND 00	359 SF IND 00	411 X#Y?
256 GTO 51	308 RCL IND 01	360 SF IND 01	412 GTO 49
257 LBL 35	309 X#Y?	361 RCL 01	413 FS? IND 01
258 RCL 14	310 GTO 43	362 RCL 04	414 GTO 49
259 STO 00	311 FS? IND 01	363 +	415 XEQ 12
260 XEQ 06	312 GTO 43	364 SF IND X	416 FS?C 28
261 RCL 14	313 XEQ 12	365 SF IND 02	417 GTO 49
262 STO 00	314 FS?C 28	366 20	418 RCL 15
263 XEQ 07	315 GTO 43	367 ST+ 13	419 STO 02
264 RCL 14	316 RCL 15	368 RTN	420 LBL 46
265 STO 00	317 STO 02	369 LBL 10	421 FS? IND 02
266 XEQ 13	318 LBL 40	370 RCL 00	422 GTO 48
267 RTN	319 FS? IND 02	371 RCL 03	423 RCL 01
268 LBL 06	320 GTO 42	372 +	424 RCL 04
269 FS? IND 00	321 RCL 01	373 RCL 04	425 +
270 GTO 38	322 RCL 04	374 -	426 RCL IND X
271 XEQ IND 16	323 +	375 FS? IND X	427 RCL IND 02
272 FS?C 28	324 RCL IND X	376 SF 28	428 X#Y?
273 GTO 38	325 RCL IND 02	377 RTN	429 GTO 48
274 RCL 15	326 X#Y?	378 LBL 11	430 RCL 02
275 STO 01	327 GTO 42	379 RCL 00	431 8
276 LBL 36	328 RCL 02	380 8	432 +
277 RCL IND 00	329 RCL 03	381 +	433 FC? IND X
278 RCL IND 01	330 +	382 FS? IND X	434 GTO 47
279 X#Y?	331 FC? IND X	383 SF 28	435 RCL 02
280 GTO 37	332 GTO 41	384 RCL 00	436 RCL 04
281 FS? IND 01	333 RCL 02	385 RCL 03	437 +
282 GTO 37	334 RCL 04	386 +	438 FC? IND X
283 RCL 01	335 +	387 RCL 04	439 GTO 50
284 RCL 04	336 FC? IND X	388 -	440 LBL 47
285 +	337 GTO 42	389 FC? IND X	441 XEQ 09
286 FC? IND X	338 LBL 41	390 SF 28	442 GTO 50
287 GTO 37	339 XEQ 09	391 RTN	443 LBL 48
288 XEQ 08	340 GTO 44	392 LBL 12	444 ISG 02
289 GTO 38	341 LBL 42	393 RCL 01	445 GTO 46
290 LBL 37	342 ISG 02	394 8	446 LBL 49
291 ISG 01	343 GTO 40	395 +	447 ISG 01
292 GTO 36	344 LBL 43	396 FC? IND X	448 GTO 45
293 LBL 38	345 ISG 01	397 SF 28	449 CF IND 00
294 ISG 00	346 GTO 39	398 RTN	450 LBL 50
295 GTO 06	347 CF IND 00	399 LBL 13	451 ISG 00
296 RTN	348 LBL 44	400 FS? IND 00	452 GTO 13
297 LBL 07	349 ISG 00	401 GTO 50	453 RTN
298 FS? IND 00	350 GTO 07	402 XEQ IND 16	454 LBL 51
299 GTO 44	351 RTN	403 FS?C 28	455 FS? IND 00
300 XEQ IND 16	352 LBL 08	404 GTO 50	456 XEQ 61

457 ISG 00	482 FIX 1	507 XEQ 63	532 E
458 GTO 51	483 CF 12	508 LBL 57	533 ST+ 13
459 RCL 14	484 SF 21	509 DSE 02	534 RTN
460 STO 00	485 SF 28	510 GTO 60	535 LBL 63
461 14	486 LBL 54	511 PRBUF	536 RCL 13
462 STO 03	487 RCL IND 00	512 GTO 17	537 RCL 04
463 LBL 52	488 ACX	513 LBL 58	538 /
464 FS? IND 00	489 1	514 35	539 FRC
465 XEQ 61	490 SKPCHR	515 ACCHR	540 STO 13
466 ISG 00	491 ISG 00	516 GTO 55	541 LASTX
467 GTO 52	492 GTO 54	517 LBL 59	542 INT
468 23.026	493 111	518 42	543 STO 02
469 STO 00	494 ST+ 13	519 ACCHR	544 RTN
470 LBL 53	495 100	520 GTO 56	545 LBL 64
471 FS?C IND 00	496 STO 04	521 LBL 60	546 RCL 13
472 XEQ 62	497 XEQ 63	522 43	547 "CLUE:"
473 ISG 00	498 LBL 55	523 ACCHR	548 ARCL X
474 GTO 53	499 DSE 02	524 GTO 57	549 AVIEW
475 RCL 14	500 GTO 58	525 LBL 61	550 TONE 0
476 STO 00	501 ,1	526 RCL 00	551 STOP
477 CF 11	502 STO 04	527 RCL 03	552 GTO 17
478 CF 14	503 XEQ 63	528 +	553 END
479 SF 26	504 LBL 56	529 CF IND X	;1048 BYTES
480 FC? 55	505 DSE 02	530 RTN	
481 GTO 64	506 GTO 59	531 LBL 62	

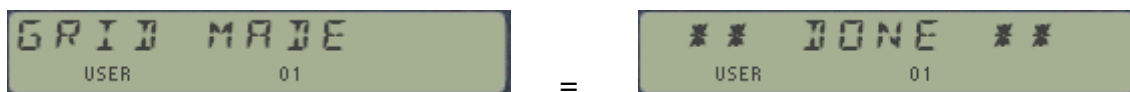


Sudoku Grid Generator

JM Baillard - - <http://hp41programs.yolasite.com/sudoku.php>

The HP-41 strikes back! Program **"GRID"** will prepare a Sudoku grid by randomly clearing some of the elements in a solved Sudoku pre-loaded for this purpose. The number of zeroed elements is defined in X before calling **GRID**. The final grid will be place in compact mode (each element a nibble of the mantissa) in registers R1-R9, ready for **"SUDOKU"** in case you have given in.

GRID uses a random-number generator **RNG** to determine which elements within each row will be cleared. RNG uses the TIME Module – so this function will fail if the timer is not present. When the execution ends the message **"GRID MADE"** is shown in the display.



User Instructions:

STACK	INPUTS	OUTPUTS
Y	/	/
X	N	/

where N is an integer between 1 and 81 to get a puzzle with N non-empty cells

Example: You want to get a sudoku with 28 non-empty cells, and you choose 1 as random seed.

28 XEQ "GRID" (attention: you need to stop the program after the call to RNG and replace the value with "1")

and we get the grid in registers R01 thru R09
(the integer part doesn't really matter):

R01 = 1.800600130
R02 = 1.003050004
R03 = 1.000900068
R04 = 1.008016000
R05 = 1.005003800
R06 = 1.006500003
R07 = 1.000361000
R08 = 1.600000307
R09 = 1.000005601

```

8 0 0 | 6 0 0 | 1 3 0
0 0 3 | 0 5 0 | 0 0 4
0 0 0 | 9 0 0 | 0 6 8
-----
0 0 8 | 0 1 6 | 0 0 0
0 0 5 | 0 0 3 | 8 0 0
0 0 6 | 5 0 0 | 0 0 3
-----
0 0 0 | 3 6 1 | 0 0 0
6 0 0 | 0 0 0 | 3 0 7
0 0 0 | 0 0 5 | 6 0 1

```

-So, the puzzle is:

If you don't solve the grid, one solution is in registers R19 thru R27
In this example, "SDK" gives another solution.

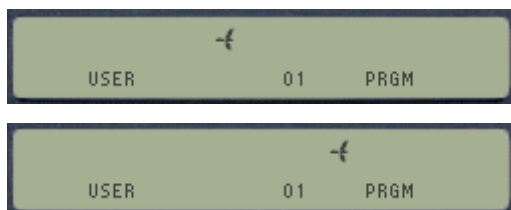
Notes:

Replace the value (5) in line 28 by a larger integer if you think that the original grid is not shuffled enough.

This routine does not always return a proper sudoku (i-e with a unique solution), especially if N is small. If it happens ... it's only by chance !

Behold: A left Goose on your LCD, and moving backwards !

Observant users will no doubt note that **GRID** is a left-handed program. The 41 knows that, and instructs the goose to behave accordingly – flipping left and running backwards! – all thanks to functions **LEFT** and **GOOSE**, written by Nelson F. Crowle, one of the authors of the AECROM module among other landmarks.



The usage of these functions is shown in the following code snippets. First GOOSE puts up the left goose on the display, and LEFT then should be called at every iteration of a loop, so that the display contents is shifted one position to the left. Note also the other combinations to amuse your friends, such as SF 25, SF 99 and AVIEW to rotate the contents of ALPHA right.

LBL 10	Left Goose flies left
GOOSE	
LBL 01	
LEFT	
0	
GTO 01	
LBL 11	Alpha Rotates Left
AVIEW	
LBL 02	
LEFT	
0	
GTO 02	
LBL 12	Right Goose Flies LEFT
50	
TOGF	
LBL 03	
LEFT	
0	
GTO 03	
LBL 13	Left Goose Flies RIGHT
GOOSE	
50	
TOGF	
LBL 04	
0	
GTO 04	

TOGF toggles flag status, also used in other programs (like Sea Battle, etc.)

Program listing:

00 LBL "GRID"

01 SIZE?	12 STO 01	24 STO 07
02 28	13 1.415973628	25 1.186294537
03 X>Y?	14 STO 02	26 STO 08
04 PSIZE	15 1.362185974	27 1.257316849
05 -SUDOKU 1C	16 STO 03	28 STO 09
06 81	17 1.674528193	29 7.00003
07 R^	18 STO 04	30 STO 15
08 -	19 1.521739486	31 5
09 INT	20 STO 05	32 STO 16
10 STO 00	21 1.839461752	33 CLD
11 1.798642315	22 STO 06	34 GOOSE
	23 1.943857261	35*LBL 01

36 LEFT	82 RCL 10	128 +
37 RCL 15	83 +	129 E3/E+
38 STO 10	84 INT	130 REGMOVE
39*LBL 02	85 10^X	131*LBL 07
40 RNG	86 STO 12	132 LEFT
41 3	87 9	133 9
42 *	88 STO 14	134 STO 11
43 INT	89*LBL 06	135*LBL 08
44*LBL 03	90 LEFT	136 LEFT
45 LEFT	91 RCL IND 14	137 RNG
46 RNG	92 RCL 11	138 9
47 3	93 *	139 *
48 *	94 STO Y	140 INT
49 INT	95 E1	141 E
50 X=Y?	96 MOD	142 +
51 GTO 03	97 INT	143 10^X
52 RCL 10	98 STO 13	144 STO 12
53 ST+ Z	99 -	145 RCL IND 11
54 +	100 RCL 12	146 E
55 RCL IND Y	101 RCL 11	147 X=Y?
56 X<> IND Y	102 /	148 GTO 09
57 STO IND Z	103 STO 17	149 RDN
58 DSE 10	104 *	150 *
59 GTO 02	105 STO Y	151 STO Y
60 RCL 15	106 E1	152 E1
61 STO 10	107 MOD	153 MOD
62*LBL 04	108 INT	154 INT
63 LEFT	109 ST- Y	155 X=0?
64 RNG	110 X<> 13	156 GTO 08
65 3	111 +	157 -
66 *	112 RCL 17	158 RCL 12
67 INT	113 /	159 /
68*LBL 05	114 RCL 13	160 STO IND 11
69 LEFT	115 +	161 DSE 00
70 RNG	116 RCL 11	162 X=0?
71 3	117 /	163 GTO 10
72 *	118 STO IND 14	164*LBL 09
73 INT	119 DSE 14	165 DSE 11
74 X=Y?	120 GTO 06	166 GTO 08
75 GTO 05	121 DSE 10	167 GTO 07
76 RCL 10	122 GTO 04	168*LBL 10
77 +	123 DSE 16	169 RASP
78 INT	124 GTO 01	170 "GRID MADE"
79 10^X	125 9	171 AVIEW
80 STO 11	126 E3/E+	172 END
81 CLX	127 18	

Sudoku Solver (FOCAL version)

JM Baillard – <http://hp41programs.yolasite.com/sudoku.php>

This version is the "slow" brother of the MCODE implementation reviewed in the MCODE section. A main driver program "**SUDRPN**" orchestrates the execution of the data entry "SDKIN", puzzle resolution "SLSDK", and output of the results "SDKOUT". Note that some of these modules are shared by the MCODE implementation as well, saving space and providing more consistency to the games.

Thus the grid entry also employs the MCODE function ^SROW, although now each of the digits will be stored into an individual data register, from R01 to R81 0 which is remarkably inefficient since tall values are integers less than 10, but that's another story. The good news is that storing 81 values one at a time would be close to unbearable if it weren't for ^SROW of course...

You can use the same examples as shown in the MCODE section, but bear in mind that the execution times on a plain HP-41C are very long: about 1 hour for SDK1 and close to 2 ½ hours for SDK2. Obviously using TURBO mode on V41 (or the CL) is a must if you want to use the FOCAL slow counterparts.

Example1: Solve the following sudoku:

```
0 0 0 | 3 0 0 | 0 5 0
0 0 5 | 4 0 6 | 0 0 2
2 7 0 | 0 1 0 | 3 6 0
```

```
-----
7 0 4 | 2 3 0 | 0 0 0
5 1 0 | 0 0 0 | 0 3 7
0 0 0 | 0 4 7 | 9 0 1
```

```
-----
0 4 6 | 0 9 0 | 0 1 5
1 0 0 | 6 0 8 | 7 0 0
0 5 0 | 0 0 4 | 0 0 0
```

A more difficult example: With the grid:

```
0 9 0 | 0 4 2 | 0 1 0
0 0 5 | 0 0 0 | 0 0 0
3 0 0 | 0 0 0 | 9 0 4
```

```
-----
0 0 0 | 0 0 0 | 1 9 3
5 2 0 | 7 0 0 | 0 0 6
0 0 0 | 0 0 1 | 0 0 0
```

```
-----
9 0 0 | 0 5 0 | 0 6 0
0 0 0 | 2 0 4 | 0 0 7
0 0 0 | 0 1 6 | 8 0 0
```

The same back-tracking technique is used, in fact the FOCAL programs are an excellent "practice pad" before attempting the MCODE programming, to check the algorithms and verify the proof of concepts.

To provide some background to the user the addresses of the current registers are displayed during the execution (so at least you know what's going on...)

The programs use R01 to R81 to store the grid values, plus R00 and R89 for scratch.

Program listing:

01*LBL "SUDRPN"

02*LBL 00

03 -SUDOKU 1C

04 XROM "SDKIN"

05 -SOUND FX

06 XROM "SLSDK"

07 X=0?

08 GTO 01

09 CLAXON

10 "NO SOLUTION"

11 GTO 00

12*LBL 01

13 RASP
14 " ** DONE **"

00 LBL "SLSDK"

01 3
02 STO 86
03 6,009
04 STO 87
05 9
06 STO 88
07 7
08 STO 89
09 73,00009
10 STO 90
11 82
12 STO 91
13 STO 00
14 E3
15 STO 92
16 SIGN
17 STO 94
18*LBL 01
19 DSE 00
20 X=0?
21 RTN
22 RCL IND 00
23 X#0?
24 GTO 01
25 VIEW 00
26 RCL 88
27 STO 82
28*LBL 02
29 RCL 00
30 RCL 94
31 -
32 STO 85
33 RCL 88
34 ST/ 85
35 MOD
36 STO 83
37 RCL 90
38 +
39 X<> 85
40 INT
41 STO 84
42 RCL 94
43 +
44 RCL 88
45 *
46 STO 93
47*LBL 03
48 RCL IND 93

15 AVIEW
16 PSE

49 ABS
50 RCL 82
51 X=Y?
52 GTO 05
53 RCL IND 85
54 ABS
55 X=Y?
56 GTO 05
57 DSE 93
58 CLX
59 DSE 85
60 GTO 03
61 RCL 86
62 X<> 83
63 RCL 86
64 /
65 INT
66 RCL 84
67 RCL 86
68 /
69 INT
70 RCL 88
71 *
72 +
73 RCL 89
74 +
75 RCL 86
76 *
77 STO 85
78 RCL 86
79 -
80 RCL 92
81 /
82 ST+ 85
83*LBL 04
84 RCL IND 85
85 ABS
86 RCL 82
87 X=Y?
88 GTO 05
89 DSE 85
90 GTO 04
91 RCL 87
92 ST- 85
93 DSE 83
94 GTO 04
95 X<>Y
96 CHS
97 STO IND 00

17 XROM "SDKOUT"
18 END

98 GTO 01
99*LBL 05
100 DSE 82
101 GTO 02
102*LBL 06
103 RCL 00
104 RCL 94
105 +
106 STO 00
107 RCL 91
108 X=Y?
109 SF 41
110 RCL IND 00
111 X>0?
112 GTO 06
113 CHS
114 ST+ IND 00
115 STO 82
116 VIEW 00
117 DSE 82
118 GTO 02
119 GTO 06
120 END

00 LBL "SDKOUT"

01 82
02 E3/E+
03 9
04 E3/E+
05*LBL 01
06 "C"
07 AINT
08>": "
09 9
10 E3/E+
11*LBL 00
12 RCL IND Z
13 AINT
14 >": "
15 RDN
16 ISG Z
17 ISG X
18 GTO 00
19 AVIEW
20 RDN
21 ISG X
22 GTO 01
23 END

Kibur, or the “alphabetical disorder”

Robert Pulluard – L’Ordinateur de Poche N3 p43

If your handheld computer can display letters, the game offered here should fascinate you with its numerous reversals.

The Rubik cube is already a great classic among the games, as evidenced by the vast epidemic of acute “Rubikitis” that has been raging for some time in a large part of the world. Micro-pocket users may have been touched by the idea of adapting this game on their machine, but they have given up on it, because the means are clearly insufficient. What we can do, however, it is to adapt a game of the same kind, based on the Rubik cube principle.

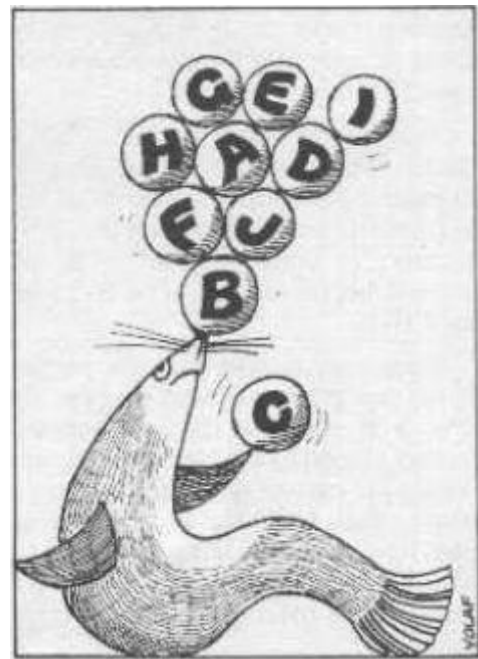
The program described here allows precisely to toy such a game, called “jeu da Kibur” (the origin of the name should no longer have any mystery anymore), but which of course is far from having the complexity of the original cube. It does offer a certain interest, especially if you play it together.

The microcomputer here, an HP-41C, has ten letters (A to J) in an arbitrary order: the goal of the game is to find the normal alphabetical order in a minimum of moves. Each move is defined by the inversion of two letters, one fixed, being the first on the left, in position “0”, the other being at the choice of the player among all ten letters (from 0 to 9: it is enough to introduce the rank of the desired letter). If there were only that, it would be far too easy and of little interest, which is why a “side effect” has been added to this inversion: it is the inversion of the two letters located on either side of the one chosen to be inverted with the letter in position zero.

Consider for example, the combination:

“**BFJHHACDIGE**”. By choosing “2” (position of the letter J) for insertion we have: “**JHBFACDIGE**” - i.e. “B”

(position 0) has been reversed with “J” (position 2, chosen), but in addition, the letters “F” and “H”, on either side of “J”, are also reversed.



Let us also note that this line of letters is looped, i.e. if we play the position “0” it has the effect of inverting the letters in position 1 and 9, while playing “9” results in the letter “0” changing to “9”, the letter “9” changing to “8”, and the letter “8” changing to “0”! Finally, in a similar way, playing “1” is like making a circular permutation of the three letters in positions “0”, “1”, and “2”. Thus, taking again the previous example where we left it if we play “1” we will obtain: “**BJHFACDIGE**” (the B took the place of the J who himself took the place of the H, who, in turn, came to occupy the position of the B).

These features complicate certainly the game, but they contribute also, by its own interest, by sometimes making the solution much less obvious than it seems.

The program as it is presented offers three different entry options. The first, by doing XEQ "KIBUR", asks the calculator to generate itself a starting combination, which it starts from the normal alphabet order, by applying ten successive random operations (but identical to resolution operations): that is, theoretically on average we should be able to solve all the problems posed by the calculator in about ten moves. It is "enough" to repeat the same operations, but in the opposite order. However, this is not entirely true for operations "1" and "9", which are not reversible, but cyclical of period 3: so if the calculator has performed one of these two other operations once between two other operations, the reverse operation is obtained by performing said operation two times in a row, whereas if one of these two operations has been performed twice in a row, performing this operation only once is enough to give its reverse. Unfortunately, or fortunately, depending on the point of view from which we are placed, there is no way to know the random sequence (and to remove from the program the - classical - generator function used).

In any case, it is not always necessarily the most efficient solution (in particular this random sequence can include twice in a row the same operation, which is therefore cancelled, except for operations "1" and "9" mentioned above).

The second possibility of access by means of XEQ "KIBIN", allows the player, or his colleague, to introduce a combination of his choice, for example an interesting combination, or a particularly thorny one and known as such.

The third access mode, with XEQ "KIBIS", allows to start a game again with the combination of the beginning of the previous game played: this combination is automatically put back into play by the calculator, and it is thus very easy to enjoy several successive games from the same combination, which is particularly interesting when playing with several players, the problem being then identical for all the competitors, each trying to reach the correct alphabetical order in a minimum number of moves.

The calculator displays after each ten-letter combination the number of strokes already played. This allows you to know where you stand at any time. If more than 99 moves have already been played, the calculator displays "LOST" and automatically makes an entry in "KIBIS", i. e. it resumes the start combination for a new attempt.

When you have made your choice of operation on the keyboard and have pressed a numeric key, the displayed situation (current combination and number of strokes) disappears, but it is kept in the machine's alpha register until you press the R/S key; it is therefore possible, if there is a doubt at the very last second, to check this situation again, but do not forget to switch back to digital mode once this check has been made, otherwise the machine will report an "ALPHA DATA" error.



Use of the program:

A: KIBUR variant, the calculator itself generates a combination,

1. do XEQ "KIBUR"
2. enter the position number of the letter you want to switch with the first one and press R/S. Warning: the first letter is in the zero position, the next letter in 1, then 2, etc.
3. return to 2

B: KIBIN variant: the player or one of his partners introduces the starting combination.

1. do XEQ "KIBIN". The display shows "COMBINATION", and you enter the first ten letters of the alphabet in the order you want. Pressing R/S wt I; you will find yourself at step 2 of KIBUR.

C: KIBIS variant: the calculator uses the last combination of start used (from which you can re-start the same game by trying to improve your score). Do XEQ "KIBIS", then proceed as in step 2 of KIBUR.

Exemple de partie n° 1

joué	résultat
12 54 76 80 11	ABEDGFGHJIC : 0
8	IBEDGFGHCAJ : 1
6	HBEDGFCIFAJ : 2
7	FBEDGCAHIJ : 3
5	CBEDAFGHIJ : 4
3	DBACEFGHIJ : 5
1	ADBCEFGHIJ : 6
2	BCADEF GHIJ : 7
1	ABCDEFGHIJ : 8

Exemple de partie n° 2

joué	résultat
22 90 54 76 80	CBIFEDGAHJ : 0
2	IFCBEDGAHJ : 1
6	GFCBEAIDHJ : 2
7	DFCBEAHGIJ : 3
6	HFCBEGDAIJ : 4
5	GFCBDHEAIJ : 5
6	EFCBDAGHIJ : 6
1	CEFB DAGHIJ : 7
2	FBCEDAGHIJ : 8
3	EBDFCAGHIJ : 9
4	CBDAEFGHIJ : 10
2	DACBEFGHIJ : 11
1	CDABEFGHIJ : 12
2	ABCDEFGHIJ : 13

Program listing:

<see next page>

01*LBL "KIBUR"

02 "ABCDEFGHIJ"

03 SF 01

04 XEQ 03

05 E1

06 STO 11

07*LBL A

08 RCL 00

09 PI

10 +

11 5

12 Y^X

13 FRC

14 STO 00

15 E1

16 *

17 INT

18 XEQ 02

19 DSE 11

20 GTO A

21 XEQ 01

22 ASTO 12

23 ASHF

24 ASTO 13

25*LBL B

26 FIX 0

27 0

28 STO 16

29*LBL C

30 XEQ 01

31 >"'"

32 ARCL 16

33 PROMPT

34 XEQ 02

35 99

36 RCL 16

37 E

38 +

39 STO 16

40 X<=Y?

41 GTO C

42 "PERDU"

43 PROMPT

44 GTO 10

45*LBL "KIBIN"

46 "COMBINAISON?"

47 AON

48 PROMPT

49 AOFF

50*LBL 03

51 ASTO 12

52 ASHF

53 ASTO 13

54*LBL "KIBIS"

55*LBL 10

56 1,01

57 STO 11

58 CLA

59 ARCL 12

60 ARCL 13

61 ASTO 14

62 ASHF

63 ASTO 15

64*LBL D

65 " "

66 ARCL 14

67 ARCL 15

68 ASTO IND 11

69 ASHF

70 ASTO 14

71 ASHF

72 ASTO 15

73 " "

74 ARCL IND 11

75 ASHF

76 ASTO IND 11

77 ISG 11

78 GTO D

79 FS?C 01

80 RTN

81 GTO B

82*LBL 01

83 CLA

84 1,01

85 STO 11

86*LBL a

87 ARCL IND 11

88 ISG 11

89 GTO a

90 RTN

91*LBL 02

92 E

93 +

94 10

95 X<>Y

96 X>Y?

97 X<>Y

98 STO 17

99 E

100 +

101 STO 18

102 RCL 01

103 RCL IND 17

104 STO 01

105 RDN

106 STO IND 17

107 RCL 17

108 1

109 -

110 X=0?

111 E1

112 STO 17

113 11

114 RCL 18

115 X=Y?

116 E

117 STO 18

118 RCL IND 17

119 RCL IND 18

120 STO IND 17

121 RDN

122 STO IND 18

123 END

Petals around the Rose

Edward E. Keefe – UPL #00479C

The Story behind “Petals around the rose”

(Apocryphal anecdote to acquaint you with the puzzle).

Once upon a time... during WWII, General Montgomery called his staff together and declared that he would grant a week-end pass to any and all officers who could come up with the correct answer to a simple guessing game.

He produced four dice and proceeded to roll them. Each time he queried: “how many petals are there around the rose?” He assumed that all his officers, being highly educated men, would “crack the puzzle” in short order. But, as the story goes, at the end of an entire week-end of tiring dice rolling, no one had cracked the puzzle (except for several hundred precocious children, a handful of mathematical geniuses and some adults who claim to have psychic powers). Those who have successfully solved the puzzle have since bonded together in a society known as the Order of the Rose. There is only one rule for this noble society and that is the rule of OMERTA – silence! – secrecy! No one may reveal the significance of the phrase “Petals around the rose”.

I, personally, have seen old men and women driven to much drunk after many hours of trying to get the significance of the five, frustrating die and the cabalistic and oft-repeated phrase.

I have also since realized that, for those who tend to be introspective and like to “watch” their brain at work, this little puzzle an interesting but somewhat erstwhile, illustration of the rudiments of the scientific method in process.

Game description:

The HP-41 requests your name and gender (it’s more chivalrous than sexist). It then proceeds to “roll: 5 dice across the display. You are to enter a guess for the number of “petals” around the rose”.

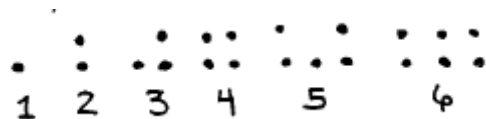
After 10 incorrect guesses you will receive 1 hint.

After three successive correct guesses you will be declared a knight or lady of the Order of the Rose.

Program re-cycles, rolls dice anew and calls for new guess. Note that you don’t need to press the R/S key after entering your guess value.

The program ends with a dubbing and announcement of number of guesses.

The unique feature of the program is the use of the ALPHA display to "show" the 5 dice as they are "rolled" by the HP-41C (a sort of "visual braille, so to speak):



If you can crack the code, the significance of the phrase "petals around the rose" should become less obscure. (Examining the "faces" of actual dice may also be of some help).

Example:

Input	Display	Comments
0, STO 10	0	
XEQ "PAR"	NAME?	
ED, R/S	SEX? M/F	key in M or F
M	ROLL.	
	: :: . :: .	pseudo-dice
	#=7 24131	blinking display

calls for guess. Numbers are given in case you misread the pseudo-dice
remember not to press RS !

24	SORRY, E0 THERE ARE 2 PETALS AROUND THE ROSE ROLL ... : :: . :: .	pseudo-dice
	#=7 24414	
2	SORRY, E0 THERE ARE 2 PETALS AROUND THE ROSE ROLL ... : :: . :: .	
	PETALS? 13623	
2	SORRY, E0`	etc...

Program listing:

01*LBL "PETALS"

02 13	48 >" :"	95 PSE
03 XROM "INIT"	49 FS?C 03	96 ,002
04 ,	50 >" ::"	97 STO 07
05 X<>F	51 FS?C 04	98 E
06 "NAME?"	52 >" ::"	99 ST+ 06
07 AON	53 FS?C 05	100 RCL 06
08 PROMPT	54 >" ::"	101 E1
09 AOFF	55 FS?C 06	102 X=Y?
10 ASTO 08	56 >" ::"	103 XEQ 03
11 ASHF	57 AVIEW	104 GTO 00
12 ASTO 09	58 2	105*LBL 03
13 "SEX? M/F"	59 /	106 "HINT:"
14 AVIEW	60 FRC	107 AVIEW
15*LBL 08	61 X=0?	108 PSE
16 GETKEY	62 GTO 02	109 "EVEN # <=20"
17 34	63 LASTX	110 AVIEW
18 X=Y?	64 ST+ X	111 PSE
19 GTO 09	65 E	112 RTN
20 13	66 -	113*LBL 04
21 -	67 ST+ 12	114 AVIEW
22 X#Y?	68*LBL 02	115 PSE
23 GTO 08	69 DSE 00	116 PSE
24 SF 00	70 GTO 01	117 FC? 22
25*LBL 09	71 PSE	118 GTO 04
26 ,002	72 CLA	119 RTN
27 STO 07	73 CF 22	120*LBL 06
28*LBL 00	74 "#=? "	121 ISG 07
29 E	75 ARCL 05	122 GTO 00
30 ST+ 11	76 ARCL 04	123 "I DUB THEE, "
31 "ROLL..."	77 ARCL 03	124 ARCL 08
32 AVIEW	78 ARCL 02	125 ARCL 09
33 CLA	79 ARCL 01	126 AVIEW
34 5	80 XEQ 04	127 PSE
35 STO 00	81 0	128 "A LADY"
36*LBL 01	82 X<> 12	129 FS?C 00
37 RNG	83 "SORRY, "	130 "A KNIGHT"
38 6	84 ARCL 08	131 AVIEW
39 *	85 ARCL 09	132 BEEP
40 E	86 X=Y?	133 PSE
41 +	87 " RIGHT."	134 CLA
42 INT	88 AVIEW	135 ARCL 11
43 STO IND 00	89 PSE	136 " GUESSES"
44 SF IND X	90 X=Y?	137 AVIEW
45 FS?C 01	91 GTO 06	138 FIX 3
46 >" ."	92 "#PETALS="	139 SF 29
47 FS?C 02	93 ARCL X	140 END
	94 AVIEW	

Snakes & Ladders

JM Baillard-<http://hp41programs.yolasite.com/snld.php>

Overview

You and the HP-41 are on square #0

The 2 players alternately roll 2 dice. (Flag F01 is set if it's your turn to play)
If the dice are D1 D2, the player on square S moves forward on square S+D1+D2

The snakes and the ladders may appear at random on every square #S provided that
 $1 < S < 100$

When a snake appears, you move back. If a ladder appears, you move forward.

Ladders are long near square#1, short near square#99
Snakes are short near square#1, long near square#99

The winner is the first one who reaches (or exceeds) square#100

Example: 5, STO 00, XEQ "SNLD"

" 00 / 00 "	and flag F01 is set: it's your turn to play, press any key "
" 04 / 00 - - 3 + 1 "	the dice are 3 & 1 so you move on square#04. Flag F01
	is cleared so the HP-41 roll the dice
" 04 / 07 - - 6 + 1 "	the dice are 6 & 1 the HP-41 move on square#07...
" SNAKE "	but there is a snake, the calculator moves back...
" 04 / 04 - - 6 + 1 "	on square #04...
" LADDER "	a ladder! HP-41 moves forward
" 04 / 19 - 6 + 1 "	on square#19 and flag F01 is set again: you press a key
" 13 / 19 - 3 + 6 "	the dice are 3 & 6 so you move on square#13

... and so forth ...

Notes:

The HP-41 displays " S1/S2 D1+D2" where S1 is your square, S2 is HP-41's square,
D1+D2 are the 2 dice.

Actually, the game is completely determined by the seed you've stored in register R00
If you want to influence the chance, add FS? 01 ST+ 00 after line 16:
the keys you'll press will modify the random numbers in register R00.

Do not store a number like PI in R00: the sequence of "random" numbers would be 0 0 0 0
.... (or change the random number generator lines 94 to 97)

There is a snake if $r > 0.8$; where r is the content of register R00
There is a ladder if $r < 0.8/6$
Change lines 59-62-63 if you prefer more (or less) snakes and/or ladders.

Program listing:

01*LBL "SNLD"

02 6	39 RCL 02	77 +
03 XROM "INIT"	40 X<Y?	78 INT
04 " 00/00"	41 >"0"	79 CHS
05 AVIEW	42 ARCL 02	80 " SNAKE"
06 RNG	43 >" "	81 AVIEW
07 STO 00	44 ARCL 03	82 GTO 16
08 CLX	45 >"+"	83*LBL 05
09 STO 01	46 ARCL 04	84 15
10 STO 02	47 AVIEW	85 RCL 05
11 CF 01	48 FS? 01	86 7
12 XEQ 06	49 RCL 01	87 /
13 4	50 STO 05	88 -
14 X>Y?	51 99	89 INT
15 SF 01	52 X<Y?	90 E
16*LBL 01	53 GTO 07	91 +
17 FS? 01	54 SIGN	92 " LADDER"
18 GETKEY	55 X=Y?	93 AVIEW
19 XEQ 06	56 GTO 03	94 GTO 16
20 STO 03	57 RCL 00	95*LBL 06
21 XEQ 06	58 R-D	96 RCL 00
22 STO 04	59 FRC	97 R-D
23 +	60 STO 00	98 FRC
24*LBL 16	61 .8	99 STO 00
25 FS? 01	62 X<Y?	100 6
26 ST+ 01	63 GTO 04	101 *
27 FC? 01	64 6	102 E
28 ST+ 02	65 /	103 +
29 COS	66 X>Y?	104 INT
30 COS	67 GTO 05	105 RTN
31 COS	68*LBL 03	106*LBL 07
32 RCL 01	69 FC?C 01	107 FS? 01
33 E1	70 SF 01	108 " YOU WIN"
34 " "	71 GTO 01	109 FC?C 01
35 X>Y?	72*LBL 04	110 " HP-41 WINS"
36 >"0"	73 RCL 05	111 FIX 4
37 ARCL 01	74 3	112 SF 29
38 >"/"	75 /	113 AVIEW
	76 E	114 END

Sphynx for the HP-41

JM Baillard - <http://hp41programs.yolasite.com/sphynx.php>

Overview

"Sphynx" is a number guessing game in which you have to find a mystery-number X. After you've keyed in a guess N, the HP-41 displays "N / d" where d = the sum of the digits of | X - N |.

For example, with N = 456789 and X = 259191, the HP-41 displays "456789 / 39" because | 259191 - 456789 | = 197598 and 1+9+7+5+9+8 = 39.

Exception: if d = 0 (ie. if X = N), the calculator displays: "X // G" where G is the number of your guesses.

In the following program, X is always a 6-digit number. It uses ATOX, ALENG and the CX-Function GETKEYX.

The program only stops when you've found the solution! After keying in your guess, press any key (for instance ENTER^) except the numeric keys, CHS and the decimal point.

Data Registers:

- R00 = r (random numbers)
- R01 = X R02 = 0, 1, 2, ..., G

Register R00 is to be initialized before executing "SPHYNX"

01 LBL "SPHYNX"	23 -	45 CLX
02 RCL 00	24 ABS	46 LBL 03
03 R-D	25 ARCL X	47 48
04 FRC	26 CLX	48 GETKEYX
05 STO 00	27 ALENG	49 X=0?
06 9 E5	28 CHS	50 GTO 01
07 *	29 48	51 X<>Y
08 INT	30 *	52 X<>L
09 E5	31 ENTER^	53 -
10 +	32 LBL 02	54 X<>Y
11 STO 01	33 STO Y	55 10
12 0	34 ATOX	56 *
13 STO 02	35 +	57 +
14 FIX 0	36 X#Y?	58 GTO 03
15 CF 29	37 GTO 02	59 LBL 04
16 LBL 01	38 " "	60 "~/"
17 X<>Z	39 ARCL Z	61 ARCL 02
18 ISG 02	40 "~/"	62 FIX 4
19 CLX	41 X=0?	63 SF 29
20 ENTER^	42 GTO 04	64 AVIEW
21 CLA	43 ARCL X	65 END
22 RCL 01	44 AVIEW	

NIM-41

Bob Laughton - Australian Tech Notes V6p11

Reviewed by Mark Power. DataFile V8N8p2 ; (December 1989)

Whilst perusing HP41 material taken from SWAP_07, I came across a program called "NIM41". After further investigation I found it to be an excellent implementation of that well-known game. Without the need for any extension modules whatsoever, the author has produced a fast and efficient game which allows just about every conceivable option.

For those of you who don't know how to play NIM, the idea is to remove sticks or counters from a number of piles. Each player takes it in turn to remove any number from the pile. The winner is simply the player who takes the last stick. This is not as easy as it sounds.

A strategy exists to play the game "perfectly". That is, given a "winning" position from the outset, you can guarantee that you win. However, to ensure then NIM41 is played I shall not give any clues.

The features available in the program are:

1. You can play against the calculator, or against another opponent.
2. You may select who goes first
3. You may have 3, 4, or 5 piles of sticks
4. The piles may be automatically set up (with a maximum which you specify), or you may set up the piles manually to replay a previous game (in an attempt to beat the machine?)
5. All input follows prompts ending with a question mark, and terminated by pressing R/S
6. Any errors in input are indicated as such by a message (either 'ERROR' or "Y/N" as necessary). The message is shown for a short period and the prompt redisplayed.

To best describe operation of the program, I will go through a sample game:

Output	Keys Pressed	Comment
	XEQ "NIM"	needs SIZE 015
RN. SEED?	1, R/S	Random number seed
12 PLAYERS?	1, R/S	play against HP41
A FIRST?	Y, R/S	Me first (could be N)
3, 4, 5 PILES?	3, R/S	Simple game
AUTO SETUP?	Y, R/S	I'll do the work
MAX. SIZE?	9, R/S	Could do it manually
1 : 5 : 3 : A		initial piles
	R/S	when ready to carry on
PILE?	2, R/S	from middle pile
TAKE?	5, R/S	take 5 sticks
PILE 2:TAKE 5		Summary of your move

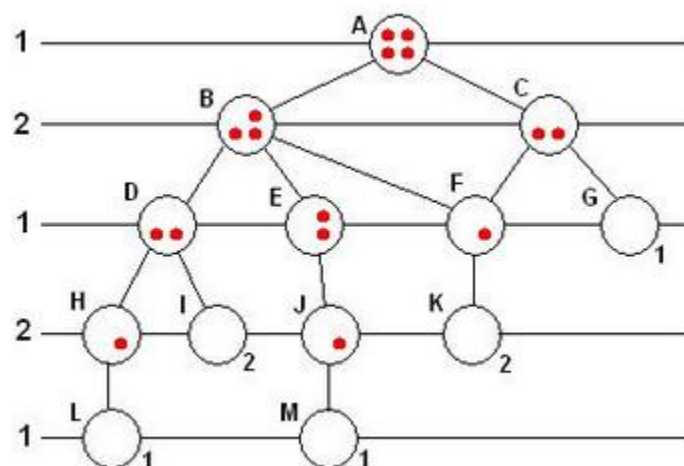
Output	Keys Pressed	Comment
0 : 0 : 1 HP THINKING...		HP41s state of piles Its turn now
PILE 3 TAKE 2		Its move
1 : 0 : 1 A	R/S	When ready
PILE?	1, R/S	From pile 1
TAKE?	1, R/S	Take a stick
PILE: TAKE 1		Summary of your move
0 : 1 : 1 HP THINKING...		HP41s state of piles Its going to win!
PILE 3 TAKE 1		Told you so!
HP WINS ANOTHER?	N, R/S	

If you play a two-player game the program follows much the same path but rather than THINKING itself it prompts the other player. Similarly if you answer "N" to "A FIRST?", your opponent goes first.

For automatic setup you are prompted for the number of sticks in each pile. Typing "Y" at the "ANOTHER?" prompt takes you back to the initial prompts for another game.

Have fun playing the game (and trying to see how it works if you want). FOCAL programs don't come more clean than this one! Finally, many thanks to whoever wrote it - I think it is brilliant!

Ed's note: The program was written by Bob Laughton, published in the Australian "Technical Notes" TN V6p11 issue



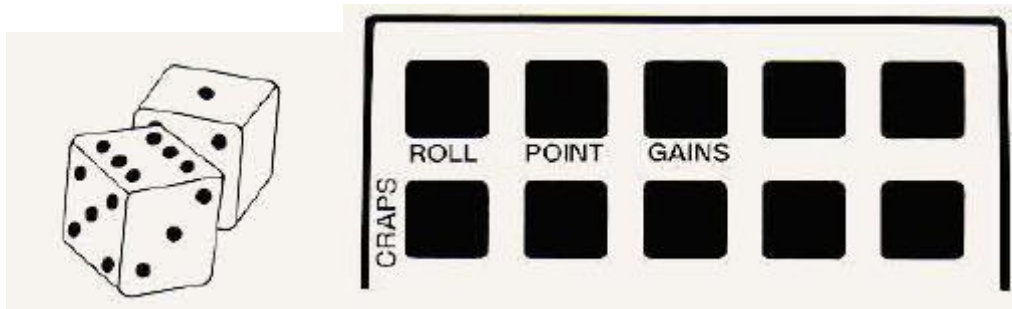
Program listing:

01*LBL "NIM"	49 /	97*LBL 18	146 FS? 10
02 SIZE?	50 STO 00	98 FC?C 05	147 GTO 08
03 15	51*LBL 04	99 SF 05	148 RCL 10
04 X>Y?	52 "AUTO SET	100 " B"	149 X<>Y
05 PSIZE	UP?"	101 FS? 01	150*LBL 19
06 RNG	53 XEQ 22	102 "HP"	151 BEEP
07 STO 13	54 FS? 10	103 FC? 05	152 "PILE:"
08*LBL 50	55 GTO 04	104 " A"	153 ARCL Y
09 CF 00	56 FS? 09	105 FS? 00	154 "` TAKE:"
10 CF 01	57 GTO 06	106 "HP"	155 ARCL X
11 CF 02	58 XEQ 23	107 ASTO 14	156 AVIEW
12 CF 03	59*LBL 05	108*LBL 08	157 PSE
13 CF 04	60 "PILE "	109 CLA	158 ST- IND Y
14 CF 29	61 ARCL 00	110 XEQ 23	159 XEQ 23
15 FIX 0	62 "`?"	111*LBL 09	160 0
16*LBL 01	63 CLST	112 9	161*LBL 10
17 "1, 2	64 PROMPT	113 RCL IND 00	162 RCL IND 00
PLAYERS?"	65 1	114 X<=Y?	163 +
18 CLST	66 ENTER^	115 "` "	164 ISG 00
19 PROMPT	67 99	116 ARCL X	165 GTO 10
20 0	68 XEQ 21	117 "`:"	166 X=0?
21 ENTER^	69 FS? 10	118 ISG 00	167 GTO 25
22 2	70 GTO 05	119 GTO 09	168 GTO 18
23 XEQ 21	71 STO IND 00	120 ARCL 14	169*LBL 00
24 FS? 10	72 ISG 00	121 AVIEW	170 PSE
25 GTO 01	73 GTO 05	122 "HP"	171 PSE
26 SF IND X	74 GTO 18	123 ASTO X	172 "THINKING...."
27 FS? 00	75*LBL 06	124 RCL 14	173 AVIEW
28 GTO 03	76 "MAX. SIZE?"	125 X=Y?	174 XEQ 23
29 CF 05	77 CLST	126 GTO 00	175 6,9
30*LBL 02	78 PROMPT	127 STOP	176 STO 11
31 "A FIRST?"	79 E	128 "PILE?"	177*LBL 11
32 XEQ 22	80 ENTER^	129 CLST	178 RCL IND 00
33 FS? 10	81 99	130 PROMPT	179 STO IND 11
34 GTO 02	82 XEQ 21	131 E	180 ISG 11
35 CF 05	83 FS? 10	132 RCL 00	181 ISG 00
36 FS? 09	84 GTO 06	133 FRC	182 GTO 11
37 SF 05	85 STO 10	134 E3	183 CF 10
38*LBL 03	86 XEQ 23	135 *	184 0
39 "3, 4, 5 PILES?"	87*LBL 07	136 XEQ 21	185 STO 12
40 CLST	88 XEQ 24	137 FS? 10	186 64
41 PROMPT	89 RCL 10	138 GTO 08	187*LBL 12
42 3	90 *	139 STO 10	188 XEQ 20
43 ENTER^	91 E	140 "TAKE?"	189 LASTX
44 5	92 +	141 CLST	190 2
45 XEQ 21	93 INT	142 PROMPT	191 /
46 FS? 10	94 STO IND 00	143 E	192 INT
47 GTO 03	95 ISG 00	144 RCL IND 10	193 X#0?
48 E3	96 GTO 07	145 XEQ 21	194 GTO 12

195 RCL 12	228 5,005	261 X<> Z	294 "Y/N"
196 X=0?	229 ST+ 00	262 ABS	295 AVIEW
197 GTO 13	230 CF 08	263 INT	296 PSE
198 RCL 11	231 CF 09	264 X<Y?	297 RTN
199 5	232*LBL 14	265 SF 10	298*LBL 23
200 -	233 RCL IND 00	266 RCL Z	299 RCL 00
201 X<>Y	234 LASTX	267 X<>Y	300 FRC
202 GTO 19	235 X>Y?	268 X>Y?	301 E
203*LBL 13	236 GTO 00	269 SF 10	302 +
204 XEQ 24	237 ST- IND 00	270 FC? 10	303 STO 00
205 RCL 00	238 FC?C 08	271 RTN	304 RTN
206 FRC	239 SF 08	272 "ERROR"	305*LBL 24
207 E3	240 RCL 00	273 AVIEW	306 RCL 13
208 *	241 FC? 10	274 PSE	307 997
209 *	242 STO 11	275 RTN	308 *
210 1	243 RCL 11	276*LBL 22	309 FRC
211 +	244 X=Y?	277 CF 09	310 STO 13
212 INT	245 SF 09	278 CF 10	311 RTN
213 RCL IND X	246*LBL 00	279 AON	312*LBL 25
214 X=0?	247 ISG 00	280 PROMPT	313 CLA
215 GTO 13	248 GTO 14	281 AOFF	314 ARCL 14
216 XEQ 24	249 5 E-3	282 ASTO X	315 "" WINS"
217 *	250 ST- 00	283 "Y"	316 AVIEW
218 ,8	251 FC? 08	284 ASTO Y	317 PSE
219 *	252 RTN	285 X=Y?	318 "ANOTHER?"
220 1	253 LASTX	286 SF 09	319 XEQ 22
221 +	254 SF 10	287 X=Y?	320 FS? 10
222 INT	255 FC? 09	288 RTN	321 GTO 25
223 GTO 19	256 CHS	289 "N"	322 FS? 09
224*LBL 20	257 ST+ 12	290 ASTO Y	323 GTO 50
225 XEQ 23	258 RTN	291 X=Y?	324 CLST
226 RDN	259*LBL 21	292 RTN	325 END
227 SIGN	260 CF 10	293 SF 10	

Craps

HP Co. – Games Pac



The calculator plays the part of a casino operator in the game of craps. The rules for craps are as follows :

- Place a bet.
- Roll 2 die: if they total 7 or 11 on the first roll, you win. If they total 2, 3 or 12 on the first roll, you lose. Any other total on the first roll becomes your "point."
- Continue rolling the die until you either roll your "point," (you win) or you roll a 7, (you lose).

Program listing:

01*LBL A	24 11	47 FC?C 25
02 " ROLLING "	25 X=Y?	48 PROMPT
03 " " "	26 GTO 04	49 CF 05
04 CF 21	27 RDN	50 CLRG
05 AVIEW	28 2	51 SF 27
06 XEQ 00	29 X=Y?	52 XROM "INI"
07 STO 01	30 GTO 02	53 STO 00
08 " " "	31 RDN	54 GTO 03
09 ARCL X	32 3	55*LBL 04
10 XEQ 00	33 X=Y?	56 " YOU WIN"
11 " " "	34 GTO 02	57 BEEP
12 AVIEW	35 RDN	58 XROM "P"
13 ARCL X	36 12	59 CF 05
14 XROM "P"	37 X=Y?	60 RCL 05
15 RCL 01	38 GTO 02	61 ST+ 04
16 +	39 "POINT="	62 GTO 03
17 FS? 05	40 ARCL Y	63*LBL 02
18 GTO 01	41 XROM "P"	64 "YOU LOSE"
19 STO 02	42 SF 05	65 TONE 4
20 7	43 GTO 06	66 XROM "P"
21 X=Y?	<u>44*LBL "CRAPS"</u>	67 RCL 05
22 GTO 04	45 6	68 ST- 04
23 RDN	46 XROM "SIZE?"	69 CF 05

70*LBL 03	84 ARCL 02	98 7
71 SF 29	85 XROM "P"	99 X=Y?
72 FIX 2	86 PSE	100 GTO 02
73 " PLACE BET"	87 GTO 06	101 GTO 06
74 PROMPT	88*LBL C	102*LBL 00
75 CF 29	89 "\$"	103 XROM "RNDM"
76 FIX 0	90 ARCL 04	104 6
77 STO 05	91 XROM "P"	105 *
78*LBL 06	92 GTO 03	106 1
79 "ROLL"	93*LBL 01	107 +
80 AVIEW	94 RCL 02	108 INT
81 STOP	95 X=Y?	109 END
82*LBL B	96 GTO 04	
83 "POINT="	97 RDN	

Craps, v2

[Samuel & Spencer Hartman – PPCCJ V14N3 p27 \(March 1987\)](#)

Updated version of "Craps" A Game, with some new text, a new listing and barcode added. Original article published last Issue, V14N2 pages 16 and 17.

Craps, or dice as it sometimes called is a game with simple rules but it has caused much woe in the land. The game is played by rolling two dice each of which has six sides. Dots numbering from 1 to 6 are printed on the sides. After a roll, the sums of the dots facing upwards are added; the total is called the "POINT". If on the first roll, the sum is 7 or 11, the player wins. If the sum is 2, (SNAKE-EYES) 3, (CRAPS) or 12, (BOX-CARS) the player loses.

If none of the above numbers appears, the player tries to make his POINT by making successive rolls. If a 7 is rolled before making the POINT, the player loses. The point must be rolled before rolling a 7 to win. Any number of rolls can be made until either a 7 or the point appears.

Bets are made on the expected outcome. The pay-off equals the amount or the bet as does the loss.

To run the program, XEQ "CRAPS" and answer prompt for "AUTO-PLAY Y/N?". If auto-play, press Y. If manual play is desired, press any other key.

Answer prompts for "TIME", "BEGINNING BANK AMOUNT" and "BET \$\$". Press R/S after each entry.

If in auto-play, the program will run until the bank goes broke, or play is ended by R/S.

If manual play, the option of changing the bet amount is allowed. Key in the amount at the prompt for "BET \$\$".

Entering "TIME" insures a new set of random numbers for each start.

The bank automatically keeps tab of the winnings or losses. A message is printed when the amount reaches zero. "Bank - 0.00 "YOU'RE BROKE I " Then, "BGN BANK AMT" . Key in new bank amount, press R/S and continue the game.

Watch the display as the program searches for numbers other than 7, 11, 3, 2, or 12. These numbers are displayed and then printed representing successive rolls of the dice.

By Samuel I. J. Hartman #12558
5954 Colfax Avenue Apt 3
N. Hollywood, CA 91681

Spencer J. Hartman #9541
2432 Stow Street
Simi, CA 93063

Program listing:

01*LBL "CRAPS"

02 CF 27	50 FMT	99 GTO 03
03 CF 19	51 ACX	100 X<>Y
04 CF 20	52 ARCL X	101 12
05 PWRDN	53 XEQ 13	102 X=Y?
06 PWRUP	54 FC? 20	103 GTO 05
07 ADV	55 " "	104 X<>Y
08 "CRAPS"	56 ADV	105 3
09 SF 12	57 X#0?	106 X=Y?
10 ACA	58 GTO 03	107 GTO 06
11 FMT	59 "YOU'RE BROKE!"	108 X<>Y
12 ADV	60 PRA	109 2
13 CF 12	61 ADV	110 X=Y?
14 "AUTOPLAY"	62 FS?C 20	111 GTO 07
15 ACA	63 SF 19	112 CF 29
16 ",Y/N?"	64 GTO 01	113 " "
17*LBL 00	65*LBL 03	114*LBL 04
18 XEQ 13	66 FC? 20	115 XEQ 10
19 GETKEY	67 GTO 03	116 CF 21
20 X=0?	68 RCL 02	117 TONE 8
21 GTO 00	69 X>Y?	118 VIEW X
22 71	70 X<>Y	119 SF 21
23 X=Y?	71*LBL 03	120 E1
24 SF 19	72 "BET \$\$"	121 X>Y?
25 "NO"	73 XEQ 11	122 ACA
26 X=Y?	74 FS? 19	123 X<>Y
27 "YES"	75 STO 02	124 ACX
28 FMT	76 STO 03	125 7
29 ACA	77 RCL 01	126 X=Y?
30 ADV	78 X<Y?	127 GTO 08
31 FIX 0	79 GTO 02	128 X<>Y
32 "TIME"	80 FS?C 19	129 RCL 04
33 XEQ 11	81 SF 20	130 X#Y?
34 PI	82 FIX 0	131 GTO 04
35 *	83 XEQ 10	132 PRBUF
36 XROM "MT"	84 "YOUR "	133*LBL 03
37 E1	85 ACA	134 RCL 03
38 /	86 "POINT IS "	135 CHS
39 STO 00	87 ACA	136 "YOU WIN!"
40*LBL 01	88 ACX	137 BEEP
41 FIX 2	89 ARCL X	138 BEEP
42 "BGN BNK AMT"	90 XEQ 13	139 GTO 09
43 XEQ 11	91 PRBUF	140*LBL 05
44 STO 01	92 STO 04	141 "BOX-CARS"
45 GTO 03	93 11	142 GTO 03
46*LBL 02	94 X=Y?	143*LBL 06
47 FIX 2	95 GTO 03	144 "CRAPS"
48 "BNK= "	96 X<>Y	145 GTO 03
49 ACA	97 7	146*LBL 07
	98 X=Y?	147 "SNAKE-EYES"

148*LBL 03	168 FMT	188 X=0?
149 "!"	169 ADV	189 RDN
150 CF 21	170 ADV	190 FS? 20
151 AVIEW	171 ST- 01	191 GTO 03
152 SF 21	172 RCL 01	192 BEEP
153 ACA	173 GTO 02	193 PROMPT
154 FMT	174*LBL 10	194 X=0?
155*LBL 08	175 CLX	195 GTO 12
156 PRBUF	176 XROM "RN"	196 FC?C 22
157 RCL 03	177 11	197 GTO 12
158 "YOU LOSE"	178 *	198*LBL 03
159 TONE 8	179 2	199 ACX
160 TONE 8	180 +	200 ADV
161 TONE 6	181 INT	201 RTN
162*LBL 09	182 RTN	202*LBL 13
163 SF 29	183*LBL 11	203 CF 21
164 CF 21	184 ACA	204 TONE 8
165 AVIEW	185 FMT	205 AVIEW
166 SF 21	186 "?"	206 SF 21
167 ACA	187*LBL 12	207 END

Gambler

Neil G. Jarman - DataFile V3N2 p20 ; (Mar/Apr 1983)

A game for any number of people, SIZE being set to $8+2*n$ (n = no. of players) if it isn't large enough already. Each player is given an initial Bank of \$100, and at each go the player types in his/her bet.

The '41 picks two numbers within a specified range, showing you (the player) only one of these. You must decide whether the second number is Higher or Lower. If you are correct, you can either stop or gamble. If you stop your winnings (twice your bet) are added to your Bank, and the '41 moves on to the next player. If you gamble, the range of the two numbers is changed (an extra \$10 for each gamble), and you repeat the guessing process. If you lose a guess, the '41 moves on to the next player. A Bank statement is issued before and after each player plays. If everyone is out, the '41 will cycle until stopped manually.

NOTES

- i) remember names max 6 chrs...
- ii) don't bet more than your Bank.
- iii) as the only likely place to stop play is at the "FRED, BET ?" prompt, you can get back here by XEQ 01.

Happy gambling!

Neil G. Jarman (43M – 11052)

Program listing:

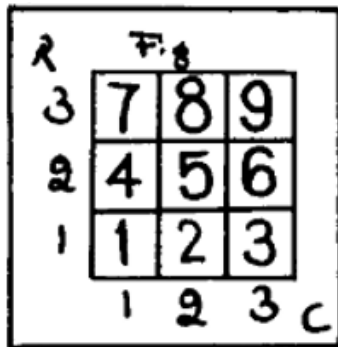
01*LBL "GAMBLR"	22 /	43 GTO 00
02 FIX 0	23 8	44 AOFF
03 CF 05	24 +	45 RCL IND 01
04 CF 06	25 STO 01	46 STO 03
05 SF 28	26 STO 02	47 1
06 CF 29	27 AON	48 ST+ 01
07 "#PLAYERS=?"	28 1	49*LBL 01
08 E	29*LBL 00	50 RCL IND 01
09 PROMPT	30 "PLAYER "	51 CLA
10 X=Y?	31 FC? 06	52 ARCL 03
11 SF 06	32 ARCL X	53 ", BET=?"
12 ST+ X	33 "` ?"	54 PROMPT
13 8	34 STOP	55 X<=Y?
14 +	35 ASTO IND Y	56 GTO 02
15 SIZE?	36 ISG Y	57 "NO CREDIT"
16 X<>Y	37 E1	58 AVIEW
17 X>Y?	38 STO IND Z	59 PSE
18 PSIZE	39 RDN	60 GTO 01
19 E	40 E	61*LBL 02
20 -	41 +	62 STO 04
21 E3	42 ISG Y	63 ST- IND 01

64 0	104 "NO"	144*LBL 11
65 STO 05	105 XEQ 13	145 TIME
66*LBL 03	106 0	146 STO 00
67 XEQ 12	107 STO 04	147 *
68*LBL 04	108*LBL 06	148 INT
69 RCL 05	109 RCL 04	149 1
70 XEQ 11	110 ST+ IND 01	150 +
71 STO 06	111 XEQ 08	151 RTN
72 RCL 05	112*LBL 07	152*LBL 12
73 XEQ 11	113 PSE	153 10
74 X=Y?	114 ISG 01	154 ST+ 05
75 GTO 04	115 X<0?	155 "RANGE: 1 , "
76 STO 07	116 XEQ 09	156 ARCL 05
77 CLA	117 RCL IND 01	157 AVIEW
78 ARCL 06	118 STO 03	158 RTN
79 " ? (H/L)"	119 ISG 01	159*LBL 13
80 AVIEW	120 RCL IND 01	160 " , I CHOSE "
81 23	121 X<=0?	161 ARCL 07
82 33	122 GTO 10	162 AVIEW
83 XEQ 14	123 FC? 06	163 PSE
84 RCL 06	124 XEQ 08	164 RTN
85 RCL 07	125 GTO 01	165*LBL 14
86 -	126*LBL 08	166 GETKEY
87 FS?C 05	127 CLA	167 X#0?
88 CHS	128 ARCL 03	168 GTO 15
89 X<0?	129 "'S BANK IS \$"	169 RDN
90 GTO 05	130 ARCL IND 01	170 GTO 14
91 2	131 AVIEW	171*LBL 15
92 ST* 04	132 RTN	172 X=Y?
93 "YES"	133*LBL 09	173 RTN
94 XEQ 13	134 RCL 02	174 X<>Y
95 "GAMBLE (Y/N)"	135 STO 01	175 RDN
96 AVIEW	136 RTN	176 X=Y?
97 71	137*LBL 10	177 SF 05
98 41	138 CLA	178 X=Y?
99 XEQ 14	139 ARCL 03	179 RTN
100 FS?C 05	140 "'S OUT"	180 CLX
101 GTO 03	141 0	181 RCL T
102 GTO 06	142 AVIEW	182 GTO 14
103*LBL 05	143 GTO 07	183 END

Catch the Goufers

Thomas Fange – PPCCJ V8N6 p44 (Aug/Dec 1981)

Goufers are small, disagreeable furious, fluffy, ugly animals. They live in a 9 room cave (see diagram below). Your mission is to kill all goufers. You have 10 shots, The goufers position will be shown to you in the form of a quick R-C-N prompt, with R being the ROW, C being the column, and N being the number of goufers remaining.



When this is shown, you have about 1 second during a pause to enter the right key to shoot the goufer. When you shoot (or if you are too slow and don't get to shoot), a message will scroll around the screen indicating whether you got the goufer or missed him. At the end, you will be told how many you killed. Short but fun, Let the hunt begin!

Operating Limits and Warnings:

Don't touch the animals: they bite.

The program material contain herein is supplied without representation or warranty of any kind. The author therefore assumes no responsibility, and shall have no liability if the user of the program is bitten by a goufer of any kind, or gets hurt in one or another way using this program material or any part thereof.

THOMAS FANGE [17397]
PPC-GOTHENBURG
STORANGSGATAN 24
S-413 19 GÖTEBORG
SWEDEN

Program listing:

<see next page>

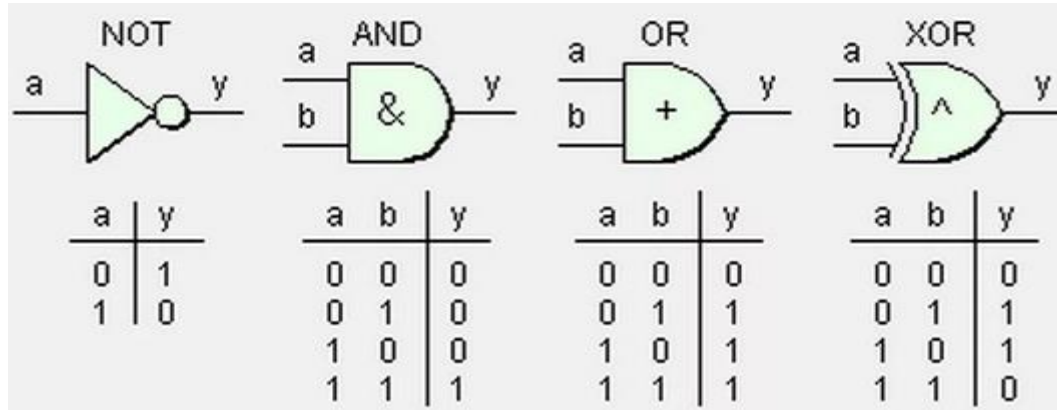
01*LBL "GOUF"

02 9	31 9	61 FRC
03 XROM "INIT"	32 +	62 STO 00
04 CF 21	33 INT	63 RTN
05 SF 27	34 STO 03	64*LBL 01
06 RNG	35*LBL 10	65 "1-1"
07 STO 00	36 DSE 03	66 RTN
08*LBL A	37 GTO 10	67*LBL 02
09 E1	38 AVIEW	68 "1-2"
10 STO 01	39 PSE	69 RTN
11 CLX	40 RCL 02	70*LBL 03
12 STO 04	41 X#Y?	71 "1-3"
13 "-----"	42 GTO 11	72 RTN
14*LBL 13	43 "GOT HIM"	73*LBL 04
15 SF 25	44 E	74 "2-1"
16 AVIEW	45 ST+ 04	75 RTN
17 SF 99	46 GTO 12	76*LBL 05
18 XEQ 00	47*LBL 11	77 "2-2"
19 9	48 "MISSED"	78 RTN
20 *	49*LBL 12	79*LBL 06
21 E	50 DSE 01	80 "2-3"
22 +	51 GTO 13	81 RTN
23 INT	52 AVIEW	82*LBL 07
24 STO 02	53 PSE	83 "3-1"
25 XEQ IND 02	54 "GOT "	84 RTN
26 >"-L:"	55 ARCL 04	85*LBL 08
27 ARCL 01	56 PROMPT	86 "3-2"
28 XEQ 00	57*LBL 00	87 RTN
29 37	58 RCL 00	88*LBL 09
30 *	59 997	89 "3-3"
	60 *	90 END

XOR Game

Dejan Ristanovic - DataFileV2N1 p18 ; (Jan/Feb 1983)

First of all, you must be familiar with the logic functions of: .OR. & .XOR.



THE GAME

The goal of the game is to score five points before the HP-41 does the same. To score a point, you must reach a 1111 combination. You will start with a 0000 combination. The machine will offer you a number - which you may take or reject. If you take the number, it will be summed (or rather XOR'ed) with the number that you already have, bit by bit. For example; let's say that you have 1011. The machine offers you 0110 - which you do decide to accept. Your new number is now 1101. If you reject the offered number, then the machine will take this number. It will be summed with that that the machine already has, BUT, by using the .OR. function instead of "XOR. So, as you can see the machine has an advantage (it cannot lose bits once scored) but you have the advantage of having a choice.

It is very simple to use the program. Enter any seed between 0 and 1 and then XEQ "XOR". The 41 will display the number on offer. If you want to accept, just press any number key during the next 2 seconds (it's quite short so you have to think very fast). The 41 will then display NNNN PLAYER - where NNNN is the number that you as the PLAYER have currently. If you decide not to take the number on offer, then DO NOTHING. The machine will then display NNNN HP41C - this being the number that the 41 now has.

Once you manage to score a point against the machine, the display will show POINT PLAYER - and should the machine score a point then POINT HP41C is displayed. After five points are scored, the name of the winner is displayed in the same way.

To restart the game" just enter a new seed and XEQ "XOR". You will find the game quite hard to win. To do so, you must think and react as fast as possible - PLUS, of course have some good luck on your side. Have fun,

Dejan Ristanovic, Belgrade 15.7.82

Program listing:

01*LBL "XOR"

02 CLRG	41 FS? 22	81*LBL 11
03 STO 09	42 "PLAYER"	82 RCL 05
04 5	43 ASTO 08	83 4
05 STO 10	44 14	84 -
06 STO 15	45 FC? 22	85 DSE IND X
07*LBL 01	46 19	86 GTO 13
08 4	47 STO 00	87 "WINNER "
09 STO 00	48 STO 05	88 ARCL 08
10 RCL 09	49 4	89 AVIEW
11 PI	50 STO 06	90 STOP
12 +	51*LBL 04	91*LBL 13
13 ENTER^	52 2	92 "POINT"
14 X^2	53 RCL IND 06	93 E3
15 *	54 RCL IND 00	94 /
16 FRC	55 +	95 RCL 05
17 STO 09	56 X#Y?	96 +
18 FIX 0	57 GTO 05	97 STO 00
19 16	58 E	98 .
20 *	59 FS? 22	99*LBL 14
21 INT	60 .	100 STO IND 00
22*LBL 02	61*LBL 05	101 DSE 00
23 2	62 STO IND 00	102 GTO 14
24 /	63 E	103 GTO 12
25 ENTER^	64 ST- 00	104*LBL 99
26 FRC	65 DSE 06	105 CLA
27 2	66 GTO 04	106 ENTER^
28 *	67 RCL 05	107 ENTER^
29 STO IND 00	68 XEQ 99	108 4
30 RDN	69 ASTO Y	109 -
31 INT	70 "1111"	110 E3
32 DSE 00	71 ASTO X	111 /
33 GTO 02	72 X=Y?	112 +
34 4	73 GTO 11	113 STO 07
35 XEQ 99	74 CLA	114*LBL 98
36 CF 22	75 ARCL Y	115 ARCL IND 07
37 AVIEW	76*LBL 12	116 DSE 07
38 PSE	77 " "	117 GTO 98
39 PSE	78 ARCL 08	118 END
40 "HP41C"	79 AVIEW	
	80 GTO 01	

Lunar Lander

Mark Gessner - PPCCJ V12N3 p26 ; (March 1985)

You are the pilot of the Lunar Excursion Module on a descent to the surface of the moon. You have a limited store of fuel, and you must decide how much fuel to burn based on information about your vertical velocity, and your altitude above the surface. The objective is to land the craft without inflicting serious damage to either the module or the crew. Mission control in Houston has guided your craft to a position directly over the place where you need to land, and they are leaving the rest up to you.

There are two ways to play this game. Option one is the beginner's mode, and will give you more time to see ship status, such as velocity, height, and fuel remaining. To start this beginner mode, XEQ "LOONER" and enter a number 1 at the "^1 BEGINNER" prompt. You will see two number display simultaneously. The left-hand number is your vertical velocity. If it is negative, you're travelling downward. The right-hand number is your altitude. Try to bring both values to zero at the same time, by supplying the proper amount of fuel when asked to do so by the on-board computer.

When you become good at the beginner game, XEQ "LOONER" and press R/S at the "^BEGINNER" prompt or simply XEQ "LL". You will have a three-number display, which is identical in meaning to the two-number display described above. Except that the fuel remaining is now displayed along with your velocity and height. This mode permits faster playing of the game.

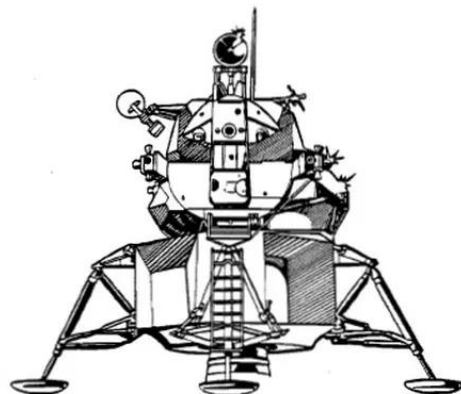
Global key [A] will restart the landing if you should run into trouble, so long as it is pressed before the "CRASHED" or "EXCELLENT..." processing is completed; this is an option the APOLLO astronauts wished for, no doubt.

The original form of this game was written by Shawn D. Johnson, a non-PPC member who also wrote the original TARG game. The beginner/advanced display routines were written by Pail G. Mitchell (7497), and I have made countless trivial changes to it over the course of the last year.

Mark D. Gessner (11922)
401 Stasney 107
College Station, TX 77840
USA

Program listing:

<See next page.>



01*LBL "LOONER"

02 CF 22	53 STO 01	105 .	157 TONE 0
03 "^1 BEGINNER"	54 CLA	106 "FUEL? "	158 "CRASHED"
04 PROMPT	55 ARCL 01	107 ARCL 02	159 AVIEW
05 FS?C 22	56 " " "	108*LBL 12	160 PSE
06 SF 02	57 FS? 02	109 CLST	161 "VEL= "
07*LBL A	58 GTO 13	110 PROMPT	162 ARCL 07
08*LBL 00	59 RCL 00	111 ABS	163 AVIEW
09 CF 08	60 99	112 RND	164 PSE
10 FIX 0	61 X<Y?	113 RCL 02	165 5 E1
11 2	62 FIX 0	114 X<>Y	166 RCL 07
12 STO 04	63*LBL 13	115 X>Y?	167 ABS
13 -5	64 ARCL 00	116 GTO 05	168 +
14 STO 05	65 FS? 02	117 ST- 02	169 CHS
15 RCL 10	66 GTO 11	118 RCL 04	170 ST+ 10
16 SIN	67 " " "	119 *	171 CLA
17 5	68 FIX 1	120 RCL 05	172 FIX 0
18 /	69 12	121 +	173 " " "
19 ST+ 05	70 ALENG	122 STO 03	174 ARCL X
20 2	71 X<Y?	123 .5	175 " " PTS"
21 /	72 SF 01	124 *	176 AVIEW
22 ST+ 04	73 FS? 01	125 RCL 07	177 PSE
23 E	74 ARCL 02	126 +	178 GTO 10
24 ST+ 11	75 FS?C 01	127 ST+ 06	179*LBL 03
25 RCL 10	76 GTO 12	128 RCL 06	180 RCL 03
26 SIN	77 FIX 0	129 INT	181 ST+ 07
27 E^X	78 11	130 X>0?	182 GTO 01
28 2 E2	79 X=Y?	131 GTO 03	183*LBL 04
29 *	80 SF 01	132 X<0?	184 RCL 03
30 5 E1	81 FS? 01	133 GTO 07	185 .5
31 +	82 ARCL 02	134 RCL 03	186 *
32 STO 06	83 FS?C 01	135 ST+ 07	187 RCL 07
33 FIX 4	84 GTO 12	136 RCL 07	188 +
34 .85	85 RCL 02	137 X>0?	189 ST+ 06
35 Y^X	86 E2	138 GTO 01	190 RTN
36 CHS	87 X<=Y?	139*LBL 02	191*LBL 05
37 STO 07	88 SF 01	140 RCL 07	192 TONE 1
38 CHS	89 FS? 01	141 -5	193 TONE 1
39 SQRT	90 -	142 X<=Y?	194 TONE 1
40 12.5	91 FS? 01	143 GTO 09	195 TONE 0
41 *	92 " " ":"	144 RCL 07	196 TONE 0
42 3 E1	93 FC?C 01	145 - E1	197 "OUT OF FUEL"
43 -	94 RDN	146 X<=Y?	198 AVIEW
44 INT	95 9	147 GTO 08	199 PSE
45 STO 02	96 X<>Y	148 TONE 9	200 RCL 02
46*LBL 01	97 X<=Y?	149 TONE 8	201 RCL 04
47 FIX 1	98 "0"	150 TONE 7	202 *
48 RCL 06	99 ARCL X	151 TONE 6	203 RCL 05
49 RND	100 GTO 12	152 TONE 5	204 +
50 STO 00	101*LBL 11	153 TONE 4	205 STO 03
51 RCL 07	102 AVIEW	154 TONE 3	206 XEQ 04
52 RND	103 PSE	155 TONE 2	207 RCL 06
	104 PSE	156 TONE 1	

208 X<0?	246 -	283 ST- 10	320 "SOFT
209 GTO 07	247 SQRT	284 "-20 PTS"	LANDING"
210 RCL 01	248 CHS	285 AVIEW	321 AVIEW
211 X>Y?	249 STO 07	286 PSE	322 PSE
212 GTO 06	250 GTO 02	287 GTO 10	323 "VELOCITY: "
213 RCL 03	251*LBL 07	288*LBL 09	324 ARCL 07
214 ST+ 07	252 RCL 01	289 TONE 1	325 AVIEW
215 RCL 07	253 X^2	290 TONE 3	326 PSE
216 X^2	254 RCL 00	291 TONE 4	327 "BONUS:"
217 -.5	255 RCL 03	292 TONE 5	328 AVIEW
218 *	256 *	293 TONE 7	329 PSE
219 RCL 05	257 2	294 TONE 9	330 50
220 /	258 *	295 "EXCELLENT,"	331 RCL 07
221 ST+ 06	259 -	296 AVIEW	332 30
222 "HT= "	260 SQRT	297 PSE	333 *
223 ARCL 06	261 CHS	298 ".. CAPTAIN!"	334 +
224 "" FT"	262 STO 07	299 AVIEW	335 ST+ 10
225 AVIEW	263 GTO 02	300 PSE	336 FIX 0
226 PSE	264*LBL 08	301 RCL 02	337 +
227 RCL 06	265 TONE 9	302 3	338 ARCL X
228 -2	266 TONE 0	303 *	339 ALENG
229 *	267 TONE 9	304 5 E1	340 9
230 RCL 05	268 TONE 0	305 +	341 X<=Y?
231 *	269 TONE 9	306 STO 10	342 "" PTS"
232 SQRT	270 TONE 0	307 +	343 AVIEW
233 CHS	271 TONE 9	308 CLA	344 PSE
234 STO 07	272 TONE 0	309 ARCL X	345*LBL 10
235 GTO 02	273 TONE 0	310 "" PTS"	346 FIX 0
236*LBL 06	274 TONE 0	311 AVIEW	347 "SCORE= "
237 RCL 03	275 "HEAVY	312 PSE	348 ARCL 10
238 ST+ 07	DAMAGE"	313 FIX 1	349 AVIEW
239 RCL 07	276 AVIEW	314 RCL 07	350 CLA
240 X^2	277 PSE	315 ABS	351 STOP
241 RCL 06	278 "VEL= "	316 1.1	352 GTO 00
242 RCL 05	279 ARCL 07	317 X>Y?	<u>353*LBL "LL"</u>
243 *	280 AVIEW	318 GTO 10	354 CLRG
244 2	281 PSE	319 TONE 9	355 GTO 00
245 *	282 2 E2		356 END

Planet Lander for the HP-41C/CV/CX

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

The object here is to perform a vertical descent ending in soft landing on the planet of your choosing. You select the planet before you begin by specifying the acceleration of gravity in feet per second per second. Some values are given below:

Body	$g \text{ (f/s}^2\text{)}$
Earth	32.2
Moon	5.32
Mars	12.3
Ganymede	5.25
Pluto	7.25
Icarus (asteroid)	0.394

For interest, zero and negative values of g are allowed. The fuel allocated, as calculated from g , is more than adequate for a minimum use landing. At least twice as much fuel as needed is given. Although it takes longer to calculate, 3 seconds is the time your burn is stretched over. You can only key a burn in during the zero of each three second count down.

Note that if zero or negative g is selected and you run out of fuel, you may not impact. In this case you will see "DEEP SPACE. . ." instead of the normal "VF=" for final velocity.

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Key in gravity.	$g(\text{f/s}^2)$	[XEQ] GRAVITY	G=
3				FUEL=
				*V=
				A=
				THREE...
				TWO...
				ONE...
				ZERO...

4	You have one second during the "ZERO..." prompt to key in a fuel burn.	burn		
5	Go to 3 for next status.			
6	For a new game with the same g go to step 3.		[A]	G=
7	For a new game with a different g go to step 2.			
*	When you see "VF= ", this is your landing or crash velocity.			

Example

Try a landing on the moon ($g = 5.32 \text{ f/s}^2$).

Keystrokes:	Display:
[XEQ] [ALPHA]	
SIZE [ALPHA] 007	
5.32 [XEQ] [ALPHA]	
GRAVITY [ALPHA]	G=5.32
	FUEL=5456
	V=-500 F/S
	A=5000 V
	THREE...
	TWO...
	ONE...
	ZERO...
(to free fall, just do nothing)	
	FUEL=5456
	V=-516 F/S
	A=3476 F
	THREE...
	TWO...
	ONE...
	ZERO...
20	FUEL=5436
	V=-512 F/S
	A=1 934 F
	THREE...
	TWO...
	ONE...

[Program listing:](#)

<See next page.>

01 LBL "GRAVITY"

02 SF 27	54 "TWO..."	107 RCL 04
03 STO 01	55 AVIEW	108 *
04 ABS	56 PSE	109 RCL 02
05 800	57 "ONE..."	110 +
06 *	58 AVIEW	111 X<> 02
07 1200	59 PSE	112 RCL 00
08 +	60 CLX	113 X^2
09 STO 05	61 "ZERO..."	114 RCL 04
10 LBL A	62 AVIEW	115 *
11 5000	63 PSE	116 2
12 STO 06	64 CLD	117 /
13 -500	65 STO 00	118 X<>Y
14 STO 02	66 ABS	119 RCL 00
15 RCL 05	67 RCL 03	120 *
16 STO 03	68 X>Y?	121 +
17 "G="	69 RDN	122 ST+ 06
18 FIX 02	70 ST- 03	123 GTO 09
19 CF 29	71 RCL 00	124 LBL 02
20 ARCL 01	72 SIGN	125 "DEEP SPACE..."
21 AVIEW	73 *	126 RCL 01
22 PSE	74 3	127 X#0?
23 FIX 00	75 /	128 GTO 03
24 LBL 09	76 RCL 01	129 0
25 "FUEL="	77 -	130 STO 06
26 ARCL 03	78 STO 04	131 RCL 02
27 AVIEW	79 RCL 06	132 X<0?
28 PSE	80 *	133 GTO 09
29 RCL 02	81 2	134 PROMPT
30 RND	82 *	135 LBL 03
31 RCL 06	83 RCL 02	136 RCL 01
32 .5	84 X^2	137 RCL 06
33 "V"	85 X<>Y	138 *
34 X>Y?	86 -	139 2
35 "↑F"	87 SF 00	140 *
36 "↑="	88 X<0?	141 RCL 02
37 ARCL Z	89 GTO 01	142 X^2
38 "↑ F/S"	90 SQRT	143 +
39 AVIEW	91 RCL 02	144 X<0?
40 PSE	92 +	145 PROMPT
41 X>Y?	93 CHS	146 SQRT
42 RTN	94 RCL 04	147 RCL 02
43 "A="	95 X=0?	148 +
44 ARCL 06	96 GTO 01	149 RCL 01
45 "↑ F"	97 CF 00	150 /
46 AVIEW	98 /	151 X<0?
47 PSE	99 3	152 PROMPT
48 RCL 03	100 X<>Y	153 RCL 01
49 X=0?	101 X>Y?	154 *
50 GTO 02	102 RDN	155 ST- 02
51 "THREE..."	103 LBL 01	156 0
52 AVIEW	104 FS?C 00	157 STO 06
53 PSE	105 3	158 GTO 09
	106 STO 00	159 END

Orbital Lander for the HP-41C/CV/CX

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

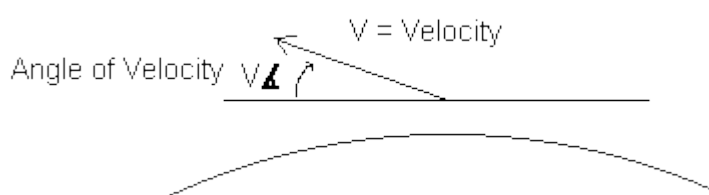
This program simulates a Lunar Excursion Module in orbit 100 km above the surface of the moon. The object is to execute a soft landing (velocity less than 5m/sec, at an angle not more than 5° from vertical) given a limited supply of fuel. On each move, you have the option of either free-falling for a specified period of time, or applying a specified thrust during a specified time period. Thrust is calculated and applied from your input of change in velocity over a given amount of time in a given direction from 0° to +/-180°. Your velocity will not actually change by this amount, of course, since gravity is also acting.

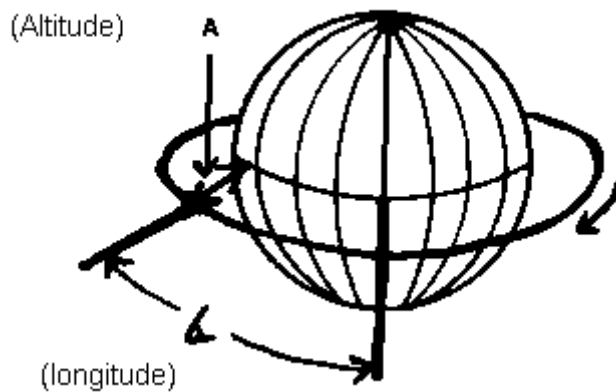
You are not allowed to apply a thrust of greater than 7 Gees (69m/sec/sec of time period). If you run out of fuel, your thrust will be reduced to the fuel supply on hand. Thereafter, any thrust value you provide will be automatically changed to zero. When you pass zero altitude (i.e., land or crash), the program will calculate and display your velocity at impact. (Note to skilled pilots: try also to land at 0° longitude.)

Because the orbital equations are time-independent, the program has to convert the desired "delta-t" into a variable the equations can work with. This conversion process is not completely accurate, but the only error it introduces is that the actual duration of the jump may be slightly different from the one you specify. No positional error is introduced--you will still be on exactly the correct orbit--but you will find yourself at a slightly different point along that orbit.

For example, a 2000 second jump along the initial orbit will take you almost a third of the way around the moon, but the conversion approximation will be about 10 percent shorter than an actual 2000 second jump.

Variable Conventions:





Important: The altitude (A) is from the surface of the moon, not the center.

The angle of velocity (V_A) is given from horizontal, with 0° being forward and 90° straight up. Thrust angles also follow V_A conventions. 180° is a retrofire.

Note: Requires 1 Memory Module on HP-41C

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] ORBIT	
3	*Mission status: altitude			A=
	longitude		[R/S]	<)=
	velocity		[R/S]	V=
	angle of flight		[R/S]	V<)=
	fuel remaining		[R/S]	F=
4	To free fall: key in number of seconds. Go to step 3 for outputs.	n	[A]	
	Go to step 3 for outputs.			
5	To fire rockets: key in total change in V	dV(m/s)	[ENTER]	
	key in angle of thrust	θ (deg)	[ENTER]	
	key in number of seconds for total burn	n (sec)	[B]	
	Go to step 3 for outputs.			

	When A=0.00, you are down.			
*	Continuing [R/S] will repeat status.			

Example

Keystrokes:	Display:
[XEQ] [ALPHA]	
SIZE [ALPHA] 015	
[XEQ] [ALPHA]	
ORBIT [ALPHA]	A=100000.00 M (altitude)
[R/S]	<)=0.00 (longitude)
[R/S]	V=1631.77 M/S (velocity)
[R/S]	V<)=0.00 (angle from horizontal)
[R/S]	F=2,000.00 (fuel)
1000 [A]	A=99,957.06 M
[R/S]	<)=55.65
[R/S]	V=1,631.80 M/S
[R/S]	V<)= 0.00
[R/S]	F=2,000.00

For 10 seconds apply 7 gravities as retrofire

69 [ENTER] 10 [X]	
180 [ENTER]	
10 [B]	A=99,908.77 M
[R/S]	<)=55.95
[R/S]	V=941.88 M/S
[R/S]	V<)= -0.59
[R/S]	F=1,310.00
200 [A]	A=78,392.28 M
[R/S]	<)=61.89
[R/S]	V=974.77 M/S
[R/S]	V<)= -12.14
[R/S]	F=1,310.00
.	.
.	.

Program listing:

01 LBL "ORBIT"	15 0	31 ST+ 05	47 RCL 01
02 SF 27	16 GTO 01	32 RCL 05	48 /
03 CLRG	17 LBL B	33 9	49 -
04 CF 05	18 STO 12	34 X>Y?	50 STO 06
05 2000	19 69	35 GTO 21	51 RCL 01
06 STO 00	20 *	36 R^	52 RCL 05
07 1839000	21 R^	37 ST+ 04	53 *
08 STO 01	22 RCL 00	38 LBL 01	54 STO 07
09 4.89663	23 X<=Y?	39 RCL 04	55 X^2
E12	24 X<>Y	40 X^2	56 RCL 03
10 STO 03	25 RDN	41 RCL 05	57 /
11	26 X<=Y?	42 X^2	58 STO 08
1631.765625	27 X<>Y	43 +	59 *
12 STO 05	28 RDN	44 2	60 2
13 1739000	29 ST- 00	45 /	61 *
14 STO 11	30 P-R	46 RCL 03	62 RCL 03

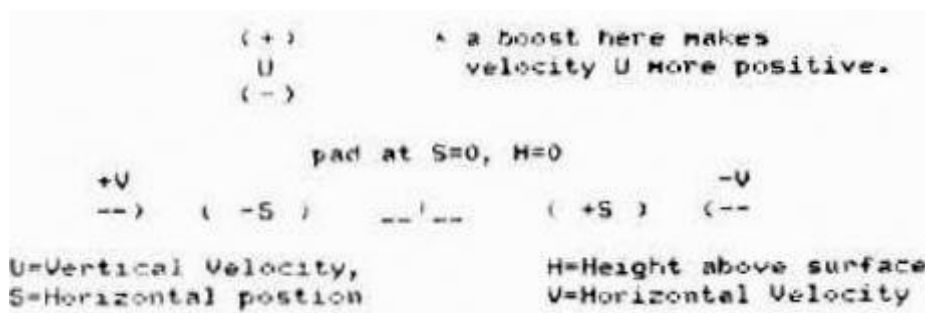
63 /	113 X<>Y	163 RCL 13	212 /
64 1	114 -	164 P-R	213 CHS
65 +	115 RCL 12	165 STO 05	214 RCL 05
66 SQRT	116 *	166 RDN	215 +
67 STO 09	117 RCL 01	167 STO 04	216 RCL 12
68 RCL 08	118 +	168 RCL 01	217 *
69 RCL 01	119 R-P	169 RCL 11	218 X<>Y
70 /	120 RDN	170 -	219 RCL 03
71 1	121 ST+ 02	171 X<0?	220 RCL 01
72 -	122 RCL 08	172 GTO 22	221 X^2
73 RCL 09	123 RCL 09	173 LBL 10	222 /
74 /	124 RCL 02	174 FIX 02	223 RCL 12
75 FIX 07	125 RCL 10	175 ADV	224 *
76 RND	126 -	176 "A="	225 -
77 ACOS	127 COS	177 RCL 01	226 ST+ 04
78 RCL 04	128 *	178 RCL 11	227 2
79 RCL 05	129 1	179 -	228 /
80 *	130 +	180 X<0?	229 CHS
81 X>0?	131 /	181 CLX	230 RCL 04
82 SF 05	132 STO 01	182 ARCL X	231 +
83 RDN	133 RCL 03	183 "┐ M"	232 RCL 12
84 FS?C 05	134 X<>Y	184 AVIEW	233 *
85 CHS	135 /	185 STOP	234 RCL 01
86 RCL 02	136 RCL 06	186 "<)"="	235 +
87 +	137 +	; "\0D="	236 R-P
88 360	138 2	187 RCL 02	237 STO 01
89 MOD	139 *	188 1	238 X<>Y
90 STO 10	140 SQRT	189 P-R	239 ST+ 02
91 RCL 12	141 STO 13	190 R-P	240 LBL 00
92 LBL A	142 RCL 01	191 ARCL Y	241 RCL 04
93 STO 12	143 *	192 AVIEW	242 RCL 05
94 0	144 RCL 07	193 STOP	243 R-P
95 ENTER	145 X<>Y	194 "V="	244 STO 13
96 ENTER	146 /	195 ARCL 13	245 X<>Y
97 RCL 05	147 FIX 07	196 "┐ M/S"	246 GTO 20
98 9	148 RND	197 AVIEW	247 LBL 22
99 X>Y?	149 ACOS	198 STOP	248 ST- 01
100 GTO 21	150 RCL 07	199 "V<)"="	249 3
101 RDN	151 RCL 02	200 ARCL 14	250 *
102 RCL 12	152 RCL 10	201 AVIEW	251 RCL 04
103 *	153 -	202 STOP	252 X^2
104 RCL 03	154 SIN	203 "F="	253 +
105 RCL 01	155 *	204 ARCL 00	254 ABS
106 X^2	156 X<0?	205 AVIEW	255 SQRT
107 /	157 SF 05	206 STOP	256 CHS
108 RCL 12	158 RDN	207 GTO 10	257 STO 04
109 *	159 FS?C 05	208 LBL 21	258 GTO 00
110 2	160 CHS	209 RDN	259 END
111 /	161 LBL 20	210 RDN	
112 RCL 04	162 STO 14	211 2	

Planet Lander v2

Mark Gesner- PPCCJ V11N9 p39 ; (Nov/Dec 1984)

This game allows you to land on a planet of your choice, in a planetary lander craft of your own imagining. The program was written mostly by the friend of an old college roommate, so I don't have any comments on the individual program lines, except lines 03 and 06, which are not designed to be AVIEWed, but are only comment lines. This was the first time I had ever seen ALPHA strings used as in-program comment lines. I like the idea, so long as space is plentiful

Firing diagram: Rocket at H



Use the firing diagram to interpret the information the computer gives you. Send fuel to the appropriate booster rockets to adjust position, velocity and height. When you get sick of landing on the moon, change the value of gravity in line 07 to suit your interplanetary tastes. The fuel factor in line 04 determines the potency of the fuel mixture in your tanks. When you develop expertise in landing on different planets, try some weaker or stronger fuel. The higher the fuel factor, the more potent the mix.

I would like to see other PPC lander games. This one is pretty basic but the program is badly in need of streamlining. The time module should be employed to make a real-time simulator. Is there anyone out there who has a game which simulates the APOLLO missions precisely? That would be fun.

Happy landing!

Mark D. Gessner (11922)
401 Stashey 107
College Station, TX 77840
USA

Program listing:

01 LBL "LANDER"	51 170	101 " ^"	151 RCL 08
02 CF 27	52 +	102 PROMPT	152 ST+ 17
03 "FUEL FACTOR:"	53 INT	103 STO 42	153 RCL 07
04 3	54 STO 02	104 CLX	154 X>0?
05 STO 04	55 GTO 01	105 " <--"	155 GTO 01
06 "GRAVITY:"	56 LBL 27	106 PROMPT	156 LBL 10
07 -5.3667	57 E	107 STO 43	157 RCL 16
08 STO 05	58 STO 48	108 RCL 42	158 ABS
09 "PLAYER?"	59 RTN	109 +	159 5
10 PROMPT	60 LBL 01	110 RCL 41	160 X<>Y
11 20	61 .	111 +	161 X>Y?
12 +	62 STO 48	112 STO 14	162 XEQ 15
13 STO 10	63 RCL 06	113 RCL 42	163 RCL 07
14 10	64 E2	114 X=0?	164 -5
15 +	65 X>Y?	115 GTO 22	165 X<>Y
16 STO 11	66 XEQ 27	116 RCL 41	166 X>Y?
17 LBL 00	67 FIX IND 48	117 RCL 43	167 GTO 06
18 1	68 RCL 06	118 -	168 -10
19 CF 02	69 RND	119 RCL 42	169 X<>Y
20 ST+ IND 11	70 STO 00	120 /	170 X>Y?
21 RCL IND 10	71 RCL 07	121 CHS	171 GTO 07
22 RAD	72 RND	122 ATAN	172 LBL 19
23 SIN	73 STO 01	123 STO 15	173 3
24 DEG	74 RCL 16	124 GTO 23	174 SQRT
25 E^X	75 RND	125 LBL 22	175 ABS
26 2 E2	76 STO 16	126 RCL 41	176 LOG
27 *	77 RCL 17	127 RCL 43	177 3
28 5 E1	78 RND	128 -	178 SQRT
29 +	79 STO 17	129 X>0?	179 LOG
30 STO 06	80 CLA	130 GTO 24	180 3
31 3 E2	81 LBL C	131 90	181 ABS
32 -	82 "U="	132 STO 15	182 5
33 STO 16	83 ARCL 01	133 GTO 23	183 +
34 RCL 06	84 ">" H="	134 LBL 24	184 "CRASHED"
35 .85	85 ARCL 00	135 -90	185 AVIEW
36 Y^X	86 PROMPT	136 STO 15	186 PSE
37 CHS	87 "S="	137 LBL 23	187 "VEL="
38 STO 07	88 ARCL 16	138 RCL 14	188 ARCL 07
39 .595	89 ">" V="	139 RCL 02	189 AVIEW
40 *	90 ARCL 17	140 X<>Y	190 PSE
41 STO 17	91 PROMPT	141 X>Y?	191 ARCL 16
42 RCL 07	92 "FUEL="	142 GTO 02	192 AVIEW
43 ABS	93 ARCL 02	143 ST- 02	193 PSE
44 2.8	94 AVIEW	144 XEQ 03	194 5 E1
45 Y^X	95 PSE	145 X>0?	195 RCL 07
46 1/X	96 CLX	146 GTO 04	196 ABS
47 RCL 06	97 "-->"	147 X<0?	197 +
48 3.1	98 PROMPT	148 GTO 05	198 RCL 16
49 Y^X	99 STO 41	149 RCL 03	199 ABS
50 *	100 CLX	150 ST+ 07	200 +

201 ST- IND 10	253 RCL 05	305 /	357 FS? 02
202 CHS	254 /	306 RCL 07	358 GTO 19
203 CLA	255 -.5	307 +	359 "NICE TOUCH"
204 " "	256 *	308 ST+ 06	360 AVIEW
205 ARCL X	257 ST+ 06	309 RCL 06	361 RCL 02
206 >" PTS"	258 "HT="	310 RTN	362 3
207 AVIEW	259 ARCL 06	311 LBL 04	363 *
208 PSE	260 AVIEW	312 RCL 03	364 5 E1
209 LBL 11	261 PSE	313 ST+ 07	365 +
210 "SCORE="	262 RCL 06	314 RCL 08	366 ST+ IND 10
211 ARCL IND 10	263 RCL 05	315 ST+ 17	367 +
212 AVIEW	264 -2	316 GTO 01	368 PSE
213 PSE	265 *	317 LBL 05	369 ARCL X
214 RCL IND 10	266 *	318 RCL 01	370 >" PTS"
215 RCL IND 11	267 SQRT	319 X^2	371 AVIEW
216 /	268 CHS	320 RCL 00	372 PSE
217 "AVG="	269 STO 07	321 RCL 03	373 RCL 07
218 ARCL X	270 GTO 10	322 *	374 ABS
219 AVIEW	271 LBL 03	323 2	375 1.1
220 PSE	272 RCL 41	324 *	376 X<=Y?
221 "VEL="	273 RCL 43	325 -	377 GTO 11
222 ARCL 07	274 +	326 SQRT	378 "SOFT "
223 AVIEW	275 X^2	327 CHS	379 AVIEW
224 PSE	276 RCL 42	328 STO 07	380 PSE
225 PSE	277 X^2	329 RCL 01	381 "BONUS+ 100"
226 "S="	278 +	330 X^2	382 AVIEW
227 ARCL 16	279 SQRT	331 2	383 PSE
228 AVIEW	280 STO 14	332 RCL 03	384 E2
229 PSE	281 RCL 14	333 *	385 ST+ IND 10
230 PSE	282 RCL 15	334 RCL 00	386 GTO 11
231 CLA	283 COS	335 *	387 LBL 07
232 FIX 4	284 *	336 -	388 FS? 02
233 STOP	285 RCL 04	337 SQRT	389 GTO 19
234 GTO 00	286 *	338 CHS	390 "HEAVY
235 LBL 02	287 RCL 05	339 RCL 01	DAMAGE"
236 RCL 02	288 +	340 CHS	391 AVIEW
237 STO 14	289 STO 03	341 +	392 PSE
238 XEQ 03	290 RCL 14	342 RCL 03	393 "VEL="
239 X<0?	291 RCL 15	343 /	394 ARCL 07
240 GTO 05	292 SIN	344 STO 09	395 AVIEW
241 X=0?	293 RCL 04	345 X^2	396 2 E1
242 GTO 05	294 *	346 RCL 08	397 ST- IND 10
243 RCL 00	295 *	347 *	398 GTO 11
244 X<>Y	296 CHS	348 2	399 LBL 08
245 X<Y?	297 STO 08	349 /	400 RCL 03
246 GTO 08	298 2	350 RCL 09	401 ST+ 07
247 RCL 03	299 /	351 RCL 17	402 RCL 08
248 ST+ 07	300 RCL 17	352 *	403 ST+ 17
249 RCL 08	301 +	353 +	404 RCL 07
250 ST+ 17	302 ST+ 16	354 ST+ 16	405 X^2
251 RCL 07	303 RCL 03	355 GTO 10	406 RCL 06
252 X^2	304 2	356 LBL 06	407 RCL 05

408 *	418 RCL 07	428 RCL 07	438 "NO
409 2	419 X^2	429 CHS	SURVIVORS"
410 *	420 2	430 +	439 AVIEW
411 -	421 RCL 05	431 RCL 05	440 TONE 0
412 SQRT	422 *	432 /	441 CF 02
413 CHS	423 RCL 06	433 RCL 17	442 RTN
414 STO 07	424 *	434 *	443 END
415 XEQ 17	425 -	435 ST+ 16	
416 GTO 10	426 SQRT	436 RTN	
417 LBL 17	427 CHS	437 LBL 15	

And yet another Lander (v4)

Whodunit – MoHP Disks

Obviously there's no shortage of these; here's another lander for you, again with no documentaion or references to the authors. This one comes form the Museum of HP Cals Disks (only the program listing was there).

Program listing:

1 <u>LBL "LANDER"</u>	22 PROMPT	43 SF 01
2 LBL E	23 FC?C 22	44 ST/ Y
3 1737720	24 700	45 ST/ Z
4 STO 09	25 STO 03	46 LBL 07
5 STO 01	26 ST+ 10	47 „V=“
6 E3	27 LBL C	48 ARCL Z
7 STO 10	28 „FUEL=“	49 >“ H=“
8 FIX 1	29 ARCL 03	50 ARCL Y
9 SF 27	30 AVIEW	51 CLX
10 CF 22	31 PSE	52 STO 00
11 „HEIGHT?“	32 LBL 02	53 PROMPT
12 PROMPT	33 RCL 02	54 RCL 09
13 FC?C 22	34 STO 04	55 RCL 09
14 2350	35 RCL 02	56 RCL 03
15 ST+ 01	36 RCL 01	57 -
16 „VEL?“	37 RCL 09	58 SF 05
17 PROMPT	38 -	59 X=Y?
18 FC?C 22	39 E3	60 GTO 08
19 -470	40 CF 01	61 CF 05
20 STO 02	41 X>Y?	62 LASTX
21 „FUEL?“	42 GTO 07	63 RCL T

64 X>Y?	95 LBL 08	126 GTO 03
65 GTO C	96 STO 06	127 X>0?
66 ST- 03	97 RCL 00	128 GTO 02
67 STO 04	98 CHS	129 „PERFECT LANDING“
68 RCL 10	99 4.92 E12	130 ACIEW
69 RCL 04	100 RCL 01	131 GTO 04
70 -	101 STO 07	132 LBL 03
71 STO 05	102 /	133 CF 01
72 RCL 10	103 RCL 06	134 RCL 04
73 /	104 RND	135 X^2
74 LN	105 STO 01	136 RCL 07
75 E3	106 /	137 RCL 09
76 *	107 -	138 -
77 STO 00	108 STO 00	139 2
78 2	109 RND	140 *
79 /	110 RCL 02	141 RCL 00
80 RCL 02	111 STO 04	142 *
81 -	112 +	143 -
82 RCL 01	113 STO 02	144 SQRT
83 -	114 RCL 05	145 CHS
84 2	115 STO 10	146 „** CRASH **“
85 /	116 RCL 01	147 AVIEW
86 STO 06	117 RCL 09	148 PSE
87 X^2	118 -	149 „TERMINAL VEL=“
88 2.46 E12	119 X>0?	150 ARCL X
89 RCL 01	120 GTO 02	151 AVIEW
90 /	121 FC?C 05	152 LBL 04
91 -	122 X<0?	153 FIX 3
92 SQRT	123 GTO 03	154 END
93 RCL 06	124 RCL 02	
94 -	125 X<0?	
155		

Moon Lander (French version)

Whodunit - Swap disks

1 LBL "LEM"	47 >" SEC."	93 CLX	135 GTO 24
2 LBL 00	48 AVIEW	94 STO 09	136 X<>Y
3 CLRG	49 PSE	95 GTO 04	137 RCL 03
4 FIX 00	50 "ALT:"	96 LBL 10	138 X<Y?
5 CF 29	51 RCL 02	97 " DEBIT ?"	139 GTO 04
6 CF 05	52 INT	98 PROMPT	140 STO 04
7 120	53 ARCL X	99 STO 10	141 RCL 13
8 STO 01	54 >","	100 X=0?	142 RCL 10
9 1	55 RCL 01	101 GTO 13	143 RCL 04
10 STO 18	56 1.609	102 4	144 *
11 32500	57 *	103 X>Y?	145 RCL 14
12 STO 13	58 RCL 02	104 GTO 11	146 +
13 16500	59 -	105 X<>Y	147 X<=Y?
14 STO 14	60 E3	106 90	148 GTO 19
15 .001	61 *	107 X<>Y	149 RCL 13
16 STO 07	62 .5	108 X<=Y?	150 RCL 14
17 1.8	63 +	109 GTO 13	151 -
18 STO 00	64 INT	110 LBL 11	152 RCL 10
19 LBL 04	65 ARCL X	111 TONE 07	153 /
20 1	66 >" KM"	112 TONE 06	154 STO 04
21 ST+ 09	67 AVIEW	113 "DEBIT	155 LBL 19
22 RCL 01	68 PSE	IMPOSSIBL	156 XEQ 55
23 1.609	69 "VIT:"	"	157 RCL 06
24 *	70 RCL 19	114 >"E"	158 X<=0?
25 INT	71 INT	115 AVIEW	159 GTO 39
26 STO 02	72 ARCL X	116 1	160 RCL 16
27 RCL 18	73 >" KM/H"	117 ST- 09	161 X<0?
28 5792.4	74 AVIEW	118 GTO 04	162 GTO 21
29 *	75 PSE	119 LBL 13	163 XEQ 63
30 STO 19	76 "FUEL="	120 "UN	164 GTO 15
31 .36	77 RCL 15	INSTANT...	165 LBL 21
32 /	78 INT	"	166 RCL 18
33 STO 17	79 ARCL X	121 AVIEW	167 X>0?
34 RCL 13	80 >" KG"	122 10	168 GTO 43
35 RCL 14	81 AVIEW	123 STO 03	169 XEQ 63
36 -	82 PSE	124 RCL 10	170 GTO 15
37 .45359	83 "d./10 S.="	125 ,45353	171 LBL 24
38 *	84 RCL 17	126 /	172 BEEP
39 STO 15	85 INT	127 STO 10	173 SF 05
40 CLX	86 ARCL X	128 LBL 15	174 FIX 03
41 STO 11	87 AVIEW	129 RCL 13	175 "PANNE DE
42 TONE 09	88 PSE	130 RCL 14	CARBUR"
43 "T="	89 RCL 09	131 -	176 >"ANT
44 RCL 12	90 X#0?	132 ,001	APRES"
45 INT	91 GTO 10	133 X<>Y	177 AVIEW
46 ARCL X	92 XEQ 79	134 X<Y?	178 >" "

179 ARCL 12	228 RCL 13	275 PROMPT	322 *
180 >" SEC."	229 RCL 14	276 GTO 00	323 STO 04
181 AVIEW	230 -	277 LBL 33	324 XEQ 55
182 PSE	231 ,45359	278 BEEP	325 XEQ 63
183 RCL 01	232 *	279 BEEP	326 GTO 39
184 RCL 07	233 ARCL X	280 "ALUNISSA	327 LBL 43
185 2	234 >" KG"	GE DOUT"	328 RCL 00
186 *	235 AVIEW	281 >"EUX"	329 RCL 10
187 *	236 PSE	282 PROMPT	330 *
188 RCL 18	237 E	283 GTO 00	331 1/X
189 X^2	238 RCL 05	284 LBL 34	332 RCL 07
190 +	239 X<Y?	285 BEEP	333 *
191 SQRT	240 GTO 29	286 "MATERIEL	334 RCL 13
192 RCL 18	241 E1	ENDOMM"	335 *
193 -	242 X>Y?	287 >"AGE."	336 CHS
194 RCL 07	243 GTO 31	288 AVIEW	337 1
195 /	244 X<>Y	289 PSE	338 +
196 STO 04	245 25	290 "BONNE	339 2
197 RCL 07	246 X>Y?	CHANCE	340 /
198 *	247 GTO 33	PO"	341 STO 05
199 ST+ 18	248 X<>Y	291 AVIEW	342 X^2
200 RCL 04	249 60	292 >"UR LE	343 RCL 18
201 ST+ 12	250 X>Y?	RETOUR"	344 +
202 LBL 27	251 GTO 34	293 PROMPT	345 SQRT
203 FC?C 05	252 TONE 06	294 GTO 00	346 RCL 05
204 BEEP	253 TONE 06	295 LBL 39	347 +
205 FIX 03	254 TONE 06	296 ,005	348 RCL 10
206 RCL 18	255 TONE 05	297 RCL 04	349 *
207 3600	256 "DESOLE,	298 X<Y?	350 RCL 00
208 *	AUCUN S"	299 GTO 27	351 *
209 STO 05	257 >"URVIVA	300 RCL 00	352 1/X
210 "SUR LA	NT"	301 RCL 10	353 RCL 18
LUNE EN "	258 PROMPT	302 *	354 *
211 AVIEW	259 GTO 00	303 RCL 13	355 RCL 13
212 ARCL 12	260 LBL 29	304 /	356 *
213 >" SEC."	261 BEEP	305 CHS	357 ,5
214 AVIEW	262 BEEP	306 RCL 07	358 +
215 PSE	263 BEEP	307 +	359 STO 04
216 "VITESSE	264 BEEP	308 2	360 XEQ 55
AU CONT"	265 "ALUNISSA	309 *	361 RCL 06
217 >"ACT:"	GE PARF"	310 RCL 01	362 X<=0?
218 AVIEW	266 >"AIT"	311 *	363 GTO 39
219 RCL 05	267 PROMPT	312 RCL 18	364 XEQ 63
220 1.609	268 GTO 00	313 X^2	365 RCL 18
221 *	269 LBL 31	314 +	366 X<=0?
222 ARCL X	270 BEEP	315 SQRT	367 GTO 15
223 >" KM/H"	271 BEEP	316 RCL 18	368 RCL 16
224 AVIEW	272 BEEP	317 +	369 X<0?
225 PSE	273 "ASSEZ	318 1/X	370 GTO 43
226 "FUEL	BON	319 2	371 GTO 15
RESTANT:"	ALUNI"	320 *	372 LBL 55
227 FIX 00	274 >"SSAGE"	321 RCL 01	373 RCL 04

374 RCL 10	400 /	426 Y^X	452 -
375 *	401 +	427 20	453 RCL 07
376 RCL 13	402 RCL 08	428 /	454 RCL 04
377 /	403 X^2	429 +	455 X^2
378 STO 08	404 2	430 RCL 08	456 *
379 E-7	405 /	431 3	457 2
380 X<=Y?	406 +	432 Y^X	458 /
381 GTO 57	407 RCL 08	433 12	459 -
382 CLX	408 +	434 /	460 RCL 01
383 STO 08	409 CHS	435 +	461 +
384 LBL 57	410 RCL 00	436 RCL 08	462 STO 06
385 RCL 08	411 *	437 X^2	463 RTN
386 5	412 RCL 07	438 6	464 LBL 63
387 Y^X	413 RCL 04	439 /	465 RCL 04
388 5	414 *	440 +	466 ST+ 12
389 /	415 +	441 RCL 08	467 ST- 03
390 RCL 08	416 RCL 18	442 2	468 RCL 10
391 4	417 +	443 /	469 *
392 Y^X	418 STO 16	444 +	470 ST- 13
393 4	419 RCL 08	445 RCL 04	471 RCL 06
394 /	420 5	446 RCL 00	472 STO 01
395 +	421 Y^X	447 *	473 RCL 16
396 RCL 08	422 30	448 *	474 STO 18
397 3	423 /	449 RCL 04	475 RTN
398 Y^X	424 RCL 08	450 RCL 18	476 END
399 3	425 4	451 *	

Pinball Wizard.

Craig Pearce - Games Solution Book

Welcome to the "Wizard of Pinball" game. This program simulates, as closely as possible, the actual play in a genuine pinball machine. The user interacts with the game through the digit keys 1 and 3 (designated the left and right flippers respectively), and the digit 2, which is the tilt option. Failing to hit the correct flipper will still leave the user the solution of "tilting" the machine and placing the ball back in play (maybe!).

The "Wizard of Pinball" allows from 1 to 4 players, with play alternating from player 1 to player 2 and so on back to player 1. Each player will receive a total of 5 balls for each game. The ability to win a free ball is also possible. In this case, the same player stays until the extra ball is lost, after which the play rotates to the next player (unless another free ball is won).

Shooting the ball is accomplished by pressing any numeric key. As in most genuine pinball games, the "Wizard of Pinball" returns the same ball to the same player to be reshot if no score was made and the ball exits immediately. The game allows up to three free game thresholds that award a credit (free game) when passed. Also, the program checks for a score that passes the previous "high - score to date". Another free game is awarded if any or all of the players pass this previously stored "HI-SCR".

THE DEVICES

Listed below are the several different scoring devices used in the program. The "device" name is given first as it is displayed on the HP-41. The fullname of the device is given in parentheses after the formatted name, followed by a brief description of the device and of how it scores.

"*STAR-50" (Star Rollovers)

These are stars, like buttons on the playfield. Each time the ball rolls over one of these buttons, the player receives 50 points.

"*LANE-300" (Lane Rollovers)

Lane roll overs are special paths that the ball travels through and scores an immediate 300 points for the player.

"THUMP-x00". (Thumper-Bumpers)

Sometimes called "Jet or Pop Bumpers". In this game 100 points are scored each time the ball strikes the bumpers. At any given time, the ball can bounce 1 to 10 times, scoring 100 to 1000 points. When this display comes up, the value of "x00" is the amount of points scored; x being the number of bumps the ball made.

"SPIN-xy0" (Spinner Gate)

Spinner gates on pinball machines are the devices that spin on a horizontal axis as the ball passes under them. In this game, the spinner gate can spin up to 25 times, scoring 10 points

for each spin, and showing the actual points made (also a tone is heard for each spin). In addition, for each 5 spins of the gate, the Out Bonus is advanced by 1000 points.

"KICK-xOO" (Kick Out Hole)

Kick out holes (or saucers) are those devices that the ball drops into, scores some points and is kicked back out into play. On "Wizard", the points for the kick out hole begins at 2000 and advances by 2000 each time the ball drops in one, until a point value of 10,000 is reached. This value is held for all additional hits of the hole.

"SLING-IO" (Sling Shot Kickers)

The sling shot kickers are devices that propel the ball away when struck, and score 10 points.

"DROP-X" (Drop Targets)

Drop targets are scoring devices that fall away when struck, and score some points in the process. They are reset with each new ball, or when they are all down, which is a special case. In this game, there are three drop targets. Hitting the first and the second results in an immediate 10 points and a display of "DROP-1" and "DROP-2". When the third is hit, the player receives 100 points and is awarded another ball. The display will show "SHOOT AGAIN". Although the targets are reset and can be knocked down again, only one extra ball can be earned per ball in play. When the current ball is lost, the same player then plays the "extra ball" (it is possible to win another extra ball with the free ball currently in play).

"*A-" to "*F-" (Alpha Targets)

These are stationary targets that award the player an immediate 500 points each time they are struck. Also, during the play of anyone's ball, the calculator remembers the targets hit (in any order) and provides for higher Out Bonus scoring as follows:

- Hitting A & B displays "BONUS x 2" and the player will receive twice the Out Bonus when the ball exits.
- Getting A.B.C, & 0 displays "BONUS x 3" with resulting triple Out Bonus score.
- If all 6 targets are hit in one turn the Out Bonus is quintupled.

OUT BONUS.

All of the above devices whose formatted name begins with a"" increase the Out Bonus by 1000 points (unless otherwise stated). When the ball exits the OUT HOLE, the player collects all the Out Bonus points accumulated during that play. The maximum limit on Out Bonus points is 29,000. This value is then multiplied by the "BONUS x" factor, allowing for a maximum of 145,000 points when the ball exits. The display shows the total out bonus points and decrements this count by 1000, adding 1000 points to the player's score each time.

FLIPPERS. TILTING. AND OUT

When the ball reaches the left or right flipper the display will show "LEFT:1" or "RIGHT:3" (respectively). At this point the player has approximately one second to press the

appropriate key (1 or 3) in order to put the ball back into play. Failure to hit the proper key will result in the ball exiting through the Out Hole.

Whenever the ball enters the Out Hole, the display shows "OUT" for about one second. During this time the player has the option of TILTING the machine in a last attempt to put the ball back in play. Tilting is accomplished by pressing key "2" while OUT is displayed. The chances are 4 out of 5 that the ball will be placed back in play. However, if that one remaining chance comes up, the display will show "*TILT*" and all bonus points are lost! Also, any free ball gained during that turn will be lost! The player's score is displayed and the play moves on.

END OF GAME

When the end of the game is reached, and the last player's score is reviewed, the program will select a random number as the MATCH DIGIT. This number is always between 00 & 90 (multiples of 10) and is compared against the last digits of the player(s) score(s). If any player has a match a free game is awarded. The display will be shown as "MATCHxO", this is the number chosen by the calculator. If any player's score has passed the high score to date. The old HISCR is changed; otherwise it remains the same. The high score will be displayed as "HISCR-xxx,xxO". If any (or all) player score(s) passed the HISCR, a CREDIT is awarded.

Finally, all the player's scores are reviewed a final time and compared against the 3 free game thresholds. For each player whose score passes each of these thresholds, another free game is credited. Finally, the display shows GAMEOVER.

Data Set

To run the program, load the data and then start the program with XEQ "WIZARD".

RR000 = "OUT"	RR024 = "HISCR-"
RR001 = "*STAR-"	RR025 = "SCORE-"
RR002 = "*LANE-"	RR026 = "X 2"
RR003 = "THUMP-"	RR027 = "X 3"
RR004 = "SPIN-"	RR028 = "CREDIT"
RR005 = "KICK-"	RR029 = "X 5"
RR006 = "SLING-"	RR030 = "PLAYER"
RR007 = "DROP-"	RR031 = "*TILT*"
RR008 = "*A-"	RR032 = +0.000000000E+0
RR009 = "*B-"	RR033 = +0.000000000E+0
RR010 = "*C-"	RR034 = +0.000000000E+0
RR011 = "*D-"	RR035 = +0.000000000E+0
RR012 = "*E-"	RR036 = +0.000000000E+0
RR013 = "*F-"	RR037 = +1.000000000E+1
RR014 = "RGHT-3"	RR038 = +5.000000000E+1
RR015 = "BALL "	RR039 = +1.000000000E+2
RR016 = "LEFT-1"	RR040 = +1.000000000E+3
RR017 = +0.000000000E+0	RR041 = +0.000000000E+0
RR018 = "GAME "	RR042 = +0.000000000E+0
RR019 = " OVER"	RR043 = +0.000000000E+0
RR020 = "BONUS "	RR044 = +0.000000000E+0
RR021 = "SHOOT"	RR045 = +0.000000000E+0
RR022 = "AGAIN"	RR046 = +0.000000000E+0
RR023 = "MATCH-"	RR047 = +0.000000000E+0

RR048 = +0.000000000E+0
 RR049 = +0.000000000E+0
 RR050 = +0.000000000E+0
 RR051 = +0.000000000E+0
 RR052 = +0.000000000E+0
 RR053 = +0.000000000E+0
 RR054 = +0.000000000E+0
 RR055 = +0.000000000E+0

RR056 = +0.000000000E+0
 RR057 = +0.000000000E+0
 RR058 = +0.000000000E+0
 RR059 = +0.000000000E+0
 RR060 = +0.000000000E+0
 RR061 = +0.000000000E+0
 RR062 = +0.000000000E+0
 RR063 = +0.000000000E+0

Program listing:

01*LBL "WIZARD"

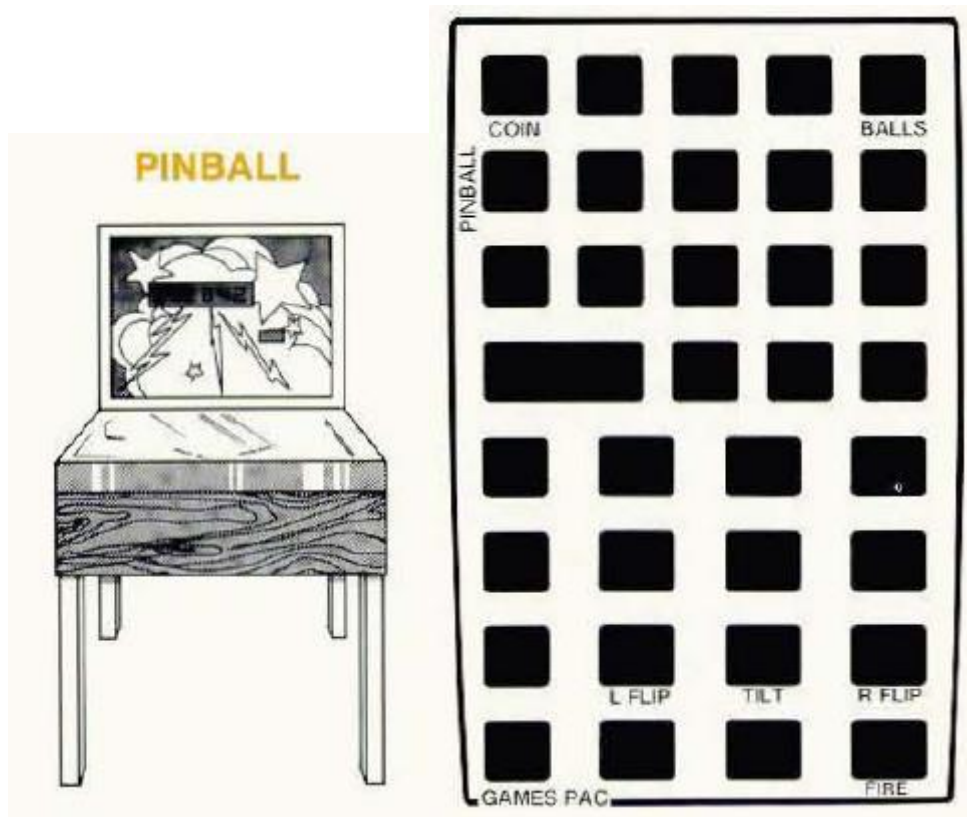
02 64	41 RCL 53	81 +	121 " NO. "
03 XROM "INIT"	42 E	82 0	122 ARCL 48
04 XROM "PDAT"	43 X#Y?	83*LBL 12	123 AVIEW
05 SF 27	44 "S"	84 STO IND Y	124 PSE
06 " CAP-PINBALL"	45 AVIEW	85 ISG Y	125 CLA
07 AVIEW	46 RTN	86 GTO 12	126 ARCL 15
08*LBL a	47*LBL B	87 E	127 ARCL 50
09 .010	48 RCL 53	88 STO 48	128 CF 22
10*LBL 10	49 X=0?	89 STO 50	129*LBL 15
11 CF IND X	50 GTO D	90 TONE 9	130 AVIEW
12 ISG X	51 0	91 TONE 8	131 PSE
13 GTO 10	52 FS?C 01	92 TONE 8	132 FC?C 22
14 SF 01	53 STO 45	93 TONE 8	133 GTO 15
15 CF 20	54 RCL 45	94 TONE 7	134*LBL 16
16 CF 21	55 4	95 TONE 7	135 12
17 FS? 55	56 X=Y?	96 BEEP	136 XEQ 09
18 SF 21	57 GTO 11	97 TONE 8	137 4
19 RTN	58 E	98 TONE 8	138 -
20*LBL E	59 ST+ 45	99 TONE 8	139 X>0?
21 FIX 2	60 ST- 53	100 TONE 9	140 GTO 17
22 "\$ "	61 TONE 5	101*LBL 13	141 FS? 02
23 ARCL 52	62 XEQ D	102 41	142 GTO 18
24 FIX 0	63 PSE	103 STO 58	143 "NO "
25 AVIEW	64*LBL 11	104 E	144 ARCL 25
26 RTN	65 CLA	105 STO 48	145 ARCL 21
27*LBL C	66 ARCL 30	106*LBL 14	146 ARCL 22
28 .25	67 "S ="	107 E	147 AVIEW
29 ST+ 52	68 ARCL 45	108 STO 46	148 PSE
30 TONE 4	69 AVIEW	109 3	149 GTO 14
31 39	70 RTN	110 E3/E+	150*LBL 17
32 RCL 53	71*LBL A	111 STO 51	151 SF 02
33 X>Y?	72 RCL 45	112 0	152 CLA
34 GTO D	73 X=0?	113 STO 47	153 ARCL IND X
35 ISG 53	74 RTN	114 25	154 GTO IND X
36*LBL D	75 40	115 STO 49	155*LBL 18
37 CLA	76 +	116 XEQ a	156 -2
38 ARCL 53	77 RCL 40	117 SF 01	157 X<>Y
39 " "	78 /	118 CF 02	158 X=Y?
40 ARCL 28	79 41	119 CLA	159 0
	80 STO 58	120 ARCL 30	160 X<0?

161 GTO 19	213 AVIEW	265 CLA	317*LBL 09
162 RCL 00	214 FS? 03	266 ARCL 23	318 RNG
163 CLD	215 GTO 14	267 ARCL 35	319 *
164 PSE	216 RCL 45	268 AVIEW	320 E
165 FC?C 22	217 RCL 48	269 PSE	321 +
166 GTO 20	218 X=Y?	270 CLA	322 INT
167 2	219 GTO 25	271 ARCL 24	323 RTN
168 X#Y?	220 E	272 ARCL 36	324*LBL 21
169 GTO 20	221 ST+ 48	273 AVIEW	325 CF 03
170 5	222 ST+ 58	274 FS?C 00	326 CLA
171 XEQ 09	223 GTO 14	275 XEQ 24	327 ARCL 31
172 E	224*LBL 25	276 PSE	328 AVIEW
173 X=Y?	225 5	277 RCL 45	329 TONE 4
174 GTO 21	226 RCL 50	278 40	330 TONE 3
175 GTO 16	227 X=Y?	279 +	331 TONE 2
176*LBL 20	228 GTO 26	280 RCL 40	332 TONE 1
177 CLA	229 E	281 /	333 TONE 0
178 ARCL 20	230 ST+ 50	282 41	334 PSE
179 ARCL IND 49	231 GTO 13	283 +	335 GTO b
180 AVIEW	232*LBL 26	284 STO 57	336*LBL 19
181 PSE	233 E1	285*LBL 28	337 ENTER^
182 RCL 46	234 XEQ 09	286 CLA	338 ABS
183 29	235 E	287 ARCL 30	339 X<>Y
184 X<Y?	236 -	288 " " "	340 17
185 STO 46	237 E1	289 RCL 57	341 +
186 RCL 49	238 *	290 INT	342 X<>Y
187 24	239 STO 35	291 40	343 CLA
188 -	240 RCL 45	292 -	344 ARCL IND Y
189 ST* 46	241 40	293 ARCL X	345 AVIEW
190 RCL 40	242 +	294 " - "	346 PSE
191 RCL 46	243 RCL 40	295 ARCL IND 57	347 FC?C 22
192 *	244 /	296 AVIEW	348 GTO 29
193 ST+ IND 58	245 41	297 54.056	349 X=Y?
194 FIX 3	246 +	298 RCL IND 57	350 GTO 16
195 CF 28	247 STO 57	299 ENTER^	351*LBL 29
196 RCL 46	248*LBL 27	300*LBL 23	352 0
197 CLD	249 RCL 36	301 CLX	353 GTO 18
198*LBL 22	250 RCL IND 57	302 RCL IND Z	354*LBL 24
199 PSE	251 X>Y?	303 X<=Y?	355 CLX
200 TONE 7	252 STO 36	304 XEQ 24	356 E
201 DSE X	253 X>Y?	305 ISG Z	357 ST+ 53
202 GTO 22	254 SF 00	306 GTO 23	358 CLX
203 FIX 0	255 RCL 39	307 ISG 57	359 RCL 53
204 SF 28	256 /	308 GTO 28	360 40
205*LBL b	257 FRC	309 CLA	361 X<=Y?
206 CLA	258 RCL 39	310 ARCL 18	362 STO 53
207 ARCL 30	259 *	311 ARCL 19	363 RDN
208 " " "	260 RCL 35	312 AVIEW	364 TONE 9
209 ARCL 48	261 X=Y?	313 0	365 RTN
210 " -"	262 XEQ 24	314 STO 45	366*LBL 01
211 ARCL 25	263 ISG 57	315 BEEP	367 E
212 ARCL IND 58	264 GTO 27	316 RTN	368 ST+ 46

369 RCL 38	411 GTO 16	453*LBL 07	495 ST+ IND 58
370 ST+ IND 58	412*LBL 04	454 ISG 51	496 ARCL X
371 ARCL X	413 RCL 37	455 GTO 33	497 AVIEW
372 AVIEW	414 STO 62	456 3	498 TONE 6
373 TONE 5	415 5	457 E3/E+	499 TONE 6
374 TONE 5	416 ENTER^	458 STO 51	500 TONE 6
375 TONE 5	417 25	459 SF 03	501 TONE 6
376 TONE 5	418 SF 04	460 RCL 39	502 TONE 6
377 TONE 5	419 XEQ 30	461 ST+ IND 58	503 FC? 05
378 GTO 16	420 RCL 61	462 CLA	504 GTO 34
379*LBL 02	421 5	463 ARCL 21	505 FC? 06
380 E	422 /	464 ARCL 22	506 GTO 34
381 ST+ 46	423 INT	465 AVIEW	507 SF 20
382 3 E2	424 ST+ 46	466 TONE 8	508 26
383 ST+ IND 58	425 GTO 16	467 TONE 8	509 STO 49
384 ARCL X	426*LBL 05	468 GTO 16	510 FC? 07
385 AVIEW	427 8	469*LBL 33	511 GTO 34
386 TONE 6	428 RCL 47	470 RCL 51	512 FC? 08
387 TONE 6	429 X>Y?	471 E	513 GTO 34
388 TONE 6	430 X<>Y	472 -	514 27
389 GTO 16	431 2	473 ARCL X	515 STO 49
390*LBL 03	432 +	474 AVIEW	516 FC? 09
391 RCL 39	433 STO 47	475 TONE 5	517 GTO 34
392 STO 62	434 ENTER^	476 E1	518 FC? 10
393 6	435 ENTER^	477 ST+ IND 58	519 GTO 34
394 ENTER^	436 RCL 40	478 GTO 16	520 TONE 9
395 E1	437 *	479*LBL 08	521 TONE 9
396*LBL 30	438 ST+ IND 58	480 E	522 29
397 XEQ 09	439 ARCL X	481 ST+ 46	523 STO 49
398 X<> 62	440 AVIEW	482 6	524*LBL 34
399 RCL 62	441*LBL 32	483 XEQ 09	525 FC? 20
400 STO 61	442 TONE 7	484 ENTER^	526 GTO 16
401 *	443 DSE Y	485 ENTER^	527 CLA
402 ST+ IND 58	444 GTO 32	486 4	528 ARCL 20
403 ARCL X	445 GTO 16	487 +	529 ARCL IND 49
404 AVIEW	446*LBL 06	488 X<>Y	530 AVIEW
405*LBL 31	447 E1	489 7	531 PSE
406 TONE IND Z	448 ST+ IND 58	490 +	532 GTO 16
407 DSE 62	449 ARCL X	491 CLA	533 END
408 GTO 31	450 AVIEW	492 ARCL IND X	
409 FS?C 04	451 TONE 5	493 SF IND Y	
410 RTN	452 GTO 16	494 5 E2	

Pinball Machine.

HP Co.-Games Pac



This game simulates a pinball machine. Twenty-five cents buys three games, with three balls per game. There are eight types of scoring devices, seven of which, along with appropriate sound effects, advance points immediately upon contact. Standard pinball game features such as specials, extra balls, bonus points, and the possibility of a tilt are included. If you are both skilled and lucky enough to get 50,000 points, you win a free game. So drop in a quarter, start the game and fire a ball, then watch for your chances to flip the flippers.

Scoring Devices

Special

Two roll-over stars are shown at the top of the playing board, either of which scores 10 points. The stars alternate each time one is hit between an ON and OFF state shown by display annunciator flag 0. When the special is on (flag 0 lit in the display) all devices will score 10 times their usual amounts. If the ball goes out of play while the special is lit, the hole bonus scoring is doubled. Flag 0 is reset when a ball goes out of play.

Rollover

The two top roll-overs score 10 points.

Mushrooms

Mushrooms score 100 points each time one is hit. The ball can bounce between them up to ten times, beeping each time.

Kickout

Kickout holes score 50 points.

Spinner

When the spinner gate is hit, it scores 50 points for each spin . Twenty-five spins are possible with a beep sounding with each spin.

Bonus

Bonus advance roll-overs score 10 points when struck and add 1000 hints to the out-hole bonus. Each ball starts with 1000 out-hole bonus points already accumulated.

Flags

There are five flags (numbered I through 5) to knock down. Flags score 10 points, plus an extra ball for the fifth flag. They are reset each time they are all knocked down and before each ball is fired.

Sling Shot

Sling shot bumpers score 10 points.

L Flipper

When " **L FLIPPER**" appears in the display there is approximately 1 second to hit the left flipper (the 1 key) to get the ball back into play, otherwise the ball is lost out the out-hole .

R Flipper

When " **R FLIPPER**" appears in the display there is approximately 1 second to hit the right flipper (the 3 key) to get the ball back into play , otherwise the ball is lost out the out-hole.

Out

When OUT appears in the display the ball is just about to go out of play. If the 2 key is pressed within approximately 1 second there is a 50 percent possibility of the ball going back into play. However, the other 50 percent of the time "tilt" will appear in the display; the ball will be lost and so will the bonus points for that ball. Bonus points, if any, will be added to your score when the ball is lost through the out-hole.

Program listing:

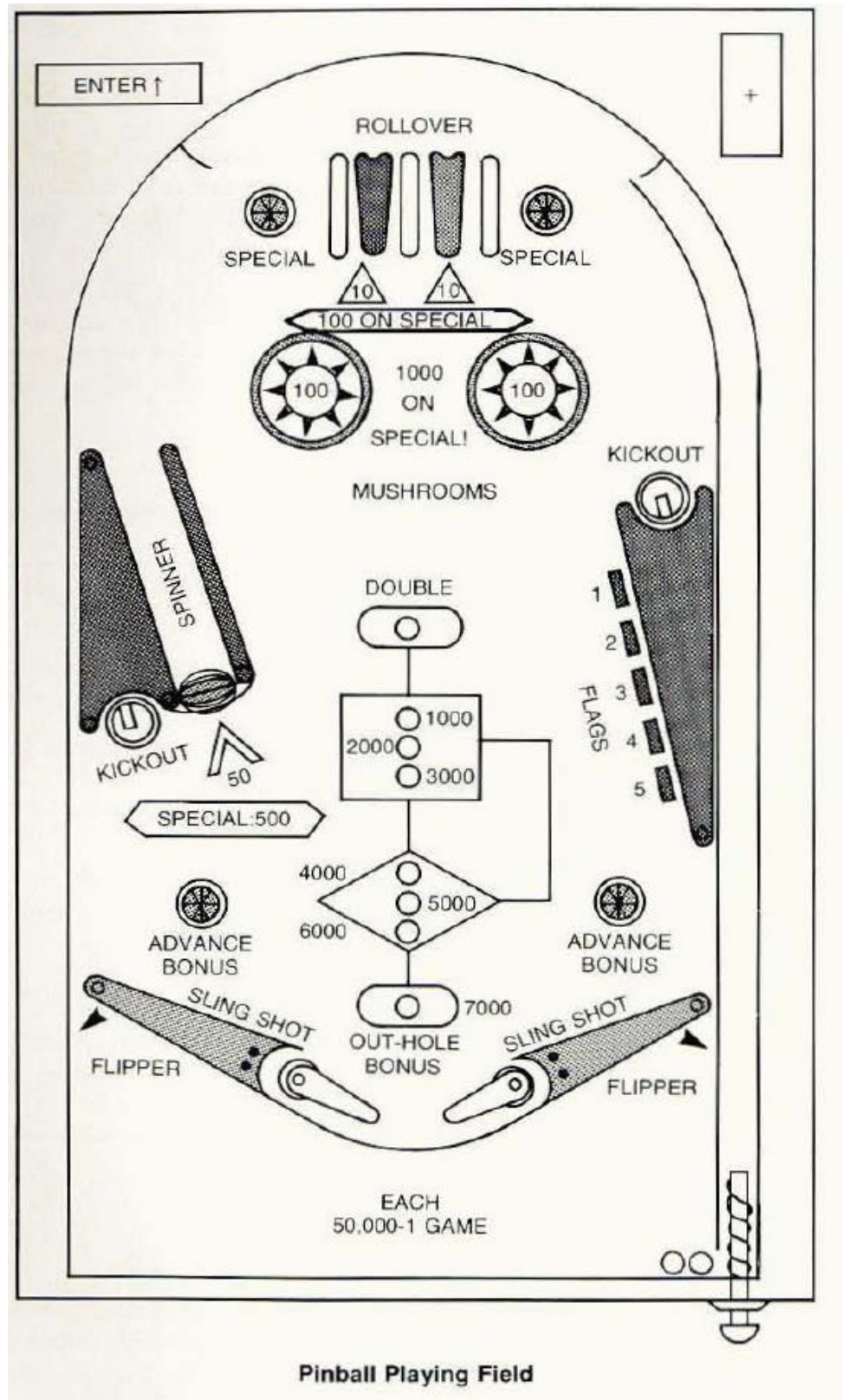
01*LBL "PINBALL"	15 CF 02	29 4
02 7	16 CF 01	30 /
03 XROM "SIZE?"	17 CF 00	31 ARCL X
04 FC?C 25	18 "\$.25=3 GAMES"	32 "" SPENT"
05 PROMPT	19 PROMPT	33 AVIEW
06 SF 27	20*LBL A	34 FIX 0
07 XROM "INI"	21 TONE 6	35 TONE 8
08 CF 21	22 TONE 7	36 TONE 6
09 STO 00	23 TONE 6	37 TONE 9
10 0	24 1	38 3
11 STO 04	25 ST+ 04	39 ST+ 05
12 STO 05	26 "\$"	40*LBL 19
13 FIX 0	27 FIX 2	41 "GAMES: "
14 CF 29	28 RCL 04	42 ARCL 05

43 AVIEW	95 GTO 16	147 GTO 14
44 TONE 8	96 XROM "RNDMW"	148*LBL 06
45 RTN	97 X=0?	149 "BONUS"
46*LBL E	98 GTO 16	150 1 E3
47 TONE 9	99 "TILT"	151 ST+ 01
48 RCL 05	100 AVIEW	152 10
49 X=0?	101 TONE 0	153 GTO 14
50 RTN	102 0	154*LBL 07
51 1	103 STO 01	155 "FLAG "
52 ST- 05	104*LBL 01	156 6
53 4	105 2	157 RCL 03
54 STO 02	106 FS?C 00	158 -
55 0	107 ST* 01	159 ARCL X
56 STO 06	108 RCL 06	160 CLX
57*LBL 15	109 BEEP	161 10
58 CF IND 02	110*LBL 12	162 DSE 03
59 DSE 02	111 VIEW X	163 GTO 14
60 X<0?	112 RCL 01	164 5
61 GTO 17	113 X=0?	165 STO 03
62 1 E3	114 GTO 15	166 CF IND 02
63 STO 01	115 RDN	167 1
64 SF IND 02	116 1 E3	168 ST+ 02
65 5	117 ST- 01	169 SF IND 02
66 STO 03	118 +	170 100
67 "FIRE"	119 STO 06	171*LBL 14
68 TONE 9	120 TONE 6	172 AVIEW
69 PROMPT	121 GTO 12	173 10
70*LBL 16	122*LBL 00	174 X<>Y
71 CLD	123 "MUSHROOMS"	175 FS? 00
72 RDN	124 100	176 *
73 11	125 ENTER^	177 TONE 7
74 XROM "RNDMW"	126 10	178 TONE 6
75 SF 25	127 GTO 00	179 CLD
76 TONE IND X	128*LBL 04	180 PSE
77 GTO IND X	129 "SPINNER"	181 ST+ 06
78*LBL 02	130 50	182 GTO 16
79 "OUT"	131 ENTER^	183*LBL 08
80 GTO 02	132 25	184 "SLING SHOT"
81*LBL 01	133*LBL 00	185 10
82 "L"	134 AVIEW	186 GTO 14
83 GTO 01	135 XROM "RNDMW"	187*LBL 09
84*LBL 03	136 1	188 "SPECIAL"
85 "R"	137 +	189 FC?C 00
86*LBL 01	138 *	190 SF 00
87 "` FLIPPER"	139*LBL 13	191 10
88*LBL 02	140 TONE 9	192 GTO 14
89 AVIEW	141 DSE L	193*LBL 10
90 PSE	142 GTO 13	194 "ROLLOVER"
91 X#Y?	143 GTO 14	195 10
92 GTO 01	144*LBL 05	196 GTO 14
93 2	145 "KICKOUT"	197*LBL 17
94 X#Y?	146 50	198 5 E4

199 RCL 06
 200 -
 201 1
 202 X>Y?

203 ST+ 05
 204 X<=Y?
 205 RTN
 206 BEEP

207 GTO 19
 208 END



Fruit Machine.

[Brian Ward - DataFile V6N6 p39 ; \(September 1986\)](#)

A nice little game for all those who can afford to throw money away (let's face it, if you can afford HP machines, that means you!). The game consists of 3 programs: Fruit, Nudge and Hold. It is in these three parts because it is the most straightforward way to prevent those tempting little extra holds and nudges when you're losing badly! The main program, Fruit, contains many simplifications, mainly to reduce execution time, such as all 3 of the reels are identical and contain only the ten symbols with no duplication. The main thing is that it plays like the real thing except maybe it doesn't rip you off as badly as some pub machines.

To run the program, you're going to need at least one memory module, Extended Functions and a wee bit of Synthetic programming. (These last two are not essential and with a little bit of work you could do without since SP is only used for Tones and the reel symbols and X-Functions to save a few lines and make it more user friendly.)

In operation the top row of keys is used. Uses are as follows:~

[A] -Nudge/Hold reel 1

[B] - N/H reel 2

[C] - N/H reel 3

[D] - Hold Cancel

[E] - SpinSize=018

SP lines are,-

TONE 89 -Fruit lines 80 & 81; Hold lines 09 & 17; Nudge line 10 Tone 73, line 11 Tone 72
Fruit lines 18, 26 & 30 are SP text-they can also be made with XTOA, or substitute your own symbols. Line 30 is purely to create a boxed star in the display.

To run, load the program, execute, and it should ask you for a seed for the random number generator (don't use pi, it is a simple generator designed for speed). Put in a number from 0-1, R/S and it should show you your winnings, followed by displaying the reels and inviting you to SPIN, press the [LN] key and off you go. If Hold is displayed instead of SPIN if you wish to hold, press the key underneath the reel. The appropriate display flag should come on. To cancel, press the LOG key and reenter your choice.

When you have held all you want press [LN], the Spin key. Your chosen reels will be held. If you get a Nudge it will randomly give you between 2 and 10 nudges. To nudge, press the key underneath the reel you wish, once for each nudge. Note if during a nudge a winning combination is displayed, you will win the appropriate prize and any remaining nudges will be forfeit.



01*LBL "FRUIT"

02 17
 03 XROM "INIT"
 04 FIX 2
 05 SF 27
 06 0
 07 STO 16
 08 X<>F
 09 RNG
 10 STO 00
 11 "++"
 12 ASTO 04
 13 "??"
 14 ASTO 05
 15 "dP"
 16 ASTO 06
 17 "ss"
 18 ASTO 07
 19 "[,]"
 20 ASTO 08
 21 "***"
 22 ASTO 09
 23 "%%"
 24 ASTO 10
 25 "\$\$"
 26 ASTO 11
 27 "(,)"
 28 ASTO 12
 29 "<,>"
 30 ASTO 13
 31 ""
 32 ASTO 14
 33 13
 34 STO 01
 35 STO 02
 36 STO 03
 37*LBL 15
 38 TONE 1
 39 TONE 0
 40 CLA
 41 RCL 16
 42 SIGN
 43 X<0?
 44 "-"
 45 ""\$"
 46 LASTX
 47 ABS
 48 ARCL X
 49 AVIEW
 50 PSE
 51 XEQ 17
 52 "" SPIN"

53 PROMPT

54*LBL "SPIN"

55*LBL E
 56 14
 57 FC? 01
 58 STO 01
 59 FC? 02
 60 STO 02
 61 FC? 03
 62 STO 03
 63 XEQ 17
 64 ,1
 65 ST- 16
 66 CF 08
 67 3
 68 E3/E+
 69 STO 15
 70 AVIEW
 71*LBL 00
 72 FS?C IND 15
 73 GTO 01
 74 XEQ 16
 75 E1
 76 *
 77 4
 78 +
 79 INT
 80 STO IND 15
 81*LBL 01
 82 TONE 9
 83 TONE 9
 84 XEQ 17
 85 AVIEW
 86 ISG 15
 87 GTO 00
88*LBL "WIN"
 89 "WIN \$"
 90 RCL 01
 91 RCL 02
 92 X#Y?
 93 GTO 03
 94 RCL 03
 95 X#Y?
 96 GTO 02
 97 BEEP
 98 SF 08
 99 3
 100 -
 101 5
 102 /
 103 ARCL X
 104 AVIEW

105 ST+ 16
 106 GTO 03
 107*LBL 02
 108 FIX 0
 109 TONE 6
 110 TONE 5
 111 SF 08
 112 RDN
 113 3
 114 -
 115 2
 116 /
 117 ,1
 118 +
 119 RND
 120 E1
 121 /
 122 FIX 2
 123 ARCL X
 124 AVIEW
 125 ST+ 16
 126*LBL 03
 127 XEQ 16
 128 ,2
 129 X>Y?
 130 GTO "HOLD"
 131 RDN
 132 ,9
 133 FS?C 08
 134 GTO 15
 135 X>Y?
 136 GTO 15
 137 XEQ 16
 138 GTO "NUDGE"
 139*LBL 16
 140 RCL 00
 141 R-D
 142 FRC
 143 STO 00
 144 RTN
 145*LBL 17
 146 CLA
 147 ARCL IND 01
 148 ARCL IND 02
 149 ARCL IND 03
 150 END

01*LBL "HOLD"
 02 CLA
 03 ARCL IND 01
 04 ARCL IND 02
 05 ARCL IND 03

06 `` HOLD"
 07 PROMPT
 08*LBL A
 09 TONE 9
 10 SF 01
 11 PROMPT
 12*LBL B
 13 TONE 9
 14 SF 02
 15 PROMPT
 16*LBL C
 17 TONE 9
 18 SF 03
 19 PROMPT
 20*LBL D
 21 0
 22 X<>F
 23 PROMPT
 24*LBL E
 25 GTO "SPIN"
 26 END

01*LBL "NUDGE"

02 SF 08
 03 9
 04 *
 05 2
 06 +

07 INT
 08 STO 15
 09*LBL 00
 10 TONE 3
 11 TONE 2
 12 DSE X
 13 GTO 00
 14 CLA
 15 FIX 0
 16 ARCL 15
 17 `` NUDGES"
 18 AVIEW
 19 E
 20 ST+ 15
 21 PSE
 22 FIX 2
 23*LBL 01
 24 CLA
 25 ARCL IND 01
 26 ARCL IND 02
 27 ARCL IND 03
 28 `` NUDGE"
 29 AVIEW
 30 RCL 01
 31 RCL 02
 32 X=Y?
 33 GTO 03
 34 DSE 15

35 STOP
 36 GTO 03
 37*LBL A
 38 E
 39 GTO 02
 40*LBL B
 41 2
 42 GTO 02
 43*LBL C
 44 3
 45*LBL 02
 46 TONE 0
 47 STO 17
 48 E
 49 ST+ IND 17
 50 RCL IND 17
 51 14
 52 X>Y?
 53 GTO 01
 54 4
 55 STO IND 17
 56 GTO 01
 57*LBL 03
 58 BEEP
 59 GTO "WIN"
 60 END

01*LBL "INIT"

02 SIZE?
 03 X<>Y
 04 X>Y?
 05 PSIZE
 06 CLRG
 07 FIX 0
 08 CF 29
 09 "LOADING..."
 10 AVIEW
 11 RASP
 12 RTN
13*LBL "CPRT"
 14 SF 21
 15 ADV
 16 SF 12
 17 9
 18 SKPCOL
 19 49.053

20 STO 13
 21*LBL 00
 22 ACCHR
 23 2
 24 SKPCOL
 25 X<>Y
 26 ISG X
 27 GTO 00
 28 PRBUF
 29 56.06
 30 STO 15
 31*LBL 01
 32 PRBUF
 33 RCL 13
 34 ACCHR
 35*LBL 02
 36 2
 37 SKPCOL
 38 RCL IND 15

39 INT
 40 43
 41 +
 42 RCL IND X
 43 ACSPEC
 44 ISG 15
 45 GTO 02
 46 9
 47 E3/E+
 48 3
 49 +
 50 ST+ 15
 51 ISG 13
 52 GTO 01
 53 ADV
 54 ADV
 55 CF 12
 56 END

Fight (Gun duel)

[Ruys Dirk - DataFile V1N3 P26 ; \(December 1982\)](#)

Something about the game :

You are sitting in a saloon when sudden a drunken cowboy steps on your feet. Because you are a famous gunman you challenge him for a fight. So you go outside and you will draw your gun at him.

When the HP 41 says "DRAW", then you have to guess the place of the enemy. There are 12 different places on the display of the HP 41. (1 to 12) eg. this is the fourth position:



When you have shot the enemy 5 times, he dies... but then his friend takes over. Yet, there is one difference: his friend hides between bushes (= strange ALPHA-characters).

You have 2 pistols, each with 6 bullets.

When you want to change your gun (because it is empty), press [1/X] (XEQ B).

When you fear for your life then press [Σ+] (XEQ A) and you will run away.

To start a new game, press [LN] (XEQ E).

Have fun !!!

RUYS DIRK
Andreas Vesaliuslaan 53
2520 EDEGEM
ANTWERP, BELGIUM

Program listing:

01*LBL "FIGHT"	15 " HEY "	29 CF 02
02 SF 26	16 ARCL 07	30 CF 29
03 "WITH TONES ?"	17 AVIEW	31 FIX 0
04 AON	18 PSE	32 CLST
05 PROMPT	19*LBL E	33 STO 02
06 ASTO Y	20 CLST	34 13
07 "N"	21 STO 03	35 STO 01
08 ASTO X	22 CF 05	36 ""
09 X=Y?	23 "LET'S DRAW"	37 ASTO 05
10 CF 26	24*LBL 20	38 "-----"
11 "YOUR NAME ?"	25 AVIEW	39 ASTO 04
12 PROMPT	26 TONE 3	40*LBL 21
13 ASTO 07	27 TONE 1	41 CF 22
14 AOFF	28 SF 01	42 SF 27

43 11	95 GTO 25	147 AVIEW
44 XEQ 99	96*LBL 24	148 PSE
45 STO 06	97 FS? 02	149*LBL 26
46 CLA	98 R^	150 "P_A_N__G"
47 FC? 05	99 RCL T	151 AVIEW
48 GTO 22	100 RCL 06	152 SF 25
49 "HE'S HIDING"	101 X#Y?	153 SF 30
50 AVIEW	102 GTO 25	154 24
51 CLA	103 " H I T ""	155*LBL 27
52 XEQ 30	104 1	156 TONE 0
53 ARCL 05	105 ST+ 02	157 DSE X
54 11	106 ARCL 02	158 GTO 27
55 RCL 06	107 ""	159 AVIEW
56 -	108 AVIEW	160 5
57 X=0?	109 TONE 9	161 XEQ 99
58 GTO 23	110 PSE	162 3
59 XEQ 30	111 " "	163 X<Y?
60 GTO 23	112 ARCL 04	164 GTO 21
61*LBL 22	113 ASTO X	165 E
62 ""	114 ASHF	166 ST+ 03
63 DSE X	115 ASTO 04	167 "A BULLET IN YOU"
64 GTO 22	116 " "	168 "R "
65 ARCL 05	117 ARCL X	169 XEQ IND 03
66*LBL 23	118 ASHF	170 TONE D
67 TONE ^	119 ASTO 05	171 AVIEW
68 AVIEW	120 RCL 02	172 GTO 21
69 E	121 5	173*LBL 01
70 ST+ 06	122 X#Y?	174 "ASS"
71 FS? 05	123 GTO 21	175 RTN
72 TONE F	124 "HE'S DEAD"	176*LBL 02
73 " DRAW"	125 AVIEW	177 "EYE"
74 AVIEW	126 PSE	178 RTN
75 TONE 1	127 "YOU'RE A KILLER"	179*LBL 03
76 PSE	128 AVIEW	180 "FEET"
77 CF 27	129 PSE	181 RTN
78 FC? 22	130 FC? 05	182*LBL 04
79 GTO 25	131 GTO 00	183 "THERE'S A HOLE "
80 E	132 "THERE'S A REWAR"	184 ""IN YOUR"
81 ST- 01	133 ""D ON YOUR"	185*LBL 28
82 6	134 GTO 28	186 AVIEW
83 FS? 02	135*LBL 00	187 PSE
84 CLX	136 SF 05	188 "" H E A D"
85 RCL 01	137 "NOW HIS FRIEND "	189 AVIEW
86 X>Y?	138 ""IS ANGRY"	190 TONE e
87 GTO 24	139 GTO 20	191 SF 27
88 "PISTOL EMPTY"	140*LBL 25	192 STOP
89 TONE 2	141 FC?C 22	193 GTO "FIGHT"
90 TONE 2	142 GTO 26	194*LBL 30
91 TONE 2	143 "NO "	195 15
92 AVIEW	144 ARCL 07	196 SF 10
93 PSE	145 "", "	197*LBL 99
94 CF 22	146 ARCL 06	198 RNG

199 STO 00	228 "`**"	257 XEQ 99
200 *	229 .	258 3
201 E	230 STO \	259 X>Y?
202 +	231 "`A"	260 "COWARD"
203 INT	232 X<> \	261 X>Y?
204 FC?C 10	233 CLA	262 AVIEW
205 RTN	234 X<> [263 5
206 28	235 ASTO X	264 XEQ 99
207 +	236 RDN	265 3
208 OCT	237 STO [266 X>Y?
209 E3	238 RDN	267 STOP
210 /	239 STO \	268 "HE COMES AFTER "
211 E1	240 ARCL Z	269 "`YOU"
212 +	241 X<>Y	270 AVIEW
213 X<> d	242 DSE X	271 GTO 21
214 FS?C 19	243 GTO 30	272*LBL B
215 SF 20	244 RTN	273 "GUN CHANGED"
216 FS?C 18	245*LBL A	274 AVIEW
217 SF 19	246 CF 27	275 FS?C 01
218 FS?C 17	247 RCL 03	276 2
219 SF 18	248 3	277 FS?C 02
220 FS?C 15	249 "YOUR FEET ..."	278 E
221 SF 17	250 X>Y?	279 SF IND X
222 FS?C 14	251 "RUNNING"	280 CF 22
223 SF 16	252 AVIEW	281 CF 27
224 X<> d	253 PSE	282 GTO 26
225 RCL \	254 X<=Y?	283 END
226 X<>Y	255 GTO 25	
227 X<> [256 5	

Poker (Against your HP-41)

JM Baillard - <http://hp41programs.yolasite.com/poker.php>

To simplify the programs, we use a deck of an infinite number of one-suit cards. In other words, the probability of each card is 1/13, and therefore, the order of the combinations is modified like this:

combinations	probability	ranking
high card	41%	0
1 pair	46%	1
2 pairs	7%	2
3 of a kind	4.6%	3
full house	0.4%	4
straight*	0.3%	5
4 of a kind	0.2%	6
5 of a kind	0.004%	7



The HP-41 is always the dealer.

0- A bet of 1000\$ is placed automatically (line 38)

1.- The calculator deals 5 cards one by one to the player and to itself, the player's cards are gradually shown and the HP-41's cards are displayed as starbursts, then the cards are sorted out in increasing order, displayed again, and the program stops.

2.- At this step, key in a stake (of about \$1000 or \$2000 for instance) and press R/S (if the player simply presses R/S the HP-41 wins)

3.- Then, the HP-41 can:

- a) fold (the player wins and the new bank of the player is displayed)
- b) match the player's bet.
- c) make a higher bid (which is displayed as " + \$)

In this case, the player can, in his turn:

- a) fold (press: 0, R/S)
- b) match the HP-41's bet (press: ENTER, R/S)
- c) make a higher bid (key in a number greater than the HP-41's bid and R/S) .. etc..

4.- At the end of the stakes, the player's cards are displayed again and the program stops.

- To exchange the cards number 1, 2, 5 (if your hand is 45QQA for example)
press: 125, R/S.

- If you have a complete hand, simply press R/S.
5. The HP-41 displays the remaining cards of the player and as many starbursts as its own remaining cards.
 6. About 12 seconds later, the new hand of the player is displayed (followed by / and 5 starbursts) and the program stops.
 7. The second round of betting takes place here just like in steps 2 and 3 (To fold immediately, simply press R/S).
 8. If the final bet or raise is not called, the new bank of the player is displayed, but the calculator's cards remain unknown. Otherwise, the HP-41 displayed its own cards one by one and then it displays the new bank of the player.
 9. Press R/S to continue the game.

Remarks:

- 1-If the player exchanges 3; 2; 1; 0 cards, the HP-41 will fold unless it has (at least) one pair; three of a kind; two pairs; a full house (respectively). (lines 183 to 191)
- 2-This strategy is very simple, so don't play too aggressively against your HP-41 ... (but remember that the HP-41 can bluff in the second round of betting (lines 177 to 182)).
- 3-Don't disturb rashly the stack during the raises.
- 4-Digit entry lines are very slow. Therefore, execution time can be saved by storing these numbers in data registers: for instance replace line 211 by RCL 41 and key in 1.015008 STO 41 just after line 27 ... etc ...

Data Registers:

R00 thru R40 are used. R00 and R12 must be initialized before executing "POKER"

R00 = random numbers

R12 = the bank of the player (the cash)

R01 thru R05 are the HP-41's cards R06 thru R11 are used for temporary data storage

R15 thru R19 are the player's cards R20 thru R22 -----

R10 is also used for the HP-41's raises.

R08 and R22 are also used to store the values of both hands.

R21 is also used to store the player's hand.

R23 = /ααααα

R24 = /αααα

R25 = /ααα (α symbolizes the starburst, the hidden cards of the HP-41)

R26 = /αα

R27 = /α

R28 = the "2s"

R29 = the "3s"

.....
R39 = the "Ks"

R40 = the "As"

Flags: F06 ; Subroutines: none.

Analysis of the program:

Lines 02 to 27 initialize the registers containing the cards (2 , 3 , ... , 9 , T , J , Q , K , A) and the hidden cards of the calculator (displayed as starbursts). They are executed only once.

Lines 28 to 59 place the initial bet (\$1000, line 38) and deal the cards one by one.

Line 60 (XEQ 09) sorts the cards, calculates the value of the HP-41's hand and stores in R10 the maximum amount of money that the HP-41 will accept.

Lines 61 to 68 modify this sum when the calculator has 4 cards in sequence (like 6789K) which can lead to a straight. In such a case, the HP-41 will raise the bets just as if it had two pairs.

Lines 70 to 94 concern the betting:

if the player keys in 0 R/S (or simply R/S at the first time) the HP-41 wins the pot.

if the player keys in ENTER^ R/S the raises are called.

if the player keys in a bet and R/S there are 3 possibilities:

- a) If the total amount of the raises is higher than the number in R10 the player wins the pot.
- b) If the total amount of the raises is smaller than R10 but greater than $0.4 \times R10$, the HP-41 is in.
- c) If the total amount of the raises is smaller than $0.4 \times R10$, the HP-41 makes a higher raise ... etc ... (This raise is chosen randomly but the total amount of the previous bets is also taken into account)

Lines 104 to 112 are the showdown: the calculator displays its cards one by one.

Lines 113 to 137 display the player's bank.

Line 138 returns to line 28

Lines 139 to 175 display the remaining cards of the player (and as many starbursts as the remaining cards of the HP-41) after discarding.

Line 176 executes the LBL 09 (see below)

Lines 177 to 182 are the calculator's bluff (only in the second round of betting): The number in R10 is divided by a random real number r ($0 < r < 1$).

Lines 183 to 191 are the HP-41's strategy (as explained in the 1st remark above)

Lines 192 to 193 return to line 70 (the betting)

Lines 194 to 209 calculate and store in R10 the maximum amount of money that the HP-41 will accept. This number suits to bets of a few thousand (or ten thousand) dollars.

Lines 210 to 222 replace the discarded cards by new ones.

Lines 223 to 263 sort out the cards in increasing order (in R01 to R05)

Lines 269 to 288 calculate the value of the hand in R08. The result is also used to obtain the number in R10 which is proportional to the square of R08. The number 37 (line 271) can be changed but must not be too small (not smaller than 16). Otherwise, the order of the different hands could become wrong.

Lines 289 to 300 deal the cards. The random number generator used by this LBL 08 is quite simple: R-D, FRC. It's perhaps not a perfect one but it's good enough for a game.

Lines 301 to 454 concern the evaluation of the hand, the cards to be exchanged ...etc...

The details are somewhat complex, but it works well!

For instance, if the HP-41's hand is 3377K we have:

R01 = R02 = 29 (the "3s") ; R03 = R04 = 33 (the "7s") ; R05 = 39 (the "K").
 R06 = 5 because the HP-41 will exchange its 5th card (the King) after the first round of betting.
 R07 = 2 = the ranking of the hand: two pairs.
 R08 = 33 ; R09 = 29 ; R10 = 39 ; R11 = 29

Then, the value of the hand is obtained by:

$$37^5 \cdot R07 + 37^4 \cdot R08 + 37^3 \cdot R09 + 37^2 \cdot R10 + 37 \cdot R11 + R01 = 202058657$$

and stored in R08 (line 287). In this example, R11 and R01 could be replaced by zero in the above formula, but it's not necessary. This number characterizes the hand and allows to compare the HP-41's hand with the player's hand.

$(R08/E6)^2 = 40827.7$ \$ is then stored in register R10 (line 210).

Program listing:

01*LBL "POKER"	23 DSE X	45 GETKEYX	67*LBL 04
02 22	24 GTO 01	46 X=0?	68 RCL IND X
03 XROM "INIT"	25 "BET\$=?"	47 GTO 10	69 ARCL IND X
04 RNG	26 PROMPT	48 LASTX	70 COS
05 STO 00	27 STO 22	49 -	71 AVIEW
06 CLX	28*LBL 16	50 CLRGX	72 X<>Y
07 STO 06	29 " "	51 GTO 03	73 DSE X
08 "A"	30 ASTO 08	52*LBL 10	74 GTO 04
09 ASTO 21	31 5	53 " "	75 RCL 07
10 "K"	32*LBL 02	54 5	76 X#0?
11 ASTO 20	33 CLA	55*LBL 13	77 GTO 09
12 "Q"	34 ARCL 08	56 RCL IND X	78 17
13 ASTO 19	35 ENTER^	57 X#0?	79 RCL 08
14 "J"	36 XEQ 14	58 ARCL IND X	80 X>Y?
15 ASTO 18	37 ARCL IND X	59 X<>Y	81 ISG 07
16 16	38 ASTO 08	60 DSE X	82 INT
17 "T"	39 AVIEW	61 GTO 13	83 14
18 ASTO 17	40 X<>Y	62 AVIEW	84 X>Y?
19 9	41 DSE X	63 5	85 DSE 07
20*LBL 01	42 GTO 02	64 XEQ 05	86*LBL 09
21 STO IND Y	43*LBL 03	65 " "	87 RCL 22
22 DSE Y	44 54	66 5	88 RCL 07

89 *	134 X<> 05	179 X>Y?	224 RDN
90 ST+ 06	135 X>Y?	180 RTN	225*LBL 09
91 RCL 06	136 X<>Y	181 RCL 02	226 X<> Z
92 " "	137 R^	182 LASTX	227 STO 08
93 X>0?	138 X>Y?	183 -	228 CLX
94 "`+"	139 X<>Y	184 PI	229 STO 07
95 ARCL 06	140 RDN	185 X<Y?	230 RTN
96 "`\$"	141 X>Y?	186 GTO 10	231*LBL 10
97 CF 22	142 X<>Y	187 R^	232 RCL 01
98 PROMPT	143 RDN	188 12	233 X=Y?
99 FS?C 22	144 X<Y?	189 X=Y?	234 GTO 09
100 STO 22	145 X<>Y	190 RTN	235 GTO 07
101 GTO 16	146 STO 01	191*LBL 10	236*LBL 06
102*LBL 14	147 STO 08	192 5	237 RCL 02
103 CLX	148 RDN	193 CHS	238 X=Y?
104 RCL 00	149 X>Y?	194 STO 07	239 GTO 10
105 R-D	150 X<>Y	195 RTN	240 RCL 01
106 13	151 STO 04	196*LBL 11	241 X=Y?
107 MOD	152 RDN	197 STO 08	242 RTN
108 STO 00	153 X>Y?	198 CLX	243*LBL 07
109 9	154 X<>Y	199 STO 07	244 3
110 +	155 STO 03	200 RTN	245 STO 07
111 INT	156 X<>Y	201*LBL 06	246 RTN
112 STO IND Y	157 STO 02	202 RCL 02	247*LBL 08
113 RTN	158 5	203 X=Y?	248 RCL 01
114*LBL 05	159 STO 07	204 GTO 10	249 X=Y?
115 RCL IND X	160 RCL 05	205 RCL 01	250 RTN
116 X=0?	161 RCL 04	206 X=Y?	251*LBL 11
117 XEQ 14	162 X=Y?	207 GTO 11	252 2
118 X<>Y	163 GTO 08	208 GTO 09	253 STO 07
119 DSE X	164 RCL 03	209*LBL 07	254 RTN
120 GTO 05	165 X=Y?	210 RCL 01	255*LBL 10
121 RCL 01	166 GTO 06	211 X=Y?	256 21
122 RCL 02	167 RCL 02	212 GTO 07	257 STO 07
123 X>Y?	168 X=Y?	213 GTO 09	258 CLX
124 X<>Y	169 GTO 07	214*LBL 08	259 RCL 01
125 RCL 03	170 RCL 01	215 RCL 03	260 X=Y?
126 X>Y?	171 X=Y?	216 X=Y?	261 RTN
127 X<>Y	172 GTO 11	217 GTO 06	262*LBL 09
128 RCL 04	173 RCL 05	218 RCL 02	263 13
129 X<Y?	174 -	219 X=Y?	264 STO 07
130 X<>Y	175 8	220 GTO 08	265 END
131 X<> 05	176 STO 07	221 RCL 01	
132 X>Y?	177 CLX	222 X=Y?	
133 X<>Y	178 5	223 GTO 11	

American Roulette

JM Baillard - <http://hp41programs.yolasite.com/roulette.php>

Overview

With this program, you play Roulette against your HP-41

Several wagers are proposed and you bet (or not) after each PROMPT.
Then the wheel spins, the HP-41 displays the result and finally, your new cash is displayed.

The board looks like this:

HP-41		Casino	0	00
1to18	1st12	1	2	3
		4	5	6
EVEN		7	8	9
		10	11	12
RED	2nd12	13	14	15
		16	17	18
BLACK		19	20	21
		22	23	24
ODD	3rd12	25	26	27
		28	29	30
19to36		31	32	33
		34	35	36
HP-41	Casino	2to1	2to1	2to1

The different wagers are as follows:

Other wagers do exist but they are omitted in this version of the game.

RED/BLACK paid 1 to 1

ODD/EVEN 0/00 are neither one paid 1 to 1

LOW/HIGH

low = numbers between 1 and 18 paid 1 to 1

high = numbers between 19 and 36 paid 1 to 1

COLUMN 1-2-3

Column1 = 1 4 7 31 34 paid 2 to 1 as written on the board

Column2 = 2 5 8 32 35 ---- 2 to 1 as written on the board

Column3 = 3 6 9 33 36 ---- 2 to 1 as written on the board

DOZEN 1-2-3

1st Dozen = 1 to 12	paid 2 to 1
2nd Dozen = 13 to 24	---- 2 to 1
3rd Dozen = 25 to 36	---- 2 to 1

STREET 1 to 12

Street1 = 1 2 3	paid 11 to 1
Street2 = 4 5 6	---- 11 to 1
.....
Street12 = 34 35 36	---- 11 to 1

STRAIGHT UP You bet on a single number paid 35 to 1

Example:

1 STO 00, 5000 STO 01 (if you start with \$5000)
XEQ "ROULETT"

Suppose you want to bet 200\$ on BLACK , 200\$ on ODD , 200\$ on column 3 , 200\$ on street 8 and 100\$ on number 16

RED/BLACK?	key in 200 ENTER^ 2 R/S
ODD/EVEN?	200 ENTER^ 1 R/S
	or simply 1, R/S because the previous bet 200\$ is still in X-reg.)
LOW/HIGH?	simply press R/S
COLUMN 1-2-3?	3, R/S - but for a bet of \$100, key in 100, ENTER^, 3, R/S
DOZEN 1-2-3?	R/S
STREET 1-12?	8 R/S
STRAIGHT UP?	100 ENTER^ 16 R/S
STRAIGHT UP?	R/S

The HP-41 displays the spinning wheel, then **11 BLACK ODD** and finally your new cash:
C=4900\$

You have bet \$900 and you receive $200 \times 2 + 200 \times 2 = 800\$$
therefore, $C = 5000\$ - 900\$ + 800\$ = 4900\$$

Press R/S to continue the game ...

Notes:

To wager on 00, you can place your bet and then:

ENTER^ 0 ENTER^ R/S
or ENTER^ 37 R/S

The first 6 wagers are proposed only once, but if you want to bet several times on these wagers simply add: GTO 06 GTO 05 GTO 04 GTO 03 GTO 02 GTO 01
after lines 64 54 44 34 24 14 respectively

Conversey, delete line 81 if you never bet twice on a single number.

Program listing:

01*LBL "RLTTE"	48 ST- 03	95 GTO 09
02 66	49 STO IND Y	96 37
03 XROM "INIT"	50*LBL 05	97 RCL 00
04 RNG	51 "DOZEN? 1-2-3"	98 R-D
05 STO 00	52 PROMPT	99 FRC
06 "BANK\$=?"	53 FC?C 22	100 STO 00
07 PROMPT	54 GTO 06	101 38
08 STO 01	55 12	102 *
09*LBL 16	56 +	103 INT
10 CF 22	57 X<>Y	104 STO 02
11 "RED/BLACK?"	58 ST- 03	105 " "
12 PROMPT	59 STO IND Y	106 X=Y?
13 FC?C 22	60*LBL 06	107 ">"00"
14 GTO 02	61 "STREET? 1-12"	108 X#Y?
15 3	62 PROMPT	109 ARCL X
16 +	63 FC?C 22	110 X#0?
17 X<>Y	64 GTO 07	111 X=Y?
18 ST- 03	65 15	112 GTO 10
19 STO IND Y	66 +	113 18
20*LBL 02	67 X<>Y	114 MOD
21 "ODD/EVEN?"	68 ST- 03	115 X=0?
22 PROMPT	69 STO IND Y	116 X<> L
23 FC?C 22	70*LBL 07	117 E1
24 GTO 03	71 "STRAIGHT UP?"	118 X<Y?
25 5	72 PROMPT	119 ISG Y
26 +	73 FC?C 22	120 CLX
27 X<>Y	74 GTO 08	121 CLX
28 ST- 03	75 X=0?	122 2
29 STO IND Y	76 X#Y?	123 MOD
30*LBL 03	77 GTO 07	124 X=0?
31 "LOW/HIGH?"	78 X<> Z	125 ">" BLACK"
32 PROMPT	79 37	126 X#0?
33 FC?C 22	80*LBL 07	127 ">" RED"
34 GTO 04	81 28	128 5
35 7	82 +	129 -
36 +	83 X<>Y	130 RCL IND X
37 X<>Y	84 ST- 03	131 ST+ X
38 ST- 03	85 STO IND Y	132 ST+ 03
39 STO IND Y	86 GTO 07	133 7
40*LBL 04	87*LBL 08	134 RCL 02
41 "COL? 1-2-3"	88 "331642335142"	135 2
42 PROMPT	89 SF 25	136 MOD
43 FC?C 22	90 AVIEW	137 X=0?
44 GTO 05	91 SF 99	138 "` EVEN"
45 9	92 41	139 X#0?
46 +	93*LBL 09	140 ">" ODD"
47 X<>Y	94 DSE X	141 -

142 RCL IND X	165 ST+ 03	188*LBL 10
143 ST+ X	166 RCL 02	189 AVIEW
144 ST+ 03	167 E	190 COS
145 RCL 02	168 -	191 COS
146 E	169 12	192 COS
147 -	170 /	193 COS
148 18	171 13	194 COS
149 /	172 +	195 RCL 02
150 8	173 RCL IND X	196 28
151 +	174 3	197 +
152 RCL IND X	175 *	198 RCL IND X
153 ST+ X	176 ST+ 03	199 36
154 ST+ 03	177 RCL 02	200 *
155 RCL 02	178 E	201 ST+ 03
156 3	179 -	202 RCL 03
157 MOD	180 3	203 ST+ 01
158 X=0?	181 /	204 " C="
159 X<> L	182 16	205 RCL 01
160 9	183 +	206 ARCL X
161 +	184 RCL IND X	207 "\$"
162 RCL IND X	185 12	208 PROMPT
163 3	186 *	209 GTO 16
164 *	187 ST+ 03	210 END

Black Jack for the HP-41

JM Baillard - <http://hp41programs.yolasite.com/blackjack.php>

This program allows you to play Blackjack against your HP-41, with the calculator always playing the bank.

Instructions:

Place your bet in X-register and XEQ "BLJ"

The HP-41 deals 2 cards for you and 1 for itself.

If you have 21 points with these 2 cards, you win 1.5 times your bet (unless the HP-41 has also 21 points in 2 cards), otherwise:

- Press the $\Sigma+$ key to double your bet (line 74): in this case, you receive one card only.
- Press ENTER^ to hit (line 65): you can receive as many cards as you want (provided your points do not exceed 21 - otherwise, you lose)
- Press any other key to stand.

If you have no Blackjack and if your total does not exceed 21, the HP-41 deals cards for itself until its total exceeds 16. Then, the 2 hands are compared and your new cash is displayed.

Place another bet and R/S to continue the game or simply press R/S without any digit entry to place the same bet.

Card Designation and Value:

A = ace = 1 or 11 points

2 = 2 , 3 = 3 , , 9 = 9 points

T = J = Q = K = 10 points

Therefore the possible Blackjacks are: AT AJ AQ AK

Notes:

- "Split" and "Insurance" are not covered by this program.
- A deck of an infinite number of one-suit cards is used. (In other words, the probability of each card = 1/13)
- Note that the GETKEY function of the X-functions module is used. This saves you from pressing R/S and makes the program properly react to the allowable keys.

Data Registers:

- R00 = random numbers
- R01 = your cash ; R02 thru R09: scratch

Example: 7, STO 00, CLX, STO 01

Place you bet, for instance \$12000 in X-register

12000, XEQ "BLJ" the HP-41 displays:

```
" 6 / "
```

```
" 6 / 5 "
```

```
" 62 / 5 "
```

Press ENTER^ to hit, you get: " 628 / 5 "

You have 19 points, so you stand: press any key except ENTER^ and SIGMA+

```
" 628 / 5 K " the HP-41 continues until its points exceed 16
```

```
" 628 / 5 K K " Busted ! HP-41 exceeds 21: you win
```

```
" + 12000 $ "
```

Simply press R/S to continue with the same bet:

```
" 5 / "
```

```
" 5 / 3 "
```

```
" 56 / 3 "
```

Press the S+ key to double your bet:

```
" 568 / 3 "
```

```
" 568 / 3 J "
```

```
" 568 / 3 J A "
```

```
" 568 / 3 J A 9 " Busted again: you win 24000 $ and your cash is:
```

```
" + 36000 $ "
```

Remarks:

Do not store PI in register R00, the "random" numbers wouldn't be randomized at all...

The display is very minimalist! You could add some pizzad to it, for example:

```
add " TIE-PUSH" X=Y? AVIEW after line 147
```

```
add " BLACKJACK" AVIEW after line 96 and after line 27
```

One register only is used to store each hand so the HP-41 will not display all the cards if you (or the HP-41) have more than 6 cards. To overcome this limitation:

```
add ARCL 11 after line 121
```

```
add ARCL 10 after line 119
```

```
replace lines 113 to 115 by ARCL 05 ARCL 11 ARCL Z ASTO 05 ASHF ASTO 11
```

```
replace lines 103 to 105 by ARCL 04 ARCL 10 ARCL Z ASTO 04 ASHF ASTO 10
```

```
add ASTO 10 ASTO 11 after line 09
```

Program listing:

01 LBL "BLJ"	42 ARCL X	83 X>Y?	124 RTN
02 FS?C 22	43 LASTX	84 CLX	125 LBL 08
03 STO 09	44 X=Y?	85 -	126 XEQ 06
04 21	45 "K"	86 X<0?	127 LBL 09
05 STO 02	46 DSE X	87 21	128 RCL 08
06 STO 03	47 X=Y?	88 STO 02	129 *
07 CLA	48 "Q"	89 LBL 04	130 X>Y?
08 ASTO 04	49 DSE X	90 XEQ 06	131 CLX
09 ASTO 05	50 X=Y?	91 RCL 08	132 - 133 21
10 2	51 "J"	92 *	134 RCL 02
11 STO 06	52 DSE X	93 X#Y?	135 X#Y?
12 CLX	53 X=Y?	94 GTO 09	136 X<0?
13 STO 07	54 "T"	95 CHS	137 GTO 10
14 STO 08	55 X<Y?	96 STO 03	138 4
15 FIX 0	56 X<>Y	97 GTO 10	139 R^
16 CF 29	57 SIGN	98 LBL 05	140 X>Y?
17 XEQ 05	58 X=Y?	99 XEQ 01	141 GTO 08
18 XEQ 06	59 "A"	100 ST- 02	142 X<0?
19 XEQ 05	60 X<>Y	101 X=Y?	143 R^
20 RCL 07	61 ASTO Z	102 STO 07	144 STO 03
21 *	62 CLA	103 ARCL 04	145 LBL 10
22 X#Y?	63 RTN	104 ARCL Z	146 RCL 03
23 GTO 02	64 LBL 02	105 ASTO 04	147 RCL 02
24 CHS	65 41	106 RCL 02	148 -
25 STO 02	66 GETKEY	107 GTO 07	149 X#0?
26 1.5	67 X#Y?	108 LBL 06	150 SIGN
27 STO 06	68 GTO 03	109 XEQ 01	151 RCL 06
28 GTO 04	69 XEQ 05	110 ST- 03	152 *
29 LBL 01	70 CLX	111 X=Y?	153 RCL 09
30 RCL 00	71 X<Y?	112 STO 08	154 *
31 R-D	72 GTO 02	113 ARCL 05	155 ST+ 01
32 FRC	73 LBL 03	114 ARCL Z	156 RCL 01
33 STO 00	74 11	115 ASTO 05	157 " "
34 9	75 X#Y?	116 RCL 03	158 X>0?
35 +	76 DSE 06	117 LBL 07	159 "~+"
36 10^X	77 X=Y?	118 " "	160 ARCL 01
37 13	78 XEQ 05	119 ARCL 04	161 "~ \$"
38 MOD	79 RCL 02	120 "~/"	162 FIX 4
39 X=0?	80 10	121 ARCL 05	163 SF 29
40 LASTX	81 RCL 07	122 AVIEW	164 AVIEW
41 CLA	82 *	123 10	165 END

Blackjack Card Counter

Richard Baker - PPCCJ V7N7 p6 (September 1980)

Steps:

- 1: Load program SIZE 045
- 2:- Must do: Clear all registers CLRG
- 3:- Set histogram parameters [] [A]
- 4:- Enter card values as each is dealt: [A]
 Aces are 1, Jacks 11, Queens 12, Kings 13
- 5:- View card count at any time [B]
- 6:- Compute remaining probabilities for each card value [C]
- 7:- View remaining probabilities [D]
 forfor each card value at any time after Step 6
- 8:- Print out two histograms [E]
 - a. card count showing number of each card that has been played
 - b. probability for each card remaining
- 9:- For new deck, return to step 2.

COMMENTS:

- Program works for a one deck deal only.
- View functions and histogram printouts do not show the card value, just the count and the two decimal probabilities for that value; i.e., the counts for the Aces are first, kings, last. Each histogram has 13 lines.
- View function requires R/S after each pause. DO NOT press R/S after "24" shows on label B or "44" on label D
- Probability figures given show number of chances in the remainder of the deal that any particular card will be the next card to be played; eg: 1 in x.
- Since blackjack rules do not require losing players to show their down cards, counts and probabilities will necessarily be off to this extent.
- When entering card counts in label A, you must allow time for a "1" to show in the display.
- Viewing and histogram printout can take place at any time during the deal.

Program listing:

01 LBL "BLAKJAK"

02 LBL a
03 .01
04 STO 00
05 4
06 STO 01
07 52
08 STO 07
09 125
10 STO 02
11 CLX
12 0
13 ENTER
14 65
15 BLDSPEC
16 65
17 BLDSPEC
18 65
19 BLDSPEC
20 65
21 BLDSPEC
22 65
23 BLDSPEC
24 65
25 BLDSPEC
26 127
27 BLDSPEC
28 STO 03
29 CLX
30 RTN
31 LBL A
32 ENTER
33 10
34 +
35 STO 05
36 1
37 ST+ IND 05
38 ST- 07
39 RTN
40 LBL B
41 11.024
42 STO 05

43 LBL 01
44 CLX
45 VIEW IND 05
46 PSE
47 ISG 05
48 RCL 05
49 INT
50 24
51 X#Y?
52 GTO 01
53 RTN
54 LBL C
55 11.023
56 STO 05
57 31.043
58 STO 06
59 LBL 02
60 RCL IND 05
61 X=0?
62 XEQ 03
63 RCL IND 05
64 1
65 X=Y?
66 XEQ 04
67 RCL IND 05
68 2
69 X=Y?
70 XEQ 05
71 RCL IND 05
72 3
73 X=Y?
74 XEQ 06
75 RCL IND 05
76 4
77 X=Y?
78 XEQ 07
79 ISG 05
80 ISG 06
81 RCL 05
82 INT
83 24
84 X=Y?

85 STOP
86 GTO 02
87 LBL 03
88 RCL 07
89 4
90 /
91 STO IND 06
92 RTN
93 LBL 04
94 RCL 07
95 3
96 /
97 STO IND 06
98 RTN
99 LBL 05
100 RCL 07
101 2
102 /
103 STO IND 06
104 RTN
105 LBL 06
106 RCL 07
107 1
108 /
109 STO IND 06
110 RTN
111 LBL 07
112 0
113 STO IND 06
114 RTN
115 LBL D
116 31.044
117 STO 06
118 LBL 08
119 CLX
120 VIEW IND 06
121 PSE
122 ISG 06
123 RCL 06
124 INT
125 44
126 X#Y?

127 GTO 08
128 RTN
129 LBL E
130 11.024
131 STO 05
132 VIEW 01
133 LBL 09
134 RCL IND 05
135 ACX
136 REGPLOT
137 ISG 05
138 RCL 05
139 INT
140 24
141 X=Y?
142 GTO 10
143 GTO 09
144 LBL 10
145 ADV
146 31.044
147 STO 06
148 RCL 07
149 STO 01
150 VIEW 01
151 LBL 11
152 RCL IND 06
153 ACX
154 REGPLOT
155 ISG 06
156 RCL 06
157 INT
158 44
159 X=Y?
160 GTO 12
161 GTO 11
162 LBL 12
163 4
164 STO 01
165 RTN
166 END

BlackJack

Whodunit – Swap Disks

Another undocumented but very nicely implementation – so much so that this one made it on the 'FUN_STUFF' module.

Program listing:

01*LBL "BJ"	41 GTO 03	81 X<=0?	121*LBL 01
02 26	42 CLX	82 PROMPT	122 21
03 PSIZE	43 E	83 STO 10	123 RCL 07
04 XEQ 11	44 X=Y?	84*LBL 01	124 X=Y?
05 CLX	45 GTO 04	85 RCL 10	125 GTO 01
06 STO 12	46 CLA	86 STO 11	126 FS? 05
07*LBL 12	47 ARCL Y	87 CF 29	127 GTO C
08 13,025	48 GTO 05	88 CLA	128 XEQ 05
09 CLD	49*LBL 03	89 ASTO 05	129 RTN
10 9	50 STO 02	90 ASTO 06	130*LBL 14
11*LBL 13	51 "T"	91 CLX	131 FS?C 10
12 STO IND Y	52 E	92 STO 07	132 GTO C
13 ISG Y	53 +	93 STO 08	133 RTN
14 GTO 13	54 X=Y?	94 XEQ 04	134*LBL 01
15 92	55 "J"	95 RCL 03	135 "DLR BJ"
16 STO 01	56 E	96 STO 04	136 AVIEW
17 CLX	57 +	97 SF 09	137 BEEP
18 RTN	58 X=Y?	98 XEQ 07	138 SF 05
19*LBL 00	59 "Q"	99 AVIEW	139*LBL C
20 FIX 0	60 E	100 XEQ 04	140 FS? 29
21 RNG	61 +	101 RCL 03	141 GTO 05
22 13	62 X=Y?	102 RCL 04	142 11
23 *	63 "K"	103 X=Y?	143 RCL 08
24 E	64 GTO 05	104 SF 20	144 X>Y?
25 +	65*LBL 04	105 SF 09	145 CF 06
26 INT	66 "A"	106 XEQ 07	146 E1
27 STO 02	67 SF 05	107 ARCL 03	147 FS?C 06
28 12	68*LBL 05	108 AVIEW	148 ST+ 08
29 +	69 RCL 02	109 FC? 06	149 FS?C 09
30 DSE IND X	70 ASTO 03	110 GTO 01	150 GTO 02
31 GTO 01	71 RTN	111 RCL 08	151 GTO 08
32 GTO 00	72*LBL A	112 11	152*LBL E
33*LBL 01	73 FC? 29	113 X#Y?	153 E1
34 DSE 01	74 GTO C	114 GTO 01	154 RCL 08
35 GTO 02	75 FC?C 22	115 "PLYR BJ"	155 X#Y?
36 XEQ 12	76 GTO 01	116 AVIEW	156 11
37*LBL 02	77 "BET?"	117 BEEP	157 X=Y?
38 RCL 02	78 E2	118 1,5	158 FS? 07
39 E1	79 X<>Y	119 ST* 11	159 GTO 05
40 X<=Y?	80 X<=Y?	120 SF 05	160 2

161 ST* 11	197*LBL 02	233 ST+ 07	269 21
162 SF 10	198 RCL 04	234 CLA	270 RCL 07
163*LBL B	199 STO 06	235 ARCL 05	271 X>Y?
164 FS? 29	200 RCL 08	236 ARCL 03	272 XEQ 06
165 GTO 05	201 X<> 09	237 ASTO 05	273 RCL 08
166 SF 07	202 STO 08	238*LBL 08	274 X=Y?
167 XEQ 04	203 XEQ 05	239 "DLR "	275 SF 10
168 21	204 GTO B	240 FS? 09	276 "PUSH"
169 RCL 08	205*LBL 04	241 ""	277 FS? 10
170 X<=Y?	206 XEQ 00	242 FS?C 09	278 AVIEW
171 GTO 14	207 FS?C 05	243 RTN	279 X<Y?
172 CHS	208 SF 06	244 ARCL 05	280 SF 05
173 STO 08	209 ST+ 08	245 AVIEW	281 RCL 11
174 XEQ 06	210 CLA	246*LBL 09	282 FS?C 05
175 GTO C	211 ARCL 06	247 RCL 08	283 CHS
176*LBL D	212 ARCL 03	248 X>0?	284 FC?C 10
177 FS?C 20	213 ASTO 06	249 FS?C 05	285 ST+ 12
178 FS? 07	214*LBL 05	250 GTO 10	286 FIX 2
179 GTO 05	215 "PLYR "	251 FS? 23	287 "BANK \$"
180 SF 08	216 ARCL 06	252 SF 06	288 ARCL 12
181 SF 09	217 AVIEW	253 17	289 AVIEW
182 RCL 04	218 RTN	254 RCL 07	290 RCL 08
183 STO 06	219*LBL 06	255 X#Y?	291 X<> 09
184 2	220 PSE	256 CF 06	292 STO 08
185 ST/ 08	221 "BUST"	257 FC?C 06	293 FS?C 08
186 RCL 08	222 AVIEW	258 X<Y?	294 GTO 08
187 STO 09	223 CLX	259 GTO 07	295 CLST
188 XEQ 05	224 RTN	260 22	296*LBL 11
189 E	225*LBL 07	261 X>Y?	297 "<"
190 X#Y?	226 XEQ 00	262 GTO 10	298 ASTO d
191 GTO B	227 FC?C 05	263 E1	299 CF 03
192 XEQ B	228 GTO 07	264 ST- 07	300 CLA
193 CF 09	229 FC? 23	265 FS?C 23	301 END
194 SF 10	230 11	266 GTO 09	
195 E1	231 SF 23	267 ST+ 07	
196 ST+ 08	232*LBL 07	268*LBL 10	

Black Jack (Twenty-One)

HP Co. – Standard Apps. (November 1979)

This program plays a simple version of the card game blackjack (twenty-one) . The calculator deals (without replacement) from a 104-card deck, reshuffling when all but 13 cards have been dealt. The player may bet any amount; if he doesn't place a bet, the value of his previous one will be used.

The player and dealer each receive two cards, one of the dealer's cards being exposed . The player may then either draw additional cards (hit) or not draw (stand). The object of the game is to reach, but not exceed, a score of 21 points, counting 10 for face cards, 1 or 11 for aces, and the face value for the remaining cards. If a player's first two cards count 21, he has blackjack and immediately collects 10 times his bet unless the dealer also has blackjack.

When hitting, a player who draws a card bringing his score over 21 is said to " bust" or " be busted" and he loses his bet. When the player stands on a score of 21 or less, the dealer must hit his own hand until his score exceeds 16. At that point the higher hand wins and the player's bank is updated. If the player and dealer should have the same score, the bet is a stand-off or a push .

Options allowed in casino-style blackjack such as splitting pairs, going down for double, and purchasing insurance are not included in this program .You must have an HP-41C with one additional Memory Module to run this program.

Program Highlight

With the 11 registers left after keying in this program, you can write a program to play blackjack using simple playing and betting schemes. The routine shown checks registers and flags used by the blackjack program to determine whether to hit or stand. If the playing program loses , it doubles its bet, eventually winning.

By adding still more memory modules to your HP-41C, more complicated playing strategies may be tried. Notice that this program requires the data memory size to be increased to 28.

Example:

Shuffle the deck, key in a seed of π , and play Blackjack using a \$2 bet.

Keystrokes:

[ASN] [ALPHA] [DL] [ALPHA] [Σ+]
 [ASN] [ALPHA] [HT] [ALPHA] [1/x]
 [ASN] [ALPHA] [S] [ALPHA] [√x]
 [USER]
 [XEQ] [ALPHA] [SH] [ALPHA]
 0 [STO] 21
 [π] [STO] 00
 2 DL

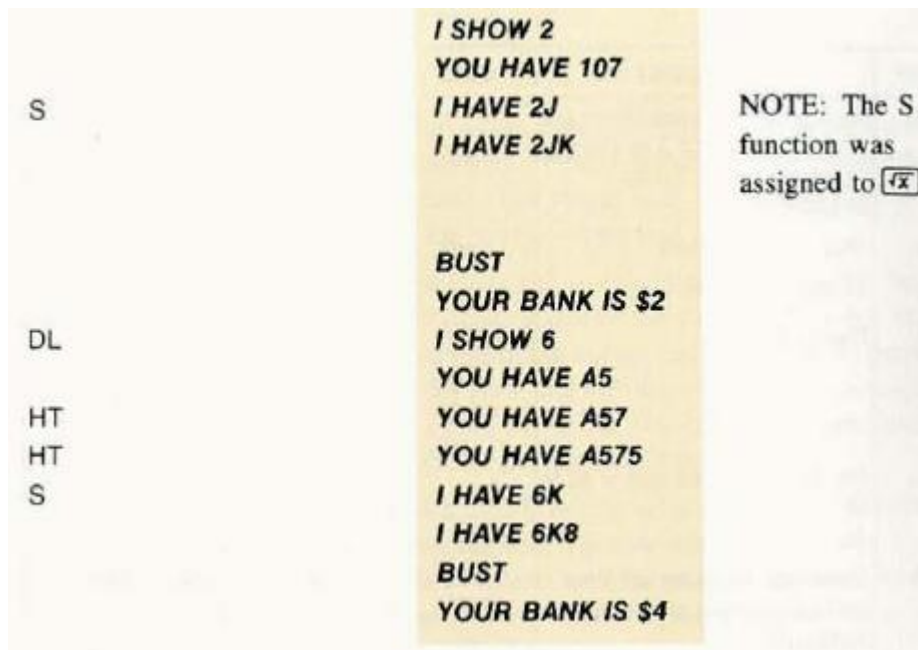
Display:

ASN DL 11
 ASN HT 12
 ASN S 13

 SHUFFLING
 104

Only FRC
(π) is used.

NOTE: The DL function was assigned to [Σ+]. Remember, your calculator must be in user mode or you will get Σ+.



Program listing:

01*LBL "CRD"	30 12	59 1	88 STO IND Y
02 CLA	31 RCL 14	60 +	89 ISG Y
03 ASTO 19	32 X>Y?	61 X=Y?	90 GTO 14
04 1	33 GTO 04	62 GTO "Q"	91 104
05 STO 15	34 XEQ "SH"	63 "K"	92 STO 14
06 RCL 00	35*LBL 04	64 GTO 01	93 CLD
07 9821	36 RCL 15	65*LBL A	94 CF 00
08 *	37 STO 16	66 "A"	95 CF 01
09 ,211327	38 10	67 CF 07	96 CF 02
10 +	39 X<=Y?	68 GTO 01	97 CF 03
11 FRC	40 GTO 00	69*LBL "Q"	98 CF 04
12 STO 00	41 X<>Y	70 "Q"	99 RTN
13 RCL 14	42 STO 16	71 GTO 01	100*LBL "DL"
14 *	43 1	72*LBL J	101 CF 09
15 INT	44 X=Y?	73 "J"	102 SF 07
16 1	45 GTO A	74 GTO 01	103 ABS
17 +	46 CLA	75*LBL "10"	104 INT
18*LBL 02	47 ARCL Y	76 "10"	105 FS?C 22
19 RCL IND 15	48 GTO 01	77*LBL 01	106 STO 22
20 X>Y?	49*LBL 00	78 ASTO 19	107 RCL 22
21 GTO 03	50 STO 16	79 RCL 16	108 STO 20
22 -	51 CLX	80 RTN	109 SF 06
23 ISG 15	52 10	81*LBL "SH"	110 CLA
24*LBL 99	53 X=Y?	82 "SHUFFLING"	111 ASTO 26
25 GTO 02	54 GTO "10"	83 AVIEW	112 ASTO 25
26*LBL 03	55 1	84 1,013	113 XEQ "CRD"
27 DSE IND 15	56 +	85 ENTER^	114 RCL 15
28*LBL 99	57 X=Y?	86 8	115 STO 17
29 DSE 14	58 GTO J	87*LBL 14	116 XEQ "CRD"

117 STO 23	155 11	192 GTO 08	229 X>Y?
118 CF 08	156 RCL 24	193 FS? 07	230 RTN
119 FS? 07	157 X>Y?	194 GTO 06	
120 SF 08	158 GTO 05	195 11	231 "BUST"
121 CLA	159 10	196 RCL 23	232 AVIEW
122 ARCL 19	160 ST+ 24	197 X>Y?	233 GTO 05
123 ARCL 25	161*LBL 05	198 GTO 06	<u>234*LBL "DB"</u>
124 ASTO 25	162 CF 07	199 7	235 "BUST"
125 "I SHOW "	163 FS? 08	200 X>Y?	236 AVIEW
126 ARCL 25	164 SF 07	201 GTO 06	237 0
127 AVIEW	165 RCL 17	202 10	238 RTN
128 SF 07	166 STO 15	203 ST+ 23	<u>239*LBL "PH"</u>
129 0	167 XEQ 04	204*LBL 08	240 ST+ 24
130 STO 24	168 XEQ "DH"	205 21,5	241 CLA
131 XEQ "CRD"	169 FS? 07	206 RCL 23	242 ARCL 26
132 XEQ "PH"	170 GTO 07	207 X>Y?	243 ARCL 19
133 XEQ "CRD"	171 11	208 XEQ "DB"	244 ASTO 26
134 XEQ "PH"	172 RCL 23	209 RCL 24	245 "YOU HAVE "
135 RCL 24	173 X#Y?	210 -	246 ARCL 26
136 10	174 GTO 07	211 X=0?	247 AVIEW
137 FS? 07	175 21,5	212 XEQ "P"	248 RTN
138 CLX	176 STO 23	213 X>0?	<u>249*LBL "DH"</u>
139 +	177 "I HAVE	214 SF 06	250 ST+ 23
140 21	BLACKJAC"	215*LBL 09	251 CLA
141 X#Y?	178 "K"	216 RCL 20	252 ARCL 25
142 SF 09	179 AVIEW	217 FS? 06	253 ARCL 19
143 FS? 09	180 GTO 07	218 CHS	254 ASTO 25
144 RTN	181*LBL 06	219 ST+ 21	255 "I HAVE "
145 21,5	182 XEQ "CRD"	220 "YOUR BANK	256 ARCL 25
146 STO 24	183 XEQ "DH"	IS \$"	257 AVIEW
147 1,5	184*LBL 07	221 ARCL 21	258 RTN
148 ST* 20	185 FS? 06	222 AVIEW	<u>259*LBL "P"</u>
149 "BLACKJACK"	186 GTO 09	223 RTN	260 "A PUSH"
150 AVIEW	187 FC? 09	<u>224*LBL "HT"</u>	261 AVIEW
151*LBL "S"	188 GTO 08	225 XEQ "CRD"	262 ST* 20
152 CF 06	189 RCL 23	226 XEQ "PH"	263 END
153 FS? 07	190 17	227 RCL 24	
154 GTO 05	191 X<=Y?	228 21,5	

Super Detective

Tom Langland, PPCCJ V12N8 p2 ; (August 1985)

There has been a robbery!! You, the Super Detective must find the guilty culprit. Here are the facts that are given to you:

1. A very expensive diamond was stolen
2. The diamond was in the center room of a nine room museum (see museum floor plan)
3. The diamond was stolen between one p.m. and midnight
4. Five people were wandering around in the museum between one p.m. and midnight. They all say that they were just visiting the museum, and were sight-seeing during that time. So no one ever stayed in the same room for two consecutive hours, but while they were wandering around they could have returned to the same room (even more than once).
5. The diamond must have been stolen at the end of an hour, because that is the only time the guard was ever away from the diamond.

You may question any of the 5 suspects by asking them about a specific time that they were in the museum, and they will tell you as much as they know. First, the suspect being questioned will tell you the room he was in at that time. If that was room #5, then the suspect will say whether or not the diamond was there or not. Next, you will be told who was in the room with the suspect. Finally, the suspect will tell you who he saw in any adjoining rooms.

Of course, because you are questioning the suspects they are getting very nervous (especially the actual thief!). So because of this, the innocent suspects may forget some information, and thus give you the wrong information. For innocent suspects, this will happen 5% of the time. For the actual thief, he wants to save himself and so lies even more. He will give you bad information 50% of the time. So when being told the room number a suspect is in, he may lie (or just be forgetting) and say he was in an adjacent room instead. The other information told to you by the suspects may also be in error and the suspects may lie and say they were with someone else instead of who they were really with, may not name someone they were with, or name someone and have really been with no one at all! This same thing can also happen with whoever a suspect says he saw in another room. Sometimes when you question the actual thief enough times, he will get so nervous that he will confess to the crime.

When you question someone who was in room #5, it is a little different. The suspect can not lie about being in that room, so if a suspect says that he was in room #5, then you know that is the truth. Also, if a suspect is in room #5, he can not lie and say he was in a different room. Statements about the diamond (because it is in room #5) are always true too, but anything else the suspect says (who he was with, or who he saw) can still be lied about! The diamond will be seen up until the moment it is stolen, so it may be seen at 2 p.m., then not seen at 3 p.m. which would mean the diamond was stolen at 2 p.m.

INSTRUCTIONS

First, set minimum size [XEQ] SIZE 098, and start the program with [XEO] "CATCHME". You will first be asked for a seed which should be between 0 and 1. This allows the calculator to make a new and different case every time you play. The program will take about 3 or 4 minutes to store the evidence for the case. When it has completed making the case for you, the calculator will beep and display **WHO STOLE THE DIAMOND?** Now press [R/S] and you will be able to begin questioning the suspects.

First, you will see the question number you are on, and then be prompted for the number of the suspect you wish to question (**ASK WHO?**). Key in the number of the suspect (1 - 5) and then [R/S] again. Next, you will be prompted for the time you want to question the suspect about (**AT TIME?**). Simply key in the time (1 - 12) and then [R/S] again. The suspect will now tell you what information he knows about that time he spent in the museum. You will see a review of who you are questioning and at what time (like asking the suspect what his name is). Next, the suspect will make some random remark, and then tell you what room he was in, followed by who he was with in that room and then who he saw in any adjoining rooms. This information should be written in a table, so that you can refer back to it as you question your suspects and can see who is lying. A free Super Detective Questioning Sheet (suitable for photocopying) will be sent to anyone sending me a self-addressed stamped envelope.

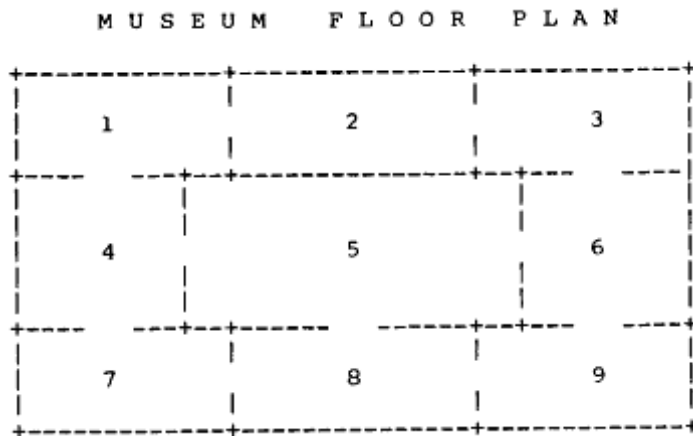
Of course after you have asked the suspects enough questions you will have a idea of who stole the diamond. Simply key in zero instead of a suspect number to question. You can then enter the number of the suspect you think is guilty, then the time you think the diamond was stolen (you must know both). If you are correct, you will hear the siren of the police car coming to pick up the suspect! If you get one of them wrong you will receive a ten question penalty, and can then continue questioning suspects. But if you get both wrong, you will be taken off the case.

Possible modifications:

This program contains no synthetic code and requires nothing more than the standard HP-41CV. Further modifications may be made to enhance the program's features or increase its speed. Here are some possibilities:

- The matrices which hold the museum floor plan and the suspects paths around the museum (registers 01 - 27 and registers 28 - 87) can be stored in an extended memory file, thus creating more room in RAM.
- Synthetic code could be added to give new tones or characters to display.
- You can change the probability of someone lying (thief line 196, any other suspect - line 190).
- Adding more suspects or perhaps a guard which may be questioned will add new variations. You might even have the suspect hide the diamond after it is stolen or maybe give it to an accomplice!
- New functions available with the PPC ROM or the Extended Functions module expand the possibilities.

I would certainly enjoy hearing from anyone who liked playing this game, and also if you have questions, comments or new variations to share!



Program listing:

01*LBL	30 5	60 2.012	90*LBL 05
"CATCHME"	31 STO 11	61 STO 90	91 RCL 91
02 FS? 55	32 STO 17	62*LBL 02	92 RCL 89
03 CF 21	33 STO 22	63 1.005	93 XEQ 97
04 RNG	34 6	64 STO 89	94 5
05 STO 00	35 STO 08	65*LBL 03	95 X=Y?
06*LBL 00	36 STO 14	66 RCL 90	96 GTO 06
07 "PLEASE WAIT"	37 STO 25	67 E	97 ISG 89
08 AVIEW	38 7	68 -	98 GTO 05
09 TONE 6	39 STO 12	69 RCL 89	99 GTO 04
10 CLX	40 STO 23	70 XEQ 97	100*LBL 06
11 STO 03	41 8	71 3	101 RCL 91
12 STO 06	42 STO 15	72 XEQ 99	102 5
13 STO 09	43 STO 20	73 X<>Y	103 XEQ 99
14 STO 21	44 STO 26	74 XEQ 96	104 STO 92
15 STO 27	45 9	75 X=0?	105 XEQ 97
16 TONE 7	46 STO 18	76 GTO 03	106 5
17 E	47 STO 24	77 RCL 90	107 X#Y?
18 STO 04	48 E2	78 RCL 89	108 GTO 06
19 STO 10	49 STO 88	79 XEQ 98	109 20
20 2	50 1.005	80 ISG 89	110 XEQ 99
21 STO 01	51 STO 89	81 GTO 03	111 E3
22 STO 07	52*LBL 01	82 ISG 90	112 /
23 3	53 9	83 GTO 02	113 STO 97
24 STO 05	54 XEQ 99	84*LBL 04	114 "WHO STOLE"
25 STO 16	55 E	85 12	115 AVIEW
26 4	56 RCL 89	86 XEQ 99	116 TONE 9
27 STO 02	57 XEQ 98	87 STO 91	117 TONE 7
28 STO 13	58 ISG 89	88 1.005	118 PSE
29 STO 19	59 GTO 01	89 STO 89	

119 "THE DIAMOND?"	170 ARCL 95	221 RCL 95	273 X<Y?
120 AVIEW	171 AVIEW	222 XEQ 97	274 TONE 6
121 TONE 5	172 TONE 5	223 STO 93	275 PSE
122 X^2	173 TONE 5	224 5	276 "SEE DIAMOND"
123 TONE 7	174 TONE 7	225 X=Y?	277 AVIEW
124 TONE 6	175 PSE	226 GTO 11	278 TONE 7
125 TONE 5	176 PSE	227 RCL 94	279 LN
126 STOP	177 "AT TIME "	228 20	280 E^X
127*LBL 07	178 ARCL 96	229 XEQ 99	281 TONE 7
128 "QUESTION "	179 >":"	230 X<=Y?	282 TONE 8
129 101	180 AVIEW	231 GTO 10	283 TONE 7
130 RCL 88	181 TONE 7	232 PSE	284 PSE
131 -	182 TONE 6	233 GTO 11	285*LBL 12
132 CF 29	183 TONE 5	234*LBL 10	286 RCL 94
133 FIX 0	184 PSE	235 3	287 20
134 ARCL X	185 PSE	236 XEQ 99	288 XEQ 99
135 AVIEW	186 E	237 RCL 93	289 X<=Y?
136 TONE 8	187 STO 94	238 XEQ 96	290 GTO 15
137 TONE 7	188 RCL 95	239 X=0?	291 1.005
138 LN	189 RCL 92	240 GTO 10	292 STO 89
139 E^X	190 X#Y?	241 5	293 RCL 96
140 TONE 6	191 GTO 09	242 X=Y?	294 RCL 95
141 TONE 7	192 E1	243 GTO 10	295 XEQ 97
142 PSE	193 STO 94	244 X<>Y	296 STO 93
143 PSE	194 ISG 97	245 STO 93	297*LBL 13
144 "ASK WHOM?"	195 GTO 09	246*LBL 11	298 RCL 89
145 AVIEW	196 "OK, I CONFESS"	247 "I WAS"	299 INT
146 TONE 6	197 AVIEW	248 AVIEW	300 RCL 95
147 TONE 8	198 TONE 8	249 TONE 5	301 X=Y?
148 STOP	199 TONE 7	250 TONE 6	302 GTO 14
149 INT	200 LN	251 PSE	303 RCL 96
150 STO 95	201 E^X	252 "IN ROOM "	304 RCL 89
151 X<=0?	202 TONE 6	253 ARCL 93	305 XEQ 97
152 GTO 20	203 LN	254 AVIEW	306 RCL 93
153 6	204 TONE 6	255 TONE 7	307 X#Y?
154 X<=Y?	205 TONE 5	256 TONE 6	308 GTO 14
155 GTO 07	206 DSE 88	257 E^X	309 "I WAS WITH "
156*LBL 08	207 PSE	258 TONE IND 93	310 ARCL 89
157 "AT TIME?"	208 PSE	259 PSE	311 AVIEW
158 AVIEW	209 "I STOLE IT"	260 PSE	312 TONE 7
159 TONE 5	210 AVIEW	261 RCL 93	313 TONE 6
160 TONE 7	211 TONE 5	262 5	314 E^X
161 STOP	212 E^X	263 X#Y?	315 TONE 7
162 INT	213 TONE 4	264 GTO 12	316 LN
163 STO 96	214 LN	265 "I DID"	317 TONE IND 89
164 X<=0?	215 TONE 3	266 RCL 96	318 PSE
165 GTO 08	216 PSE	267 RCL 91	319*LBL 14
166 13	217 GTO 23	268 X<Y?	320 ISG 89
167 X<=Y?	218*LBL 09	269 "" NOT"	321 GTO 13
168 GTO 08	219 XEQ 30	270 AVIEW	322 GTO 11
169 "SUSPECT "	220 RCL 96	271 TONE 8	323*LBL 15
		272 TONE 7	

324 PSE	376 TONE 6	427 STO 95	477 TONE 6
325 PSE	377 TONE 5	428 X<=0?	478 PSE
326 RCL 95	378 LN	429 GTO 20	479 E1
327 7	379 TONE IND 90	430 6	480 ST- 88
328 XEQ 99	380 PSE	431 X<=Y?	481 "KEEP TRYING"
329 X=Y?	381*LBL 11	432 GTO 20	482 AVIEW
330 GTO 15	382 ISG 90	433*LBL 21	483 TONE 6
331 PSE	383 GTO 17	434 "AT TIME?"	484 E^X
332 6	384*LBL 12	435 AVIEW	485 TONE 7
333 X<=Y?	385 ISG 89	436 TONE 6	486 TONE 8
334 GTO 11	386 GTO 16	437 TONE 7	487 TONE 7
335 "I WAS WITH "	387 GTO 11	438 STOP	488 PSE
336 ARCL Y	388*LBL 18	439 INT	489 GTO 07
337 AVIEW	389 PSE	440 STO 96	490*LBL 11
338 TONE 7	390 PSE	441 X<=0?	491 "NOT CLOSE"
339 TONE 6	391 RCL 95	442 GTO 21	492 AVIEW
340 E^X	392 E1	443 13	493 TONE 5
341 TONE 7	393 XEQ 99	444 X<=Y?	494 TONE 1
342 LN	394 X=Y?	445 GTO 21	495 PSE
343 TONE IND Y	395 GTO 11	446 RCL 92	496*LBL 12
344 PSE	396 5	447 RCL 95	497 "GIVE UP
345*LBL 11	397 X<Y?	448 X#Y?	NOW"
346 RCL 94	398 GTO 11	449 GTO 09	498 AVIEW
347 20	399 RDN	450 RCL 91	499 TONE 4
348 XEQ 99	400 RCL 89	451 RCL 96	500 TONE 3
349 X<=Y?	401 X=Y?	452 X#Y?	501 TONE 1
350 GTO 18	402 GTO 11	453 GTO 14	502 PSE
351 1.003	403 RDN	454 "YOU GOT	503 50
352 STO 89	404 STO 89	HIM"	504 ST- 88
353*LBL 16	405 "I SAW "	455 AVIEW	505*LBL 23
354 RCL 96	406 ARCL 89	456 1.005	506 "THIEF WAS "
355 RCL 95	407 AVIEW	457*LBL 22	507 ARCL 92
356 XEQ 97	408 TONE 6	458 TONE 8	508 AVIEW
357 RCL 89	409 TONE 5	459 TONE 7	509 TONE 7
358 INT	410 LN	460 TONE 8	510 TONE 6
359 X<>Y	411 TONE IND 89	461 TONE 7	511 LN
360 XEQ 96	412 PSE	462 ISG X	512 TONE 8
361 STO 93	413 GTO 18	463 GTO 22	513 PSE
362 X=0?	414*LBL 11	464 PSE	514 PSE
363 GTO 12	415 DSE 88	465 GTO 23	515 "AT TIME "
364 1.005	416 GTO 07	466*LBL 09	516 ARCL 91
365 STO 90	417 GTO 12	467 RCL 91	517 AVIEW
366*LBL 17	418*LBL 20	468 RCL 96	518 TONE 7
367 RCL 96	419 "WHO	469 X#Y?	519 TONE 8
368 RCL 90	GUILTY?"	470 GTO 11	520 E^X
369 XEQ 97	420 AVIEW	471*LBL 14	521 TONE 6
370 RCL 93	421 TONE 7	472 "PARTLY	522 PSE
371 X#Y?	422 E^X	RIGHT"	523 PSE
372 GTO 11	423 TONE 6	473 AVIEW	524 "RATING = "
373 "I SAW "	424 TONE 7	474 TONE 8	525 ARCL 88
374 ARCL 90	425 STOP	475 TONE 7	526 AVIEW
375 AVIEW	426 INT	476 E^X	527 TONE 8

528 TONE 8	561 "I THINK	587 GTO 11	620 +
529 E1	THAT"	588*LBL 40	621 RCL IND X
530 ST/ 88	562 GTO 11	589 "WHY ASK	622 RTN
531 E^X	563*LBL 32	ME?"	623*LBL 97
532 TONE IND 88	564 "LET ME	590 GTO 11	624 E
533 PSE	THINK"	591*LBL 41	625 -
534 PSE	565 GTO 11	592 "IT WAS "	626 12
535 "AGAIN? Y/N"	566*LBL 33	593 RCL 96	627 *
536 AVIEW	567 "I	594 7	628 +
537 TONE 6	REMEMBER..."	595 X<=Y?	629 27
538 TONE 7	568 SF 05	596 ""LATE.."	630 +
539 LN	569 GTO 11	597 X>Y?	631 RCL IND X
540 E^X	570*LBL 34	598 ""EARLY"	632 RTN
541 TONE 8	571 "JUST A SEC..."	599 X>Y?	633*LBL 98
542 TONE 7	572 GTO 11	600 SF 05	634 E
543 TONE 8	573*LBL 35	601*LBL 11	635 -
544 AON	574 "HARD TO	602 AVIEW	636 12
545 STOP	SAY.."	603 TONE 8	637 *
546 AOFF	575 GTO 11	604 E^X	638 +
547 ASTO Y	576*LBL 36	605 TONE 7	639 27
548 "Y"	577 "DON'T RUSH	606 LN	640 +
549 ASTO X	ME"	607 TONE 8	641 X<>Y
550 X=Y?	578 GTO 11	608 FS? 05	642 STO IND Y
551 GTO 00	579*LBL 37	609 LN	643 RTN
552 STOP	580 "I WAS	610 FS?C 05	644*LBL 99
553*LBL 30	LOST..."	611 TONE 7	645 RNG
554 CF 05	581 GTO 11	612 PSE	646 STO 00
555 11	582*LBL 38	613 PSE	647 *
556 XEQ 99	583 "WOULD I	614 RTN	648 INT
557 30	LIE?"	615*LBL 96	649 E
558 +	584 GTO 11	616 E	650 +
559 GTO IND X	585*LBL 39	617 -	651 END
560*LBL 31	586 "JUST	618 3	
	LOOKING"	619 *	

Mah-Jong Score keeping

[Frans de Vries - DataFile V7N6 p25 ; \(September 1988\)](#)

About a year ago, in Datafile V6N4p41-42, CofinLeggate presented a program for keeping the score during a game of Mah-Jong with four players. Judging from the program's length, 891 lines or 1739 bytes, it must have been an impressive task to write. Yet while looking at the code I noticed some obvious ways to save bytes. This aroused my interest, as any program of this size limits the use of the HP41 in leaving hardly room for other programs, key assignments, buffers, etc. a user may need. Thus I think programs that are longer than necessary are, in a way, not very user friendly, and therefore I started to work on reducing the length of the program. In the process of digging through the code I also discovered a minor bug in subroutine 43 as a numeric entry 1. or better LASTX, is missing between lines 470 and 471.

Applying the byte saving techniques that have been developed in the march of years, and which are excellently described by Alan McCornack and Keith Jarett in their recent book "HP-41 Advanced Programming Tips", I readily achieved a reduction of about 110 bytes. For example. the GTO's in lines 91, 131, 175 and 211 are superfluous, the numeric entry 10 in lines 257, 273, 289 and 305 can be replaced by LASTX, the instruction sequences CHS, ST+ rr all through the second half of the program are better replaced by ST- rr, and a GTO xx. can be substituted for each instruction sequence XEQ xx, RTN in the same area. More significantly, lines 494-512 are better superseded by three subroutine calls XEQ 56, XEQ 55, GTO 54 (that is what they exist for) and similar calls can be applied in subroutines 06, 11 and 16.

The shortness of Colin's article now turned out to be a bit of a nuisance because not only did he omit an explanation of the LBL E backup facility, or mention that an Extended Function Module and an IL Printer are required to run the program (which was easily deducted from the listing though, but that is not the point here), but no technical information (like data register usage) was given either. The obligatory use of an IL Printer (the dedicated 82143A printer won't do because of the use of the FMT instruction throughout the first half of the program) was felt as a particular severe restriction, so the next step was to remove all printer specific instructions and use only AVIEW instructions. Thus the program could be run with the dedicated printer or, better yet, without one.

Another major step in code reduction constituting not merely some local modifications; but a total rewrite of the program, was the application of looping and indirect addressing techniques. For example, lines 04-16, 20-35, 214-309, 329-388 and 394-437 can all be rolled into loops, while lines 735-878 can be turned into two subroutines accessing the proper registers indirectly. Reordering the use of labels and registers to take more advantage of shortform instructions also proved to be valuable.

All this resulted in a really huge reduction of some 600 bytes! The use of the program however had undergone only a few minor changes so far. Yet in order to 'break' the 1000-byte barrier I finally 'attacked' those parts of the program too and managed to bring the program's length down to about 965 bytes, which was less than 56 percent of the original!

The user interface of the program had been simplified for it did not print every prompt and input anymore (saving a lot of paper too), but was otherwise largely the same as before. Yet now the program could be kept in the 41 for longer periods than just while playing the game, as there was enough room for other applications the user might need from day to day. Or, if you look at it the other way around, you didn't have to remove most of your other programs anymore if you wanted to play a game of Mah-Jong.

While rewriting the program I also became acquainted with the rules of Mah-Jong from a booklet I obtained for this purpose. That is why I began to feel the need for additional features and, after so much reduction, I in turn started to extend the program. The first major feature to be included was a three player option, for which Colin offered a Separate program. Further useful features seemed to be: allowing to input a limit, to be applied to the scores that are accumulated each round, and requesting that the bonus for the player that went Mah-Jong be input separately, as well as any penalty points that each player might have blundered into. This increased the length to around 1110 bytes.

The last and also major feature to be incorporated was the draw option. That is, when a round ends because the players run out of stones, then the scores have to be settled between each two players (doubled when one of them is East Wind of course), but no other calculations have to be performed: This brought the program into its final, highly optimized form. It is 1199 bytes long and requires SIZE 028, but no peripherals.

Instructions for Using the Mah-Jong Score-keeping program.

Using the program consists of two parts, initialization and the actual score keeping

- For a game with three players, start up the program with XEQ "3JONG", otherwise, for a game with four players. use XEQ "4JONG"
- The first prompt now shows "PLAYER A ?" and Alpha mode is on. Enter the name of the first player. using at most six characters and press R/S. Repeat this for players B, C and if it is a fourplayer game, D.
- The list of player names and their associated initials A, B, C and possibly D is now displayed and, if a printer is connected, also printed. You have to press R/S each time if there is no printer, otherwise the program just runs on. It is important to remember this list, because all further input of player names to the program is by way of these initials.
- The last prompt in this initializing sequence is "LIMIT?"; which allows you to impose a limit to the number of points (positive or negative) each player can accumulate per round. If you press R/S without entering a number, there will be -no limit (uses flag 22 test). Otherwise the number you enter will be used to top off each score to that limit before it is added to a player's grand total. Of course the limit goes double for the player that is East Wind.

The program is now initialized and ready for round-to-round use.

1. The program has halted with flag 11 set, so you can turn the 41 off and start the program up again when the round is completed, simply by keying ON. It can also be started by pressing user key A.
2. The first prompt is "DRAW?". Press any number and R/S if the round ended without any player going Mah-Jong; otherwise just R/S (uses flag 22 test).
3. The following question asked is: who was "EAST WIND?". Alpha mode is on, so enter. Then initial A, B, C or, if-applicable, D of the player that was East Wind and press R/S. This assumes you apply the usual rule of rotating the name stones after each round in which East Wind does not go Mah-Jong, or after East Wind wins four rounds in a row. If you enter anything else "RUBBISH" is displayed shortly and a new prompt "A, B, OR C" or "A, B, C, OR D" appears. After a correct initial has been entered, the associated player's name is displayed shortly.
4. This step is skipped if the round ended in a draw, otherwise the initial of the player who went "MAH-JONG?" is requested. Input and resulting displays are the same as for step 3.
- 5-. Now it is time to enter the points, starting with the player that went Mah-Jong or else the one that was East Wind. The display shows his/her name with "= ?" appended. Enter the basic value of the stones and press R/S. If the player went Mah-Jong, then the "M-J BONUS?" is to be input next. Enter the basic bonus for going Mah-Jong added to any additional bonus for the way he/she did it and press R/S. Furthermore, the number of times these points have to be "DOUBLED?" must be entered, followed by R/S, and finally any "PENALTY?" points the player deserves for disturbing the game by violating some rule can be entered, as usual concluded by R/S. This step is repeated for the other players. Although this step intermediate calculations in the stack are allowed, as long as the number to be input is in X when R/S is keyed. The program always halts with a cleared stack, so when R/S is pressed without input the default value zero is registered.
6. The program now calculates the scores and when this is finished (it takes just a few seconds), it displays and, if a Printer is connected, prints the round number. Press R/S if there is no printer, otherwise the program just runs on. This also holds for the remaining parts in this step. The first part displays / prints the scores of each player in the current round in order of initials. The format is "name (pnts,dbld)= score", in which "name" is the player's name, "pnts" is his/hers basic stones value plus the Mah-Jong bonus (if applicable) minus the penalty points, "dbld" is the number of doubles and 'score' is the result from the calculations performed by the program. The second part shows the running "GRAND TOTALS" of each player after the scores have been added again in order of initials. The format is "name= total" which speaks for itself.

After the last display/print the program returns to step 1 ready for the next round. If anywhere through steps 2 thru 6 you make an erroneous input, then you can return to step 2 with the original grand totals after the previous round back in place, by pressing user key E. The display shortly shows "BACKUP" and a beep sounds to indicate this.

There is no special way to end the program. When the game ends, the set of grand totals last displayed/printed simply is the final one. For safety reasons you could manually clear

flag 11 though, or run some kind of clean-up routine to clear registers and reset flags, display settings, etc. to your favourite defaults.

Technical Details of the Mah-Jong Score-keeping Program

The program requires no other equipment than an HP-41 with sufficient memory to accommodate the 1199 bytes of code and 28 data registers. It uses 4 user flags and only a few synthetic instructions like short-form exponents and the F0 NOP. The display is set to FIX 0/CF 29 after one run through the score-keeping part.

Register Usage:

R00: Basic Score A	R10: Grand Tot.1 C	R18: Displayable Score C
R01: Basic Score B	R11: Grand Total 0	R19: Displayable Score D
R02: Basic Score C	R12: Round NumBer	R20: Player's Name A
R03: Basic Score D	R13: Limit	R21: Player's Name B
R04: Round Score A	R14: Subroutine Pointer/ Loop Counter	R22: Player's Name C
R05: Round Score B	R15: Register Pointer/ Loop Counter	R23: Player's Name 0
R06: Round Score C	R16: Displayable Score A	R24: Backup Total A
R07: Round Score D	R17: Displayable Score B	R25: BackupTotal B
R08: 'Grand Total A		R26: Backup Total C
R09: Grand Total B		R27: Backup Total D

Flag Usage:

Set	Clear	
F00: 4 Players	3 Players	F11: Auto-start
F01: Limit	No limit	F12: Double wide printing
F02: Draw	No draw	F22: Numeric input detection
F10: Scratch	F29: No radix	

The number of times a score can be doubled in step 5 of the score-keeping part is limited to 9. If higher numbers turn up in the game, which is not very likely though, the program must be changed to accommodate them. Replace line 118 E1 with E and substitute FIX 2 for line 158 FIX 1

If you use an IL Printer and want to take advantage of its battery saving feature, then insert a PWRDN instruction between lines 53 and 54 and P WRUP instructions after lines 27 and 144.



Program listing:

I am not going to give a full line by line analysis of the program, as it would simply take up too much space here and take me too much time. If you are interested, then you are invited to delve into the code yourself. I think you will like it, as a number of programming and byte-saving techniques have been employed that can be useful in other applications too. You may consider yourself to be familiar with the code when you fully understand what line 144 and the LBL 00 subroutine at line 530 do. Enjoy playing Mah-Jong and the way this program simplifies the tedious job of score keeping.

<u>01*LBL</u>	43 CF 12	88 FS? 02	129
<u>"3JONG"</u>	44 AOFF	89 RCL 15	"PENALTY?"
02 CF 00	45 CF 01	90 4	130 XEQ 07
03 GTO 00	46 CF 22	91 *	131 ST- IND Y
<u>04*LBL</u>	47 "LIMIT?"	92 ST+ 14	132 ST- IND
<u>"4JONG"</u>	48 PROMPT	93 AOFF	15
05 SF 00	49 FS?C 22	94 CLD	133 RCL 15
06*LBL 00	50 SF 01	95 RCL 14	134 ISG X
07 SIZE?	51 STO 13	96 .1	135 ""
08 28	52*LBL 43	97 %	136 4
09 X>Y?	53 ADV	98 +	137 FC? 00
10 PSIZE	54 ADV	99 3	138 DSE X
11 CLRG	55 ADV	100 -	139 MOD
12 AON	56 CLA	101 STO 14	140 STO 15
13 23	57 CLST	102 CF 10	141 ISG 14
14 FC? 00	58 SF 11	103 FC? 02	142 GTO 03
15 DSE X	59 STOP	104 SF 10	143 CLX
16 E3	60*LBL A	105 FC? 00	144 STO 04
17 /	61 RCL 08	106 ISG 14	145 STO 05
18 20	62 STO 24	107*LBL 03	146 STO 06
19 +	63 RCL 09	108 XEQ 12	147 STO 07
20 ENTER^	64 STO 25	109 "`=?"	148 XEQ IND
21*LBL 01	65 RCL 10	110 XEQ 07	14
22 XEQ 06	66 STO 26	111 STO IND Y	149 SF 21
23 ASTO L	67 RCL 11	112 STO IND	150 CF 29
24 "PLAYER "	68 STO 27	15	151 FIX 0
25 ARCL L	69 14	113 FC?C 10	152 " ROUND
26 "` ?"	70 STO 14	114 GTO 00	"
27 PROMPT	71 CF 21	115 "M-J	153 ARCL 12
28 ASTO IND	72 ISG 12	BONUS?"	154 XEQ 08
X	73 ""	116 XEQ 07	155*LBL 04
29 ISG X	74 CF 12	117 ST+ IND Y	156 XEQ 12
30 GTO 01	75 CF 22	118 ST+ IND	157 "` ("
31 RDN	76 "DRAW?"	15	158 4
32 SF 12	77 PROMPT	119*LBL 00	159 -
33 SF 21	78 FS?C 22	120	160 2
34	79 SF 02	"DOUBLED?"	161 RCL IND Y
"REMEMBER..."	80 AON	121 XEQ 07	162 FIX 1
35 AVIEW	81 "EAST	122 E1	163 XEQ 09
36*LBL 02	WIND?"	123 %	164 FC? 01
37 XEQ 06	82 XEQ 11	124 ST+ IND Z	165 GTO 00
38 "`= "	83 ST+ 14	125 2	166 RCL 15
39 ARCL IND	84 FC? 02	126 RCL Z	167 INT
X	85 "MAH-	127 Y^X	168 RCL 14
40 AVIEW	JONG?"	128 ST* IND	169 INT
41 ISG X	86 FC? 02	15	170 15
42 GTO 02	87 XEQ 11		171 -

172 4	227 PSE	283 DSE Z	338 +
173 MOD	228 RCL 24	284 GTO 10	339 CLA
174 X=Y?	229 STO 08	285 ARCL Y	340 ARCL IND
175 FS? 02	230 RCL 25	286 RTN	X
176 X<0?	231 STO 09	287*LBL 11	341 RTN
177 SF 10	232 RCL 26	288 PROMPT	342*LBL 15
178*LBL 00	233 STO 10	289 CLX	343 FS?C 02
179 RCL 15	234 RCL 27	290 SIGN	344 GTO 13
180 ENTER^	235 STO 11	291 ASTO Y	345 .
181 " `)="	236 DSE 12	292 "A"	346 SF 10
182 4	237 ""	293 ASTO X	347 XEQ 00
183 ST+ Z	238 GTO A	294 X=Y?	348 STO 05
184 ST+ X	239*LBL 06	295 GTO 09	349 STO 06
185 +	240 ABS	296 "B"	350 STO 07
186 5	241 "D"	297 ASTO X	351 GTO 14
187 RCL IND Z	242 FS? 00	298 X=Y?	352*LBL 13
188 FC? 01	243 ISG L	299 GTO 08	353 XEQ 31
189 GTO 00	244 "C"	300 "C"	354 XEQ 32
190 SIGN	245 ISG L	301 ASTO X	355 XEQ 33
191 ST* L	246 "B"	302 X=Y?	356*LBL 14
192 X<> L	247 ISG L	303 GTO 07	357 XEQ 40
193 RCL 13	248 "A"	304 FC? 00	358 XEQ 41
194 FS?C 10	249 RTN	305 GTO 00	359 GTO 42
195 ST+ X	250*LBL 07	306 "D"	360*LBL 00
196 X<Y?	251 CLST	307 ASTO X	361 FS?C 02
197 X<>Y	252 PROMPT	308 X=Y?	362 GTO 13
198 X<> L	253 ABS	309 GTO 06	363 E
199 *	254 INT	310*LBL 00	364 SF 10
200*LBL 00	255 RCL 15	311 "RUBBISH"	365 XEQ 00
201 ST+ IND Z	256 16	312 AVIEW	366 STO 04
202 FIX 0	257 +	313 TONE 0	367 STO 06
203 XEQ 09	258 X<>Y	314 PSE	368 STO 07
204 AVIEW	259 RTN	315 "A, B"	369 GTO 14
205 ISG 15	260*LBL 08	316 FS? 00	370*LBL 13
206 GTO 04	261 SF 12	317 " ` , C OR	371 XEQ 31
207 ADV	262 AVIEW	D"	372 XEQ 34
208 "GRAND	263 CF 12	318 FC? 00	373 XEQ 35
TOTALS"	264 E-3	319 " ` OR C"	374*LBL 14
209 XEQ 08	265 ENTER^	320 GTO 11	375 XEQ 38
210*LBL 05	266 ST+ X	321*LBL 06	376 XEQ 39
211 XEQ 12	267 FS? 00	322 ISG L	377 GTO 42
212 " ` ="	268 +	323*LBL 07	378*LBL 25
213 12	269 STO 15	324 ISG L	379 FS?C 02
214 -	270 RTN	325*LBL 08	380 GTO 13
215 5	271*LBL 09	326 ISG L	381 2
216 RCL IND Y	272 X#0?	327*LBL 09	382 SF 10
217 XEQ 09	273 X>0?	328 LASTX	383 XEQ 00
218 AVIEW	274 " ` "	329 STO 15	384 STO 04
219 ISG 15	275 ENTER^	330 XEQ 12	385 STO 05
220 GTO 05	276 ABS	331 AVIEW	386 STO 07
221 GTO 43	277*LBL 10	332 PSE	387 GTO 14
222*LBL E	278 RCL Z	333 RDN	388*LBL 13
223 CF 21	279 10^X	334 RTN	389 XEQ 32
224 "BACKUP"	280 X>Y?	335*LBL 12	390 XEQ 34
225 AVIEW	281 " ` "	336 RCL 15	391 XEQ 36
226 BEEP	282 RDN	337 20	392*LBL 14

393 XEQ 37	449 STO 06	505 3	561 .
394 XEQ 39	450 STO 07	506 XEQ 00	562 1
395 GTO 41	451 XEQ 32	507 STO 04	563 GTO 00
396*LBL 30	452 XEQ 33	508 X<>Y	564*LBL 32
397 FS?C 02	453 GTO 42	509 STO 05	565 SF 10
398 GTO 13	454*LBL 21	510 STO 06	566*LBL 38
399 3	455 E	511 XEQ 31	567 .
400 SF 10	456 XEQ 00	512 XEQ 32	568 2
401 XEQ 00	457 STO 06	513 GTO 40	569 GTO 00
402 STO 04	458 X<>Y	514*LBL 28	570*LBL 33
403 STO 05	459 STO 04	515 3	571 SF 10
404 STO 06	460 STO 07	516 XEQ 00	572*LBL 39
405 GTO 14	461 XEQ 32	517 STO 05	573 .
406*LBL 13	462 XEQ 36	518 X<>Y	574 3
407 XEQ 33	463 GTO 39	519 STO 04	575 FS? 00
408 XEQ 35	464*LBL 22	520 STO 06	576 GTO 00
409 XEQ 36	465 E	521 XEQ 31	577 CF 10
410*LBL 14	466 XEQ 00	522 XEQ 34	578 RTN
411 XEQ 37	467 STO 07	523 GTO 38	579*LBL 34
412 XEQ 38	468 X<>Y	524*LBL 29	580 SF 10
413 GTO 40	469 STO 04	525 3	581*LBL 40
414*LBL 16	470 STO 06	526 XEQ 00	582 E
415 .	471 XEQ 33	527 STO 06	583 2
416 XEQ 00	472 XEQ 36	528 X<>Y	584 GTO 00
417 STO 05	473 GTO 38	529 STO 04	585*LBL 35
418 X<>Y	474*LBL 23	530 STO 05	586 SF 10
419 STO 06	475 2	531 XEQ 32	587*LBL 41
420 STO 07	476 XEQ 00	532 XEQ 34	588 E
421 XEQ 34	477 STO 04	533 GTO 37	589 3
422 XEQ 35	478 X<>Y	534*LBL 00	590 FS? 00
423 GTO 42	479 STO 05	535 RCL X	591 GTO 00
424*LBL 17	480 STO 07	536 4	592 CF 10
425 .	481 XEQ 31	537 +	593 RTN
426 XEQ 00	482 XEQ 33	538 RCL IND Y	594*LBL 36
427 STO 06	483 GTO 41	539 CHS	595 SF 10
428 X<>Y	484*LBL 24	540 ENTER^	596*LBL 42
429 STO 05	485 2	541 ST+ X	597 2
430 STO 07	486 XEQ 00	542 ENTER^	598 3
431 XEQ 34	487 STO 05	543 ST+ X	599 FS? 00
432 XEQ 36	488 X<>Y	544 CHS	600 GTO 00
433 GTO 41	489 STO 04	545 STO IND T	601 CF 10
434*LBL 18	490 STO 07	546 RDN	602 RTN
435 .	491 XEQ 31	547 FS? 00	603*LBL 00
436 XEQ 00	492 XEQ 35	548 FC?C 10	604 RCL IND Y
437 STO 07	493 GTO 39	549 X>Y?	605 RCL IND Y
438 X<>Y	494*LBL 26	550 ST- IND Z	606 -
439 STO 05	495 2	551 X<>Y	607 FS?C 10
440 STO 06	496 XEQ 00	552 FC? 00	608 ST+ X
441 XEQ 35	497 STO 07	553 FS?C 10	609 4
442 XEQ 36	498 X<>Y	554 X<Y?	610 ST+ T
443 GTO 40	499 STO 04	555 ST+ IND Z	611 ST+ Z
444*LBL 19	500 STO 05	556 X<>Y	612 X<>Y
445 E	501 XEQ 33	557 RTN	613 ST+ IND T
446 XEQ 00	502 XEQ 35	558*LBL 31	614 ST- IND Z
447 STO 04	503 GTO 37	559 SF 10	615 END
448 X<>Y	504*LBL 27	560*LBL 37	

Rummy Score Keeping

Wolfgang Pawlowsky, PRISMA 87/4 p12

At the end of each round of the board game "mini Rummy" or "Rummy exclusive" (Otto Maier Verlag/Ravensburg) the minus points of each player are to be noted, while the winner gets the sum credited. Now I had the idea to have the HP 41 CV note and calculate the round and final results. Here now as result my program "RUMMY" in the simplest version for HP 41 CV + printer 82143A. SIZE = 010.

The computer prompts for the expression of the 5 headers for entering the minus points of a maximum of 4 players A to D. These values are entered positively. The winner of the lap gets a 0 as input. In program, the sum of the minus points is calculated and in the following printout this number appears positively in the column of the winner of the round. In the 1st column the consecutive number of the played rounds is ejected.

After the 2. After the 2nd round the program balances the individual column totals and prints the respective scores behind a "sigma-Character". This can be used to the winner and the winner of each game placements, which is similarly exciting It's like election night.

The maximum number of points is 999, you've been playing for 3 hours would have to. Is! the game ends, a pressure warries to the "E" key (= end) for the final double line and 4 ADVs to be able to tear off the strip for the subsequent "settlement".

Owners of the CCD module can be informed by "PRL" and "ACLX" save several bytes, but after that, there's still two magnetic cards with 4 tracks required for recording the program.

The "RUMMY" program is available for a maximum of 4. I've got a game of this game. If fewer players are the same as for the round winner zeros for the missing places. The Results of unoccupied places are available at ignore (there is the sum of all profits).

Wishing all "Rummy" gamers "Happy Playing"

Program listing:

01*LBL "RUMMY"	15 SF 27	29 STO 00
02*LBL A	16 CF 29	30*LBL 00
03 XEQ 01	17 "Rde:1 A1 B1"	31 XEQ 02
04 "*"	18 "` C1 D "	32 "` A?"
05 ACA	19 PRA	33 PROMPT
06 SF 12	20 SF 12	34 ST+ 05
07 " R u m m y "	21 "----"	35 CHS
08 ACA	22 ACA	36 STO 01
09 CF 12	23 ACA	37 XEQ 02
10 "*"	24 ACA	38 "` B?"
11 ACA	25 ADV	39 PROMPT
12 ADV	26 CF 12	40 ST+ 05
13 XEQ 01	27 FIX 0	41 CHS
14 CLRG	28 1	42 STO 02

43 XEQ 02	89 RCL 02	134 CF 12
44 "`C?"	90 XEQ 03	135 ADV
45 PROMPT	91 RCL 02	136 ADV
46 ST+ 05	92 ST+ 07	137 ADV
47 CHS	93 ACX	138 ADV
48 STO 03	94 RCL 03	139 STOP
49 XEQ 02	95 XEQ 03	140 GTO A
50 "`D?"	96 RCL 03	<u>141*LBL e</u>
51 PROMPT	97 ST+ 08	142 SF 12
52 ST+ 05	98 ACX	143 "s"
53 CHS	99 RCL 04	144 ACA
54 STO 04	100 XEQ 03	145 CF 12
55 RCL 01	101 RCL 04	146 RCL 06
56 X=0?	102 ST+ 09	147 ABS
57 RCL 05	103 ACX	148 " "
58 STO 01	104 ADV	149 10
59 RCL 02	105 1	150 X<=Y?
60 X=0?	106 ST+ 00	151 " "
61 RCL 05	107 2	152 RDN
62 STO 02	108 RCL 00	153 100
63 RCL 03	109 X>Y?	154 X<=Y?
64 X=0?	110 GTO e	155 " "
65 RCL 05	111 GTO 00	156 ACA
66 STO 03	112*LBL 02	157 RCL 06
67 RCL 04	113 "Σ SPIELER"	158 ACX
68 X=0?	114 RTN	159 RCL 07
69 RCL 05	115*LBL 03	160 XEQ 03
70 STO 04	116 ABS	161 RCL 07
71 0	117 " "	162 ACX
72 STO 05	118 10	163 RCL 08
73 RCL 00	119 X<=Y?	164 XEQ 03
74 ACX	120 " "	165 RCL 08
75 RCL 01	121 RDN	166 ACX
76 ABS	122 100	167 RCL 09
77 " "	123 X<=Y?	168 XEQ 03
78 10	124 " "	169 RCL 09
79 X<=Y?	125 ACA	170 ACX
80 "___"	126 RTN	171 ADV
81 RDN	<u>127*LBL E</u>	172 GTO 00
82 10	128 SF 12	173*LBL 01
83 X<=Y?	129 "===="	174 "-----"
84 "___"	130 ACA	175 ASTO ^
85 ACA	131 ACA	176 PRA
86 RCL 01	132 ACA	177 END
87 ST+ 06	133 ADV	
88 ACX		

Sea Battle, Printer version.

[Mark Cracknell–Data File V8N6 p9 ; \(September 1989\)](#)

Requirements: HP-41CX, Peripheral printer.

Sea Battle is an old game played on many different sized battlecharts with many different sets of rules. This version is for the solo player against the machine.

You, the player, have a limited amount of ammunition in the form of twenty salvoes, each of three shots, and a limited amount of time, one hour, to sink an enemy battlefleet consisting of one battleship, two cruisers, three torpedo-boats and four submarines.

This version also contains a large number of synthetic lines. You should note that their use saves time, particularly the global labels "[A] -[E]". If you have problems substituting non-synthetic steps for the synthetic ones then please do not hesitate to contact me and I will do my best to solve your problem. If you do not have a PPC ROM and/or an HP82162A printer, let me know, and I will attempt to modify the program to suit your system. Include suitable magnetic media (cards/tape.3.5' disc) and if I have been successful, I will record the modified version for you.

The battlechart consists of a ten by ten grid of squares. The HP41CX will obey certain rules when positioning its fleet:

- (1) No two ships may touch at their corners or lie alongside one another.
- (2) No ship, or part of a ship, may occupy square 00.
- (3) No ship may lie on a diagonal.

When you have identified the position of a ship you may cross off all the squares immediately around it. The steps listed under "L8L C" (step 541) will place a cross in the indicated square. In the event that you change your mind about crossing off a square the steps listed under "L8L 0" (step 556) will blank out the indicated square. If you accidentally put across or a blank in a numbered square the steps listed under "LBL 8" (step 515) will restore it. If you mess up the battlechart, and having corrected it, wish to see it reprinted before your next salvo, key "e" and a new battlechart will be output – but remember that the clock is still ticking away. If you lose patience, or run out of time, and want to find where you went wrong, just key "a" and all will be revealed.

The scoring system is simple. A score of 1000 indicates a hit on the battleship, 100 indicates a hit on a cruiser, 10 indicates a hit on a torpedo-boat, and 1 means you have sunk a submarine. This version of 'Battleships' does not allow you to hit a ship more than once with the same salvo; the battlechart will register both hits but only one will be scored. Thus in the sample game a score of 200 at Salvo 9 indicates hits on BOTH cruisers rather than two hits on the same cruiser.

This program is based on an HP-67 program, written by, I believe, Peter Amlinger. Due credit must also be given to the authors of the PPC ROM without whose efforts this program wouldn't have been possible.

Precise Instructions, listings and part of a sample game follow.

User Instructions

(1) Enter Program: XEQ [Alpha] "SBP" [Alpha] USER

(2) Key in seed and - "A"

After a period of time the initial battlechart will be output.

(3) First shot ENTER^

Second Shot ENTER^

Third shot "[E]"

After a period of time the revised battlechart will be output. At the beginning of each battlechart the salvo number, the location of each shot and the salvo score is printed; After the location plan of your shots to date is output a reminder of which ships were hit with which salvo is printed. Finally the elapsed time is shown and the HP41 is ready for the next salvo. The process should be repeated till either you have sunk the entire enemy battlefleet or run out of time and/or ammunition.

(4) To put a cross in a square, key in the square number and ----- "[e]"

(5) To blank out a square, key in the square number and ----- "[D]"

(6) To put a number in a square, key in the number and ENTER^
then key in the square number and --- "[B]"

(7) To reprint the battlechart ----- "[e]"

(8) To reveal the enemy battlefleet --- "[a]"

THE SAMPLE GAME

To play the sample game enter 0.01011983 and key "[A]". After a few minutes the initial battlechart will be output.

Salvo 1 >>> 14 Enter 71 Enter 51 "[E]". A bit of luck as the score of 100 shows a hit on a cruiser.

Salvo 2 >>> The cruiser could be in 14, 15, 16 so try: 15, Enter, 95, Enter, 20, "[E]" - The score of 10 shows a hit on a torpedo-boat but a miss for the cruiser.

Salvo 3 >>> Perhaps the cruiser is 61, 71, 87 so let us try: 67, Enter, 38, Enter, 83, "[E]". What luck! the score of 100 shows a hit on a cruiser. As 03 does not appear next to 02 the cruiser hit with this salvo is not the same 85 the one hit with Salvo 2.

Salvo 4 >>> Try: 87, Enter, 32, Enter, 59, "[E]" A score of 1000 shows a hit on the battleship and that the cruiser is not in 61, 77, 81.

Salvo 5 >>> Fire: 57, Enter, 28, Enter, 70, "[E]" for a go at the battleship. The score of 1000 means another hit on the battleship. It must be in 56, 57, 58, 59 as this is the only location where an 04 and an 05 are close enough. Now we can cross off squares 45-49, 55, 56, 66, 68, 69. To do this wekey nn and then "[C]" for each square.

		SALVO 2 15 95 20 S/SCORE=10	SALVO 4 87 32 59 S/SCORE=1000
MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X
		T/SCORE=110 B C 01 T 02 S	T/SCORE=1210 B 04 C 01 03 T 02 S
	0:04:18	SALVO 3 67 38 83 S/SCORE=100	SALVO 5 57 28 70 S/SCORE=1000
MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X	MCH 0 1 2 3 4 5 6 7 8 9 00X 10X 20X 30X 40X 50X 60X 70X 80X 90X
T/SCORE=100 B C 01 T S	T/SCORE=210 B C 01 03 T 02 S	T/SCORE=2210 B 04 05 C 01 03 T 02 S	

Salvo 6 >>> 58, Enter, 52, Enter, 85, "[E]". The score of 1000 shows two big holes in the ocean and the expected battleship hit.

Salvo 7 >>> Let us have a swipe at the cruiser we hit with Salvo 1 and sink the battleship so we fire: 61, Enter, 56, Enter, 8, "[E]". A score of 1100 shows success.

Salvo' 8 >>> The torpedo-boat hit with Salvo 2 could be 20, 21 so try: 41, Enter, 92, Enter, 21, "[E]". The score of 10 means a hit on a torpedo-boat but is not necessarily the same torpedo-boat as we hit with Salvo 2. Hissing the cruisers with this salvo is not a total loss as the only place where two cruiser hits are close enough to be in the same ship are in 51 and 61. If 41 is a miss then it means that a cruiser must be in 51, 61, 71 so we can cross off 40, 42, 50, 52, 60, 62, 72, 80, 81, 82 in the same manner as we did before firing Salvo 9.

Salvo 9 >>> Now that we have fixed the position of one of the cruisers it is safe to fire at both of them with the same salvo. One hit is expected, and a score of 200 or more will mean that both cruisers are well on the way to Davy Jones' locker. We will fire: 71, Enter, 37, Enter, 16, "[E]". Success – a score of 200 means one cruiser is sunk and the other is very badly damaged.

Salvo 10 >>> The remaining cruiser must be in either 36, 37, 38 or 37, 38, 39 because 37, 38 are the only two squares where an 03 and an 09 are close enough. Try: 39, Enter, 23, Enter, 74, "[E]". The score of 110 indicates a terminal hit on the cruiser and another hit on a torpedo boat.

Salvo 11 >>> We haven't sunk a submarine yet so we'll take a shot at some of the open spaces - who knows what we might find lurking beneath the waves. 64, Enter, 91 Enter, 76, "[E]" seems as good a go as any. The score of 1 means that it's 1 down, 3 to go.

I will leave you to finish this game off. There are only 6 hits needed and there are 9 salvos left so you have a fair chance.

I have two other versions of this game, one for the HP 32 column Video interface and one for the 80 column Mountain Computer Video Interface. The User Instructions are the same, it's just the presentation of the battlechart that is different. If you want either, or both, send me a self-addressed envelope for a listing. Include suitable magnetic media (cards, tape or 3.5" disc) and I will record it/them for you.

I hope that the game will give you as much pleasure in the future as it has given me over the last 9 years.

Mark Cracknell [#129]

SALVO 6
58 52 85
S/SCORE=1000

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X02 05 03
30X 04 03
40X XXXXXX
50X 0106 X 050604
60X XX03XX
70X05 01
80X 03 06 04
90X 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=3210

B 04 05 06
C 01 03
T 02
S

0:13:35

SALVO 8
41 92 21
S/SCORE=10

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X0208 05 03
30X 04 03
40X 08 XXXXXX
50X 0106 X 07050604
60X 07 XX03XX
70X05 01
80X 03 06 04
90X 08 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=4320

B 04 05 06 07
C 01 03 07
T 02 08
S

0:17:46

SALVO 10
39 23 74
S/SCORE=110

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X0208 10 05
30X 04 090310
40X08X XXXXXX
50X01X X 07050604
60X07X XX03XX
70X0509X 10 01
80XXX03 06 04
90X 08 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=4630

B 04 05 06 07
C 01 03 07 09 09 10
T 02 08 10
S

0:22:41

SALVO 7
61 56 8
S/SCORE=1100

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X02 05 03
30X 04 03
40X XXXXXX
50X 0106 X 07050604
60X 07 XX03XX
70X05 01
80X 03 06 04
90X 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=4310

B 04 05 06 07
C 01 03 07
T 02
S

0:15:35

SALVO 9
71 37 16
S/SCORE=200

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X0208 10 05
30X 04 090310
40X08X XXXXXX
50X01X X 07050604
60X07X XX03XX
70X0509X 10 01
80XXX03 06 04
90X 08 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=4520

B 04 05 06 07
C 01 03 07 09 09
T 02 08
S

0:20:38

SALVO 11
64 91 76
S/SCORE=1

```

MCE 0 1 2 3 4 5 6 7 8 9
XXXXXXXXXXXXXXXXXXXX
00X 07 01 02 05 03
10X 04 03 02 01
20X0208 10 05
30X 04 090310
40X08X XXXXXX
50X01X X 07050604
60X07X 11XX03XX
70X0509X 10 1101
80XXX03 06 04
90X 1108 02
XXXXXXXXXXXXXXXXXXXX

```

T/SCORE=4631

B 04 05 06 07
C 01 03 07 09 09 10
T 02 08 10
S 11

0:24:54

Program listing:

01*LBL "SBP"

02 CF 22	54 GTO 14	107 RCL 22	160 GTO 00
03 SF 27	55*LBL 02	108 /	161 RCL IND 25
04 CF 29	56 0	109 FRC	162 X=0?
05 156	57 STO 00	110 RCL 01	163 GTO 08
06 PSIZE	58 19	111 RCL 22	164*LBL 07
07 FIX 0	59 RCL 25	112 /	165 STO 02
08 CLST	60 X=Y?	113 FRC	166 INT
09 RTN	61 XEQ 03	114 -	167 X=0?
10*LBL "A"	62 17	115 ABS	168 GTO 00
11 PWRDN	63 X<=Y?	116 .9	169 RCL 01
12 SF 04	64 XEQ 03	117 X=Y?	170 FS? 00
13 CLRG	65 RCL 25	118 GTO 02	171 GTO 11
14 FC? 22	66 14	119 RCL 03	172 -
15 TIME	67 X<=Y?	120 ST+ 01	173 ABS
16 FRC	68 XEQ 03	121 0	174 2
17 STO 20	69 TIME	122 RCL 01	175 X>Y?
18 55	70 RN#	123 X<=Y?	176 GTO 06
19 TFX	71 RCL 23	124 GTO 02	177 RDN
20 CF 00	72 *	125 RCL 23	178 RCL 22
21 E1	73 INT	126 X<=Y?	179 -
22 STO 22	74 X=0?	127 GTO 02	180 ABS
23 X^2	75 GTO 02	128 XEQ 05	181 2
24 STO 23	76 STO 01	129 FS?C 02	182 X>Y?
25 20	77 XEQ 05	130 GTO 02	183 GTO 06
26 STO 25	78 FS?C 02	131 RCL 23	184*LBL 12
27 36	79 GTO 02	132 ST/ IND 25	185 RCL 02
28 STO 32	80 RCL 01	133 RCL 01	186 FRC
29 40	81 STO IND 25	134 ST+ IND 25	187 RCL 23
30 STO 33	82 RCL 00	135 E	188 *
31 46	83 X=0?	136 ST- 00	189 GTO 07
32 STO 34	84 GTO 01	137 RCL 00	190*LBL 06
33 52	85 TIME	138 X=0?	191 SF 02
34 STO 35	86 RN#	139 GTO 01	192*LBL 08
35 36.155	87 .5	140 GTO 10	193 RCL 21
36 " "	88 X>Y?	141*LBL 03	194 X<> 25
37 RCL b	89 SF 02	142 E	195 RTN
38 ASTO IND Y	90 E	143 ST+ 00	196*LBL 11
39 ISG Y	91 FS?C 02	144 RCL 25	197 X#Y?
40 STO b	92 RCL 22	145 RTN	198 GTO 12
41*LBL 01	93 STO 03	146*LBL 05	199 E
42 DSE 25	94 TIME	147 X<> 25	200 STO 09
43 9	95 RN#	148 STO 21	201 13
44 RCL 25	96 .5	149 20	202 RCL 25
45 X#Y?	97 X>Y?	150 STO 25	203 X<=Y?
46 GTO 02	98 SF 02	151*LBL 00	204 GTO 13
47 0	99 RCL 03	152 DSE 25	205 RCL 22
48 SETSW	100 FS?C 02	153 9	206 ST* 09
49 XROM "C"	101 CHS	154 RCL 25	207 16
50 CF 01	102 STO 03	155 X=Y?	208 RCL 25
51 55	103*LBL 10	156 GTO 08	209 X<=Y?
52 TFX	104 RCL 01	157 RCL 25	210 GTO 13
53 BEEP	105 RCL 03	158 RCL 21	211 RCL 22
	106 +	159 X=Y?	212 ST* 09

213 18	266 E	319 SF 21	372 FS? 04
214 RCL 25	267 X<Y?	320 CF 12	373 GTO 13
215 X<=Y?	268 GTO a	321 PWRUP	374 ADV
216 GTO 13	269 55	322 "MC 0 1 2 3 "	375 SF 12
217 RCL 22	270 TFX	323 >"4 5 6 7 8 9"	376 "T/SCORE="
218 ST* 09	271 56	324 PRA	377 ARCL 08
219*LBL 13	272 ST+ 29	325 XEQ 17	378 PRA
220 CLA	273 ST+ 30	326 9 E-3	379 ADV
221 RCL 09	274 ST+ 31	327 STO 27	380*LBL 04
222 ST+ 07	275 CLA	328 56.065	381 36.039
223 32	276 XEQ 03	329*LBL 16	382 "B"
224 ENTER^	277 ASTO IND 29	330 STO 28	383 XEQ 14
225 E3	278 ASTO IND 30	331 INT	384 40.045
226 XEQ 09	279 ASTO IND 31	332 56	385 "C"
227 33	280 0	333 -	386 XEQ 14
228 ENTER^	281 STO 03	334 CLA	387 46.051
229 E2	282 STO 07	335 X=0?	388 "T"
230 XEQ 09	283 STO 25	336 >"0"	389 XEQ 14
231 34	284 SF 00	337 ARCL X	390 52.055
232 ENTER^	285 RCL 04	338 >" "	391 "S"
233 E1	286 STO 01	339 ACA	392 XEQ 14
234 XEQ 09	287 XEQ 05	340 RCL 28	393 GTO 13
235 35	288 RCL 05	341 CLA	394*LBL 14
236 ENTER^	289 STO 01	342*LBL 15	395 ""
237 E	290 XEQ 05	343 ENTER^	396 SF 12
238*LBL 09	291 RCL 06	344 "X"	397 ACA
239 RCL 09	292 STO 01	345 ASTO X	398 CF 12
240 X#Y?	293 XEQ 05	346 RCL IND Y	399 " "
241 RTN	294 RCL 07	347 X=Y?	400 ACA
242 RDN	295 ST+ 08	348 GTO 14	401 CLA
243 XEQ 03	296 55	349 CLA	402 RCL b
244 RCL T	297 RASP	350 ARCL X	403 ARCL IND Y
245 RCL IND X	298*LBL e	351 ACA	404 >" "
246 ASTO IND X	299 SF 21	352 RCL Z	405 ISG Y
247 ISG IND Y	300 PWRUP	353 ISG X	406 STO b
248 RTN	301 SF 12	354 GTO 15	407 ACA
249 GTO 03	302 RCL 26	355 GTO 13	408 PRBUF
250*LBL "E"	303 BEEP	356*LBL 14	409 RTN
251 SF 21	304 "SALVO "	357 SF 12	410*LBL 13
252 PWRDN	305 ARCL X	358 ACA	411 FS?C 06
253 FS?C 04	306 PRA	359 CF 12	412 GTO 13
254 RUNSW	307 CLA	360 RCL Z	413 FS? 04
255 ISG 26	308 ARCL 04	361 ISG X	414 GTO 13
256 ""	309 >" "	362 GTO 15	415 ADV
257 STO 06	310 ARCL 05	363*LBL 13	416 SF 12
258 STO 31	311 >" "	364 ""	417 CLA
259 RDN	312 ARCL 06	365 ACA	418 RCLSW
260 STO 05	313 PRA	366 PRBUF	419 INT
261 STO 30	314 "S/SCORE="	367 E-2	420 ARCL X
262 RDN	315 ARCL 07	368 +	421 XEQ 14
263 STO 04	316 PRA	369 ISG 27	422 XEQ 14
264 STO 29	317 ADV	370 GTO 16	423 ACA
265 RCLSW	318*LBL 14	371 XEQ 17	424 ADV

425 RCLSW	464 XEQ 25	503 FRC	542 ARCL X
426 E	465 ASTO 43	504 E2	543 ARCL X
427 X<Y?	466 XEQ 09	505 *	544 ARCL X
428 GTO a	467 ASTO 44	506 INT	545 PRA
429 RCL 08	468 XEQ 09	507 GTO 14	546 RTN
430 4664	469 ASTO 45	508*LBL 09	547*LBL "C"
431 X<=Y?	470 16	509 LASTX	548 SF 01
432 GTO a	471 XEQ 25	510 FRC	549 56
433 RCL 26	472 ASTO 46	511 E2	550 +
434 20	473 XEQ 09	512 *	551 "X"
435 X>Y?	474 ASTO 47	513 INT	552 ASTO IND X
436 GTO 13	475 15	514 GTO 09	553 FS? 04
437*LBL a	476 XEQ 25	515*LBL "B"	554 RTN
438 SF 21	477 ASTO 48	516 SF 01	555 GTO 13
439 ADV	478 XEQ 09	517 56	556*LBL "D"
440 STOPSW	479 ASTO 49	518 +	557 SF 01
441 BEEP	480 14	519 X<>Y	558 56
442 SF 12	481 XEQ 25	520 XEQ 09	559 +
443 "SHIPS"	482 ASTO 50	521 ASTO IND Z	560 " "
444 PRA	483 XEQ 09	522 GTO 13	561 ASTO IND X
445 ADV	484 ASTO 51	523*LBL 03	562*LBL 13
446 CF 12	485 13	524 E1	563 CF 22
447 19	486 XEQ 25	525 RCL 26	564 FC? 01
448 XEQ 25	487 ASTO 52	526 GTO 14	565 ADV
449 ASTO 36	488 12	527*LBL 25	566 ADV
450 XEQ 09	489 XEQ 25	528 RCL IND X	567 "SALVO "
451 ASTO 37	490 ASTO 53	529 INT	568 RCL 26
452 XEQ 09	491 11	530*LBL 09	569 E
453 ASTO 38	492 XEQ 25	531 CLA	570 +
454 XEQ 09	493 ASTO 54	532 E1	571 ARCL X
455 ASTO 39	494 E1	533 X<>Y	572 "> ***"
456 18	495 XEQ 25	534*LBL 14	573 CLAXON
457 XEQ 25	496 ASTO 55	535 X<Y?	574 CF 21
458 ASTO 40	497 SF 06	536 ">0"	575 AVIEW
459 XEQ 09	498 GTO 04	537 ARCL X	576 FC?C 01
460 ASTO 41	499*LBL 14	538 RTN	577 PWRDN
461 XEQ 09	500 "':" "	539*LBL 17	578 END
462 ASTO 42	501 E1	540 ""	
463 17	502 LASTX	541 ASTO X	

True Battleship

[Luis Gasparini – UPL #00796C](#)

This program replaces one player in the battleship game. It places its ships in a different way with different seeds (!) and then plays against you.

The battleship game is played this way:

Each player has two sea boards of 10 by 10 squares. One is his own, the other his opponent's. In his own sea board, at the beginning of the game he has to place all of his 10-ship fleet. In the other sea board he puts the ships of the opponent during the game as he finds them by making shots, one at a time.

How many ships in the fleet and how are they to be placed?

First, there are 10 ships as follows: 1 of 4 squares (cruiser) , 2 of 3 squares (frigate) , 3 of 2 (torpedo ships), and 4 of 1 square (submarines). All of them are linear, not forming any diagonal angles.

Second, the ships can't be placed so that they are aside. With no blank spaces between them (nor by sides neither by corners). For example, if a ship of two squares is placed in squares 77 and 78, all squares surrounding those must be left blank and can't be occupied by any other part of a ship (i.e. squares 66, 67, 68, 69, 79, 89, 88, 87, 86 and 76).

When the two players have finished placing their own fleets one player begins and shoots first. The last shot will be done by the other player. The final objective is to sink all the opponent's ships before he sinks ours.

The program will take about 8 to 9 minutes to place its fleet in the sea board. Then the program will ask you if you want to start or defer the honors to the calculator, answering YES or NO to the question "YOU BEGIN?"

A player shoots by indicating where he's aiming, as Y:X coordinate, i.e. place "74" is the 7th row and 4th column. Then the other layer has to answer, with the following three possibilities:

1. The shot hit no ship, the answer is "WATER"
2. The shot hit a part of a ship, but the ship still has other squares not hit – so it isn't sunk yet. The answer is "HIT"
3. The shot hits the last part of a ship that was remaining untouched, and therefore it sinks. The answer is "SUNK"

Once answered is the other player's turn to shoot

Program listing:

<u>01*LBL "BSHIP"</u>	50 .009	99 RNG	148 XEQ 98
02 52	51 +	100 .5	149 X#0?
03 CF 21	52 STO 51	101 X>Y?	150 GTO 14
04 XROM "INIT"	<u>53*LBL b</u>	102 SF 05	151 E
05 FC? 55	54 RCL IND 50	<u>103*LBL 09</u>	152 ST+ 35
06 SF 21	55 RCL 51	104 E	153 ST+ 23
07 SF 27	56 INT	105 STO 35	154 RCL 41
08 4	57 E	<u>106*LBL 10</u>	155 STO IND 23
09 STO 22	58 +	107 RCL 35	156 STO 37
<u>10*LBL 07</u>	59 10^X	108 RCL 22	157 GTO 10
11 5	60 /	109 X=Y?	<u>158*LBL 14</u>
12 RCL 22	61 FRC	110 RTN	159 FS?C 06
13 -	62 E1	<u>111*LBL 11</u>	160 GTO 17
14 E3/E+	63 *	112 RCL 37	161 E
15 STO 33	64 INT	113 E1	162 ST+ 34
<u>16*LBL 06</u>	65 X=0?	114 /	163 SF 06
17 XEQ 95	66 GTO a	115 INT	164 RCL 36
18 XEQ 97	67 ISG 51	116 STO 38	165 STO 37
19 ISG 33	68 GTO b	117 LASTX	166 GTO 11
20 GTO 06	69 RCL 50	118 -	<u>167*LBL 17</u>
21 DSE 22	70 E	119 CHS	168 FS?C 07
22 GTO 07	71 +	120 E1	169 GTO 08
23 BEEP	72 E1	121 *	170 SF 07
24 GTO 15	73 MOD	122 STO 39	171 FS? 05
<u>25*LBL 95</u>	74 STO 50	123 FS? 05	172 GTO 20
26 RNG	75 CLX	124 GTO 12	173 SF 05
27 E2	76 GTO c	125 38	174 GTO 22
28 *	<u>77*LBL a</u>	126 GTO 13	<u>175*LBL 20</u>
29 INT	78 RCL 50	127*LBL 12	176 CF 05
30 STO 36	79 E1	128 39	177*LBL 22
31 XEQ 98	80 *	<u>129*LBL 13</u>	178 24
32 X=0?	81 RCL 51	130 STO 40	179 STO 23
33 GTO 04	82 INT	131 RCL IND 40	180 RCL 36
34*LBL 08	83 +	132 - E	181 STO 37
35 RCL 36	84 STO 36	133 RCL 34	182 GTO 09
36 E	85*LBL 04	134 Y^X	183*LBL 97
37 +	86 RCL 36	135 +	184 RCL 22
38 E2	87 STO 37	136 STO IND 40	185 23
39 MOD	88 24	137 X<0?	186 +
40 E1	89 STO 23	138 GTO 14	187 E3/E+
41 /	90 RDN	139 9	188 23
42 INT	91 STO IND 23	140 X<Y?	189 +
43 STO 50	92 CF 05	141 GTO 14	190 STO 23
44 LASTX	93 CF 06	142 RCL 38	<u>191*LBL 05</u>
45 -	94 CF 07	143 E1	192 RCL 22
46 CHS	95 RNG	144 *	193 RCL IND 23
47 E1	96 ST+ X	145 RCL 39	194 XEQ 99
48 *	97 INT	146 +	195 ISG 23
<u>49*LBL c</u>	98 STO 34	147 STO 41	196 GTO 05

197 RCL 23	250 +	303 STO 22	356 RCL 42
198 FRC	251 STO 41	304 4	357 RCL 33
199 24	252 XEQ 98	305 STO 33	358 E
200 +	253 X#0?	306 CF 00	359 -
201 STO 23	254 GTO 03	307 "YOU FIRST?"	360 10^X
<u>202*LBL 19</u>	255 5	308 AON	361 /
203 RCL IND 23	256 RCL 41	309 PROMPT	362 INT
204 E1	257 XEQ 99	310 AOFF	363 X#0?
205 /	258*LBL 03	311 FS?C 23	364 GTO 26
206 INT	259 ISG 31	312 GTO 50	365 DSE 33
207 STO 30	260 GTO 02	313 SF 00	366 GTO 25
208 LASTX	261*LBL 01	314 7P<>S	<u>367*LBL 26</u>
209 -	262 ISG 28	315 GTO 18	368 XEQ 97
210 CHS	263 GTO 00	316*LBL 50	369 E
211 E1	264 ISG 23	317 7P<>S	370 STO 22
212 *	265 GTO 19	318 "R&C="	371 GTO 18
213 STO 29	266 RTN	319 RCL 22	<u>372*LBL 24</u>
214 3	<u>267*LBL 98</u>	320 E	373 RCL 20
215 E3/E+	268 E1	321 X=Y?	374 FRC
216 STO 28	269 /	322 GTO 21	375 X=0?
<u>217*LBL 00</u>	270 RCL IND X	323 ARCL 35	376 GTO 16
218 RCL 28	271 X<>Y	324 TONE 0	377 E2
219 INT	272 FRC	325 PROMPT	378 *
220 RCL 29	273 E1	326*LBL 21	379 E
221 +	274 *	327 RCL 33	380 X=Y?
222 2	275 10^X	328 STO 22	381 FS? 00
223 -	276 /	329 XEQ 95	382 GTO 27
224 X<0?	277 INT	330 E	383 GTO 18
225 GTO 01	278 E1	331 STO 22	<u>384*LBL 16</u>
226 9	279 /	332 ARCL 24	385 "A TIE"
227 X<Y?	280 FRC	333 TONE 0	386 GTO 28
228 GTO 01	281 E1	334 PROMPT	<u>387*LBL 27</u>
229 RDN	282 *	<u>335*LBL C</u>	388 "I WON"
230 STO 32	283 RTN	336 E	389 AVIEW
231 3	<u>284*LBL 99</u>	337 RCL 22	390 TONE 6
232 E3/E+	285 E1	338 X=Y?	391 TONE 6
233 STO 31	286 /	339 GTO 23	392 AVIEW
234*LBL 02	287 INT	340 23	393 PSE
235 RCL 31	288 X<>Y	341 +	394 GTO 28
236 INT	289 LASTX	342 RCL 35	<u>395*LBL A</u>
237 RCL 30	290 FRC	343 STO IND Y	396 RCL 20
238 +	291 E1	344 RCL 22	397 FRC
239 2	292 *	<u>345*LBL 23</u>	398 X=0?
240 -	293 10^X	346 E	399 GTO 31
241 X<0?	294 *	347 -	400 RCL 22
242 GTO 03	295 ST+ IND Y	348 10^X	401 E
243 9	296 RTN	349 ST- 42	402 X=Y?
244 X<Y?	<u>297*LBL 15</u>	350 E	403 GTO 32
245 GTO 03	298 1234	351 ST- 20	404 5
246 RDN	299 STO 42	352 RCL 42	405 RCL 35
247 E1	300 20.2	353 X=0?	406 XEQ 99
248 *	301 STO 20	354 GTO 24	407 GTO 33
249 RCL 32	302 E	<u>355*LBL 25</u>	408*LBL 32

409 5	462 38	515*LBL 18	567 +
410 RCL 24	463 GTO 30	516 7P<>S	568 X<0?
411 XEQ 99	464*LBL 29	517 TONE 3	569 GTO 44
412 GTO 18	465 39	518 TONE 5	570 9
413*LBL 31	466*LBL 30	519 "SHOOT (R&C)"	571 X<Y?
414 "YOU WON"	467 STO 40	520 PROMPT	572 GTO 44
415 GTO 28	468 RCL IND 40	521 STO 41	573 X<>Y
<u>416*LBL B</u>	469 -1	522 XEQ 98	574 STO IND 46
417 E	470 RCL 34	523 STO 47	575 RCL 43
418 ST+ 22	471 Y^X	524 5	576 E1
419 ST- 20	472 +	525 X<>Y	577 *
420 2	473 X<0?	526 X#Y?	578 RCL 44
421 RCL 22	474 GTO 33	527 X=0?	579 +
422 X=Y?	475 9	528 GTO 39	580 STO 48
423 GTO 35	476 X<Y?	529 6	581 XEQ 98
424 22	477 GTO 33	530 X<>Y	582 5
425 +	478 RDN	531 X=Y?	583 X<>Y
426 RCL 35	479 STO IND 40	532 GTO 48	584 X=Y?
427 STO IND Y	480 RCL 38	533 .01	585 GTO 44
428 GTO 36	481 E1	534 ST- 20	586 6
<u>429*LBL 35</u>	482 *	535 X<>Y	587 X=Y?
430 CF 05	483 RCL 39	536 E	588 GTO 46
431 CF 06	484 +	537 X=Y?	589 TONE 9
432 RCL 24	485 XEQ 98	538 GTO 40	590 "HIT"
433 E1	486 5	539 SF 08	591 AVIEW
434 /	487 X=Y?	540 CF 09	592 RCL 47
435 INT	488 GTO 33	541 CF 10	593 -
436 RCL 25	489 RCL 38	<u>542*LBL 41</u>	594 RCL 41
437 E1	490 E0	543 RCL 41	595 XEQ 99
438 /	491 *	544 E1	596 GTO 50
439 INT	492 RCL 39	545 /	<u>597*LBL 39</u>
440 X=Y?	493 +	546 INT	598 RASP
441 SF 05	494 STO 35	547 STO 43	599 "WATER"
442 RNG	495 GTO 18	548 LASTX	600 AVIEW
443 ST+ X	<u>496*LBL 34</u>	549 -	601 RCL 20
444 INT	497 RCL 24	550 CHS	602 INT
445 STO 34	498 STO 35	551 E1	603 X=0?
446*LBL 34	499 GTO 35	552 *	604 GTO 27
447 RCL 24	<u>500*LBL 33</u>	553 STO 44	605 GTO 50
448 STO 35	501 FS?C 06	<u>554*LBL 47</u>	<u>606*LBL 40</u>
<u>449*LBL 36</u>	502 GTO 37	555 FS? 08	607 TONE 9
450 E1	503 SF 06	556 GTO 42	608 "SUNK"
451 /	504 E	557 43	609 AVIEW
452 INT	505 ST+ 34	558 GTO 43	610 6
453 STO 38	506 GTO 34	<u>559*LBL 42</u>	611 RCL 47
454 LASTX	<u>507*LBL 37</u>	560 44	612 -
455 -	508 FS? 05	<u>561*LBL 43</u>	613 RCL 41
456 CHS	509 GTO 38	562 STO 46	614 XEQ 99
457 E1	510 SF 05	563 RCL IND 46	615 RCL 20
458 *	511 GTO 34	564 - E	616 FRC
459 STO 39	<u>512*LBL 38</u>	565 RCL 45	617 X#0?
460 FS? 05	513 CF 05	566 Y^X	618 GTO 50
461 GTO 29	514 GTO 34		619 RCL 20

620 INT	632 ST+ 45	644 /	656 CLAXON
621 X=0?	633 SF 09	645 INT	657 AVIEW
622 GTO 16	634 GTO 41	646 STO 43	658 GTO 50
623 E	<u>635*LBL 45</u>	647 LASTX	<u>659*LBL 28</u>
624 FS? 00	636 FS?C 10	648 -	660 AVIEW
625 X#Y?	637 GTO 40	649 CHS	661 PSE
626 GTO 31	638 SF 10	650 E1	662 PSE
627 GTO 50	639 CF 08	651 *	663 "GAME OVER"
<u>628*LBL 44</u>	640 GTO 41	652 STO 44	664 PROMPT
629 FS?C 09	<u>641*LBL 46</u>	653 GTO 47	665 END
630 GTO 45	642 RCL 48	<u>654*LBL 48</u>	
631 E	643 E1	655 "BAD SHOT"	

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4			X							
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										

Naval Battle

F. Javier Chamorro Pagani – Electro1, N1 p106 ; (May 1982)

Overview

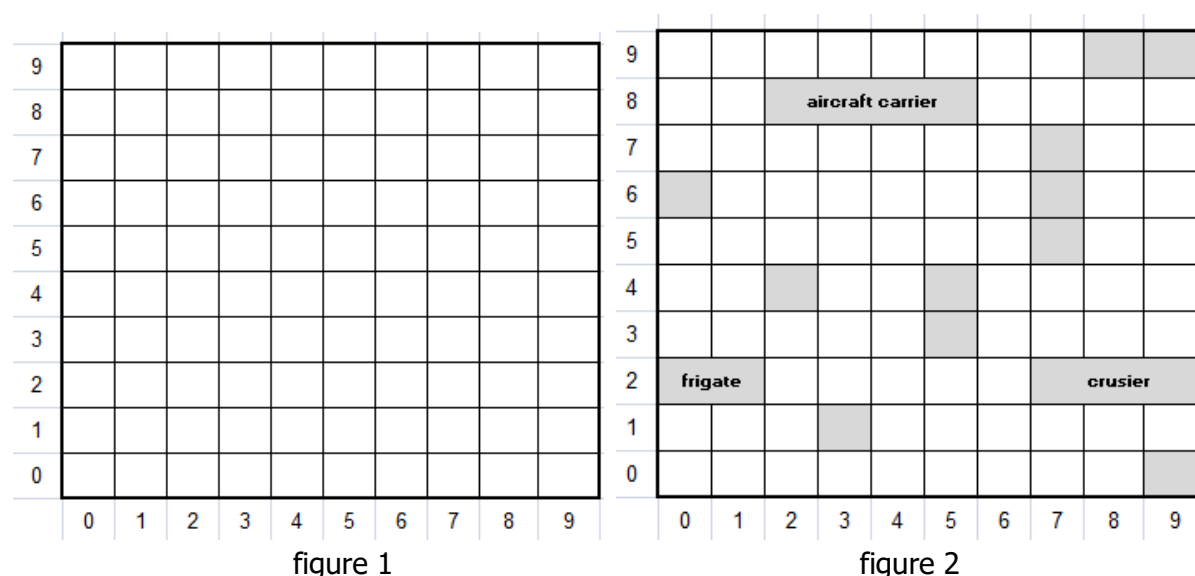
The popular and well-known naval battle game surely was, is and will be played by many students to spend time in classes of subjects they are not attracted to. The problem, however, was not finding an opponent ready for the game. I hope that this article can solve the problem to the HP-41C/V owners.

The game of naval battle requires two players, each representing a fleet of a belligerent nation "X". Each fleet consists of ten ships: one aircraft carrier, two cruisers, three frigates and four torpedo boats. Each of the fleets simultaneously warns of the presence of its enemy and, by communicating the news to the high hands, receives the order to sink the enemy at any cost. The objective of the game is to sink the opponent's entire fleet, so the first to do so wins the game.

Game rules

Before we turn to the instructions for playing the HP-41, let's review the rules of the naval battle game.

First of all, each player must build two 10x10 boards, which must be numbered on their left lower edges from 0 to 9, both horizontally (from left to right) and vertically (from bottom to top) respectively. An example of this board is shown in figure 1.



The first board will represent the space where the enemy fleet will be located; the second the space where each player must place his own fleet as follows:

- Each ship is represented by squares as shown in figure 2, i.e. the aircraft carrier, cruisers, frigates and torpedo boats are equivalent to four, three, two and one squares respectively in vertical or horizontal succession indistinctly, but never in diagonal succession.

- The ten ships will be distributed in the second board, that is to say in the "own space" in a totally random way and separated each one of them by a box as minimum with respect to the others.

In this way the game can be started by drawing lots to see who starts the game, and the throws will be made one square at a time and one turn at a time per player, unless the result of the throw was hit or sunk, which means you have the right to throw again until your throw is "agua", i.e. in the water.

But, how do you shoot a certain square? The ruler will be "shot to the square YX, where Y will be the value taken by the square on the vertical axis, and X will be the value taken by the square on the horizontal axis. Both Y and X will be integers with values between 0 and 9 inclusive, in such a way that the number YX will have the following definition domain:
 $00 \leq YX \leq 99$, any other number will give "water" as a result.

So that the positioning of squares is clear, let's study where the fleet of the board of figure 2 is: the aircraft carrier will have as coordinates the numbers 82, 83, 84, 85; the cruisers 77, 67, 57, and 27; the frigates 98, 99; 45, 35; and 20, 21; and the torpedo boats 60; 42; 13; and 09.

Evidently it is clear that neither of the two players can see the board of the opponent, because otherwise the game would lose all its notion of being and would be like playing the cat and mouse when the cat had already eaten it.

Finally, the player who first sinks all the enemy ships will have won the naval battle.

BATNAV

Now we are asked the question of how to play with the 41: very simple, thanks to the program BATNAZ that allows us to consider the calculator as if it were the second player.

The calculator will need considerable time to "think" about the positions of her boats, which it "thinks" in a certain order: first the aircraft carrier, then the cruisers, then the frigates and the torpedo boats at the end. By typing XEQ "BATNAV" the program shows "SEED?", so the machine is asking for a seed between 10×10^{-36} and 17453292.45 - due to the structure of the subroutine generating random numbers (LBL 16) of the program the seed should not be a multiple of pi either.

The machine takes this number and transforms it into a number between 00 and 99, which will be the first box of the ship it has to locate and stores it in a memory register; it again uses the generator to choose the direction that the succession of boxes (to the right, left, up or down) of the ship to be located must follow until the entire ship is completed, provided that this new ship generated is at a distance box separate from all previous ones, and if it is not so it returns to the generator to choose another position repeating the process, and so on until it places the entire fleet.

The program takes a while to place its fleet which is given according to the seed introduced and which is unpredictable, since for each seed the program acts differently. In general the

time of preparation varies between 10 and 20 minutes, with an average of 14 minutes approx (this time can be used to build the boards and position the player's fleet).

But don't worry; this is the only time the machine takes for its internal process since the "attack" itself is fast, except in the last throws when the board is very full of previous throws.

The machine warns us when it has finished "thinking" the positions of its ships showing the display : "HP-41C LISTA".

Then it draws heads or tails who begins to attack, so there are two possibilities in which the display will show:

1 "TU EMPIEZAS", you start.

Then press R/S and we receive "CASILLA?". So we can enter the coordinates of the chosen square of the enemy space and press R/S. The machine will answer to us:

a) "YX AGUA", then the player will attack us with "TIRO AL YX" answering with "1", "2", or "3" if the shot that the machine has taken has been water, hit or sunk respectively; any other answer will derive from the machine answering us "ERRONEO" and immediately repeats his question "TIRO AT YX" (shoot at...).

b) "YX TOCADO", followed by "TIRAS TU" until the throw fails.

c) "1/0 HUNDIDO", followed by "TIRAS TU" until the throw fails.

As the game progresses, it may happen that the display shows "1/0 HUNDIDO" followed by two BEEPs and "TU GANAS", so the player has won the game.

2, "EMPIEZO YO", the machine starts.

Press R/S and it shows "TIRO AL YX", to which we'll respond as in section 1a). If it has hit us or if it has sunk a ship, the machine will shoot again. If the shot has been made in water, we will proceed as in section 1) but now it will show "TIRAS TU". Contrary to section 1c) the machine may hit us or sink the last boat so that after responding properly will present "GANA HP41". The machine has won the game!

BATNAV will always give sincere answers, so it is to be expected that the player behaves in the same way, although it is possible that he is mistaken in some answer having pressed R/S and not being able to go back. BATNAV only notices some mistakes on the player's part (not described here), but I can anticipate that he will "notice" a mistake if it has enough data to know i

Entering the program.

Once entered in the machine, BATNAV occupies about 1290 bytes, or 184 program registers; it uses a SIZE = 135, that is a total of 319 memory registers plus two bytes; considering that

only two more bytes are available, I advise you to do a master clear before introducing the program in the HP41.

Those who are not familiar with synthetic programming will be surprised to see program lines such as STO N, DSE c, etc. Synthetic programming opens a wide field of possibilities in the HP-41 not described in the user manual, but in the PPC and in many books published in the USA, such as SP by W.C. Wickes.

The hexadecimal codes of the synthetic tones are indicated below:

step 209: 79; step 285: 35; step 313: 0E; step 382: 1A
step 414: 7E; step 450: 49; step 536: 74; step 537: 63

Program listing:

01 LBL "BATNAV"	35 RCL 10	69 FC? 06	103 99
02 135	36 STO 33	70 GTO 07	104 FS? 09
03 -ARMADA INV	<u>37 LBL 38</u>	71 RTN	105 GTO 05
04 5	38 RCL 33	<u>72 LBL 16</u>	106 X<Y?
05 STO 03	39 ST/ IND 30	73 LEFT	107 RTN
06 "SEMILLA=?"	40 DSE 30	74 RCL 00	<u>108 LBL 13</u>
07 PROMPT	41 GTO 38	75 R-D	109 FS? 03
08 STO 00	42 GTO 45	76 FRC	110 SF 05
09 CF 22	<u>43 LBL 09</u>	77 STO 00	111 CF 03
10 3	44 E	78 E2	112 FS? 06
11 STO \	45 ST- \	79 *	113 RTN
12 30	46 ST- 27	80 INT	114 XEQ 08
13 STO]	47 10^X	81 RTN	115 RTN
14 4	48 STO 32	<u>82 LBL 06</u>	<u>116 LBL 12</u>
15 STO [49 ST-]	83 XEQ 16	117 RCL 01
16 5,001	50 RTN	84 50	118 FS? 10
17 STO 27	<u>51 LBL 07</u>	85 X<Y?	119 RCL]
18 GOOSE	52 CF 03	86 SF 08	120 FS? 09
<u>19 LBL 44</u>	53 CF 01	87 FC? 08	121 RCL \
20 LEFT	54 XEQ 33	88 SF 07	122 -
21 E	55 XEQ 16	<u>89 LBL 36</u>	123 FS? 10
22 ST+ 34	56 STO 01	90 FS? 08	124 ,
23 RCL 34	<u>57 LBL 31</u>	91 XEQ 03	125 FS? 09
24 STO 31	58 SF 11	92 FS? 07	126 GTO 05
25 LBL 43	59 XEQ 16	93 XEQ 12	127 X>Y?
26 LEFT	60 50	94 RTN	128 RTN
27 XEQ 07	61 X<Y?	<u>95 LBL 03</u>	<u>129 LBL 14</u>
28 DSE 31	62 SF 10	96 RCL 01	130 FS? 03
29 GTO 43	63 FC? 10	97 FS? 10	131 SF 05
30 XEQ 09	64 SF 09	98 RCL]	132 CF 03
31 DSE [65 FS? 10	99 FS? 09	133 FS? 06
32 GTO 44	66 XEQ 06	100 RCL \	134 RTN
33 24,004	67 FS? 09	101 +	135 XEQ 08
34 STO 30	68 XEQ 06	102 FS? 10	136 RTN

<u>137 LBL 05</u>	190 GTO 11	243 X=Y?	296 RND
138 X<0?	191 11	244 RTN	297 RCL [
139 RTN	192 XEQ 10	245 -	298 X=Y?
140 E1	193 X=Y?	246 ST- 03	299 GTO 03
141 /	194 GTO 11	247 RTN	300 ,1
142 INT	<u>195 LBL 03</u>	<u>248 LBL 10</u>	301 +
143 ENTER^	196 RCL 01	249 RCL 01	302 X=Y?
144 RCL 01	197 E1	250 RCL Y	303 GTO 05
145 E1	198 MOD	251 +	304 DSE 27
146 /	199 X=0?	252 RCL IND 29	305 GTO 17
147 INT	200 GTO 03	253 RTN	<u>306 LBL 20</u>
148 RCL Y	201 -1	<u>254 LBL 45</u>	307 XEQ 42
149 X#Y?	202 XEQ 10	255 "HP41 LISTA"	308 >" AGUA"
150 RTN	203 X=Y?	256 AVIEW	309 9
151 FS? 08	204 GTO 11	257 50	310 XEQ 41
152 GTO 13	205 9	<u>258 LBL 40</u>	311 AVIEW
153 FS? 07	206 XEQ 10	259 TONE 9	312 RTN
154 GTO 14	207 X=Y?	260 PSE	<u>313 LBL 05</u>
155 RTN	208 GTO 11	261 DSE X	314 "YA TOCADO"
<u>156 LBL 08</u>	209 -11	262 GTO 40	315 AVIEW
157 RCL 27	210 XEQ 10	263 FC?C 22	316 TONE 4
158 STO 02	211 X=Y?	264 OFF	317 ,
159 RCL 03	212 GTO 11	265 ,	318 STO 02
160 STO 25	213 LBL 03	266 STO 25	319 RTN
<u>161 LBL 26</u>	214 DSE 29	267 STO 34	<u>320 LBL 03</u>
162 RCL 25	215 DSE 04	268 XEQ 16	321 FRC
163 STO 29	216 GTO 04	269 50	322 ,2
164 DSE 29	217 RCL 01	270 X<Y?	323 X=Y?
165 5 E-3	218 STO IND 03	271 SF 01	324 GTO 20
166 +	<u>219 LBL 35</u>	272 "EMPIEZ"	325 4
167 STO 04	220 FS? 03	273 FC? 01	326 STO \
<u>168 LBL 04</u>	221 RCL 01	274 >"AS TU"	327 25
169 E1	222 E	275 FS? 01	328 STO 01
170 XEQ 10	223 FS? 07	276 >"O YO"	329 E
171 X=Y?	224 CHS	277 PROMPT	330 STO 02
172 GTO 11	225 FS? 10	<u>278 LBL 18</u>	<u>331 LBL 15</u>
173 -10	226 E1	279 FC?C 01	332 RCL \
174 XEQ 10	227 FS? 09	280 XEQ 02	333 STO]
175 X=Y?	228 E	281 FIX 0	<u>334 LBL 21</u>
176 GTO 11	229 *	282 XEQ 24	335 RCL 02
177 RCL 01	230 +	283 GTO 18	336 ST- 01
178 E1	231 STO 01	<u>284 LBL 02</u>	337 RCL 27
179 MOD	232 FS? 03	285 "CASILLA=?"	338 INT
180 9	233 RTN	286 TONE 3	339 STO 27
181 X=Y?	234 E	287 PROMPT	340 RCL 01
182 GTO 03	235 ST+ 03	288 STO [341 X<=Y?
183 -9	236 DSE 02	289 24,004	342 GTO 03
184 XEQ 10	237 GTO 26	290 STO 27	343 DSE]
185 X=Y?	238 SF 06	<u>291 LBL 17</u>	344 GTO 21
186 GTO 11	239 RTN	292 RCL IND 27	345 E
187 E	<u>240 LBL 11</u>	293 RCL 33	346 ST+ 02
188 XEQ 10	241 RCL 27	294 *	347 DSE \
189 X=Y?	242 RCL 02	295 FIX 1	348 GTO 15

<u>349 LBL 03</u>	401 GTO 06	454 PROMPT	507 FS? 04
350 ,1	402 ,	455 E	508 GTO 00
351 RCL 33	403 STO 02	456 STO \	509 SF 03
352 /	404 RTN	457 10^X	510 RTN
353 ST+ IND 27	<u>405 LBL 06</u>	458 STO]	<u>511 LBL 00</u>
354 RCL 01	406 "GANAS TU"	459 X<>Y	512 ,027
355 STO]	407 BEEP	460 E	513 ST+ 04
356 RCL 02	408 BEEP	461 X>Y?	514 RCL IND 04
357 +	409 PROMPT	462 GTO 09	515 3
358 E3	410 RTN	463 RDN	516 STO IND Y
359 /	<u>411 LBL 03</u>	464 ENTER^	517 DSE 04
360 RCL 01	412 XEQ 42	465 FRC	518 GTO 00
361 +	413 >" TOCADO"	466 X#0?	519 CF 04
362 E	414 9	467 GTO 09	520 CF 03
363 +	415 XEQ 41	468 RDN	521 PI
364 STO 03	416 AVIEW	469 3	522 STO 28
<u>365 LBL 22</u>	417 TONE 2	470 X<Y?	523 RCL 02
366 RCL IND]	418 PSE	471 GTO 09	524 7
367 RCL 33	419 SF 01	472 RDN	525 X=Y?
368 *	420 RTN	473 2	526 SF 01
369 RND	<u>421 LBL 24</u>	474 X=Y?	527 ,
370 FRC	422 E	475 GTO 06	528 STO 02
371 ,1	423 STO \	476 RDN	529 DSE 32
372 X#Y?	424 10^X	477 3	530 RTN
373 GTO 03	425 STO]	478 X=Y?	531 "GANA HP41"
374 E	426 FS?C 01	479 SF 04	532 TONE 0
375 ST+]	427 RTN	480 X=Y?	533 TONE 0
376 ISG 03	428 SF 01	481 GTO 06	534 PROMPT
377 GTO 22	429 FC? 03	482 RCL 02	535 RTN
378 E	430 XEQ 33	483 7	<u>536 LBL 09</u>
379 ST+ 25	431 FS? 03	484 X=Y?	537 "ERRONEO"
380 CLA	432 GTO 19	485 GTO 39	538 AVIEW
381 FIX 0	433 28	486 RDN	539 TONE 6
382 ARCL 25	434 STO 04	487 RDN	540 TONE 9
383 >"^0	435 XEQ 16	488 CF 01	541 PSE
HUNDIDO"	436 STO 01	489 FS? 03	542 GTO 27
384 AVIEW	437 34	490 SF 00	<u>543 LBL 19</u>
385 TONE 6	438 +	<u>491 LBL 25</u>	544 RCL 02
386 SF 01	439 RCL IND X	492 RCL 01	545 7
387 RCL 03	440 X#0?	493 34	546 X=Y?
388 RCL 02	441 CF 01	494 +	547 GTO 39
389 -	442 X#0?	495 X<> Z	548 SF 06
390 STO 26	443 GTO 24	496 STO IND Z	549 RCL IND 04
<u>391 LBL 23</u>	444 GTO 29	497 FS? 00	550 34
392 ,1	<u>445 LBL 27</u>	498 GTO 28	551 -
393 ST+ IND 01	446 RCL 01	499 RTN	552 STO 01
394 E	447 STO 03	<u>500 LBL 06</u>	553 FC? 05
395 ST+ 01	<u>448 LBL 39</u>	501 E	554 XEQ 31
396 ISG 26	449 RCL 03	502 FS? 03	555 FS? 05
397 GTO 23	450 STO 01	503 ST+ 04	556 CF 03
398 RCL 25	451 "TIRO AL "	504 XEQ 25	557 FS? 03
399 E1	452 ARCL 01	505 RCL Z	558 GTO 30
400 X=Y?	453 TONE 3	506 STO IND 04	559 SF 03

560 XEQ 36	593 -11	626 RTN	659 GTO 12
561 FS? 03	594 XEQ 01	<u>627 LBL 28</u>	<u>660 LBL 33</u>
562 GTO 30	<u>595 LBL 03</u>	628 FC?C 00	661 CF 10
563 SF 03	596 RCL 01	629 RTN	662 CF 09
564 XEQ 35	597 34	<u>630 LBL 30</u>	663 CF 08
565 GTO 29	598 +	631 28	664 CF 07
<u>566 LBL 29</u>	599 RCL IND X	632 STO 04	665 CF 06
567 E1	600 X=0?	633 E	666 CF 05
568 XEQ 01	601 GTO 27	634 ST+ 02	667 CF 04
569 -10	602 SF 00	635 8	668 CF 02
570 XEQ 01	603 GTO 28	636 7	669 RTN
571 RCL 01	<u>604 LBL 01</u>	637 FS? 02	<u>670 LBL 12</u>
572 E1	605 RCL 01	638 E1	671 XEQ 36
573 MOD	606 +	639 FS? 02	672 FS? 03
574 9	607 X<0?	640 9	673 GTO 30
575 X=Y?	608 RTN	641 FC?C 02	674 SF 03
576 GTO 03	609 99	642 SF 02	675 RTN
577 -9	610 X<Y?	643 FS? IND X	<u>676 LBL 42</u>
578 XEQ 01	611 RTN	644 GTO 05	677 ,
579 E	612 RDN	645 FS? IND Y	678 STO 02
580 XEQ 01	613 34	646 GTO 08	679 RCL [
581 11	614 +	647 RTN	680 CLA
582 XEQ 01	615 RCL IND 04	<u>648 LBL 05</u>	681 FIX 0
<u>583 LBL 03</u>	616 X=Y?	649 FS? IND X	682 ARCL X
584 RCL 01	617 RTN	650 SF IND Y	683 RTN
585 E1	618 RCL IND Y	651 FS? IND Y	<u>684 LBL 41</u>
586 MOD	619 E	652 CF IND X	685 TONE IND X
587 X=0?	620 X<Y?	653 GTO 12	686 DSE X
588 GTO 03	621 GTO 10	<u>654 LBL 08</u>	687 GTO 41
589 -1	622 RTN	655 FS? IND Y	688 END
590 XEQ 01	<u>623 LBL 10</u>	656 SF IND X	
591 9	624 ENTER^	657 FS? IND X	
592 XEQ 01	625 XEQ 25	658 CF IND Y	

Navy War (German)

Burkhard Oerttel; HP-41 Sammlung book

Ship sinking against the HP-41

In this electronic version of the well-known student boredom game, the HP-41 proves to be a serious opponent who constantly changes his interrogation strategy and comments on both hits and "tickets" in a humorous way.

The program consists of four parts and an ASCII file. On the mass memory these sub-programs have the names "NAVY" to "NAVY3" for better identification and the ASCII file is called "CNAVY".

The Program "ADMIRAL" serves to facilitate the loading of all these program parts, which should be saved with flag 11 set so that it starts automatically. Before calling "ADMIRAL", 125 registers must be available in the main memory and 202 registers in the extended memory.

The segmentation into several parts was carried out, since the separation into construction phase and game phase offers two advantages with this program. Essentially: On the one hand, only the currently required program part occupies main memory, so that memory space can be saved; on the other hand, this facilitates division of the jump back from the joint subroutines, since on the calling programs identical trademarks can be used.

Now to the individual parts of the program:

"ADMIRAL" serves to load the other parts of the mass storage into the memory.

Unlike the other parts, "NAVY" always remains in the main memory and calls them.

"N", mass storage name: "NAVY1" contains subprograms, which are used by the program parts "O" and "P".

Synthetic commands:

Line 44: "LBL F1, 23" (hexadecimal), i.e. 241, 35 in decimal notation.

Line 52: "LBL F1,0A" (hex), i.e. 241, 10 in decimal.

The synthetic "XEQ" commands contained in "O" and "P" multiple times refer to these two.

"O", mass storage name: "NAVY3": With this program part the computer builds up its playing field and makes all further preparations.

Synthetic text line: Line 63: 246, 0, 16, 0, 33. 0, 129.

"P", mass storage name: "NAVY2" is the main program.

The "TONE" commands are arbitrary except for those in the following lines:

Lines 141 143: Tone 57

Lines 144 146: Tone 09

Lines 147 149: Tone 57

Line 314 Tone 26

Synthetic texts:

Line 40: 250, 1, 8, 9, 10, 11, 12, 18, 19, 21, 22

Line 54 : 249, 23, 23, 23, 23, 23, 3, 3, 2, 2, 21.

"C", mass storage name: "CNAVY" is an ASCII file; that with the exception of line 29, the comments can be varied at will. Please note, however, that they are used as follows:

Lines	Meaning	Max. Length in characters
0008	Comments on player's misses	9
09 18	Reaction to own misses	12
18 21	Joy about sunk player's ship	12
22 28	Joy about hits	12

Operations:

1. if the part-programs are not yet stored in the extended memory.

You call "ADMIRAL," "XEQ "READP." This program starts automatically, loads the other parts and then deletes itself.

If the program parts already exist, step 1 is omitted and you start by putting this part into the main memory with "NAVY", "XEQ" "GETP".

3. if "NAVY" is already in the main memory, then skip the previous steps, and you can immediately start the game by "XEQ "NAVY"". However, before calling "NAVY", at least 171 registers have to be set to your disposal.

4. Now the computer gives some hints, asking the player to build his ships, specify the size of the playing field and the number of ships to recall.

The size of the playing field is 10 x 10. The squares are two-digit numbers, the first digit being the line number and the second the column number (in each case from 0 to 9).

Player and computer build 9 ships each in their playing field: One of each with 5, 4, 3 and 2 boxes in length and 5 vessels with a length of only one field (submarines). The ships may not lie diagonally.

Contact with one another and to the edges of the playing field is allowed. By specifying a position number, the computer and the player alternately query the field. The player answers by specifying the field content (a 0 for an empty field, otherwise the length of the hit ship) and pressing "R/S". The HP-41 responds to a hit with the length of the ship, otherwise with a humorous comment.

Illegal entries are recognized and criticized. and obviously the computer reacts sourly!

Since the game lasts quite long, you can interrupt it at any time, by turning off the computer. After switching on the HP-41 it gets the program and continues automatically.

"ADMIRAL" 72 Bytes 11 REG SIZE 38

"NAVY" 43 Bytes 7 REG SIZE 38

"N" 133 Bytes 19 REG SIZE 38

"P" 798 Bytes 114 REG SIZE 38

"O". 175 Bytes 25 REG SIZE 38

Ed's note: the version below is included in the CL Module, and therefore doesn't need to use the Mass Storage approach – a much simpler program flow, just start with XEQ "NAVY".

Program listing:

01*LBL "K"	36 STO 12	70 19	13 TONE 0
02 RCL 13	37 GTO "W"	71 STO 15	14 TONE 9
03 RCL 10	38*LBL "NAVY"	72 STO 16	15 LASTX
04 *	39 38	73 4	16 STO b
05 ST+ IND 01	40 PSIZE	74 STO 04	17*LBL Z
06 DSE 08	41 XROM "CNAV"	75 E2	18 17.02701
07 GTO "W"	42 "FELD=10*10"	76 XROM "#"	19 REGSWAP
08 E	43 AVIEW	77 STO 12	20 RTN
09 ST- 13	44 PSE	78 3	21*LBL 98
10 RCL 13	45 CLRG	79 STO 03	22 DSE 13
11 STO 08	46	80 RCL 12	23 GTO IND 08
12 X#Y?	"JE1*5,4,3,2+5*1"	81 2	24*LBL 00
13 GTO 05	47 AVIEW	82 STO 02	25 CLX
14 5	48 5	83 MOD	26 X<>F
15 STO 08	49 STO 13	84 15	27 9
16 STO 05	50 STO 08	85 *	28 STO 08
17 GTO 05	51 "V"	86 STO 08	29 E1
18*LBL "P"	52 ASTO 05	87 CLA	30 STO 13
19 RCL 06	53*LBL 05	88 ASTO 09	31 3
20 ST- 12	54 E2	89 GTO "NAV2"	32 RCL 16
21 XROM "?"	55 XROM "#"	90 END	33 X<=Y?
22 RCL 13	56 STO 12		34 GTO 01
23 X#Y?	57 CF 07	01*LBL "NAV2"	35 2
24 GTO 05	58 XROM "?"	02 "CNAV"	36 13
25 RCL 10	59 X#0?	03 29	37 XROM "#"
26 *	60 GTO 05	04 SEEKPTA	38 X<=Y?
27 ST- IND 01	61 RCL 13	05 GTO IND 08	39 GTO IND X
28 E	62 RCL 10	06*LBL 00	40 "u"
29 ST+ 08	63 *	07 GETREC	41 AROT
30 FS? 07	64 ST+ IND 01	08 AVIEW	42 ATOX
31 GTO "P"	65 DSE 08	09 GTO 17	43 STO 07
32 SF 07	66 GTO IND 05	10*LBL X	44 RCL 12
33 - E	67 "!!* " "	11 " ` ???"	45 -2
34 ST* 06	68 RCL [12 AVIEW	46 MOD
35 RCL 11	69 STO d		47 SIGN

48 ST* 07	101 STO b	154 RCL 10	207 RCL 10
49 GTO 09	102 CLA	155 *	208 *
50*LBL 00	103 ARCL X	156 ST- IND 01	209 ST+ IND 01
51 3	104 AVIEW	157 DSE 15	210 9
52 STO 08	105 >": "	158 GTO 17	211 RCL 00
53*LBL 03	106 X<0?	159 "GRATULIERE"	212 X=Y?
54 "-----"	107 GTO X	160 AVIEW	213 GTO 03
55 RCL 13	108 99	161 TONE 0	214 DSE 16
56 AROT	109 X<>Y	162 TONE 7	215 GTO 04
57 ATOX	110 X>Y?	163 BEEP	216 "DAS WAR'S"
58 12	111 GTO X	164 TONE 0	217 AVIEW
59 -	112 X<> 12	165 TONE 7	218 BEEP
60 GTO 08	113 STO \	166 BEEP	219 XEQ Z
61*LBL 04	114 XROM "?"	167 TONE A	220 E-3
62 E1	115 .	168 TONE 3	221 -
63 X<> 07	116 X<> \	169 BEEP	222 STO 06
64 ST/ 07	117 STO 12	170 TONE 1	223 "MEINE RESTE:"
65 GTO 09	118 GTO IND 00	171 TONE 0	224 AVIEW
66*LBL 01	119*LBL 00	172 TONE 9	225 CLA
67 15	120 E	173*LBL Y	226*LBL 12
68 STO 13	121 ST+ 15	174 CLRG	227 RCL 06
69 37	122 9	175 CLST	228 17
70 GTO 08	123 XROM "#"	176 "N"	229 -
71*LBL 02	124 SEEKPT	177 TONE 7	230 INT
72 4	125 ARCLREC	178 PCLPS	231 E1
73 STO 08	126 TONE 7	179*LBL 17	232 *
74 20	127 GTO 00	180 XEQ Z	233 STO 12
75 STO 13	128*LBL 01	181*LBL 18	234 9
76 RCL 06	129 >"U-BOOT"	182 CF 10	235 +
77*LBL 08	130 TONE 0	183 XROM "N"	236 E3
78 STO 07	131 GTO 00	184 RCL b	237 /
79*LBL 09	132*LBL 02	185 STO L	238 ST+ 12
80 RCL 12	133*LBL 03	186 ARCL 12	239*LBL 13
81 RCL 07	134*LBL 04	187 `` ?"	240 RCL IND 06
82 -	135*LBL 05	188 SF 11	241 X=0?
83 E2	136 ARCL 00	189 TONE IND 00	242 GTO 02
84 MOD	137 TONE 1	190 PROMPT	243 XROM "?"
85 STO 12	138 DSE IND 00	191 CLA	244 X#0?
86 XROM "?"	139 GTO 00	192 FC?C 22	245 XEQ 01
87 X#0?	140 >"SINKT"	193 STO b	246 ISG 12
88 GTO 98	141 TONE 7	194 ARCL 12	247 GTO 13
89*LBL 16	142 TONE 7	195 >"="	248*LBL 02
90 XEQ Z	143 TONE 7	196 ARCL X	249 ISG 06
91*LBL 15	144 TONE 9	197 AVIEW	250 GTO 12
92 SF 10	145 TONE 9	198 X<0?	251 GTO Y
93 SF 11	146 TONE 9	199 GTO X	252*LBL 01
94 RCL b	147 TONE 7	200 5	253 RCL 10
95 STO L	148 TONE 7	201 X<>Y	254 *
96 "FRAG MICH"	149 TONE 7	202 X>Y?	255 ST- IND 01
97 TONE 5	150*LBL 00	203 GTO X	256 RCL 12
98 TONE 5	151 AVIEW	204 X=0?	257 INT
99 PROMPT	152 >": "	205 9	258 E1
100 FC?C 22	153 RCL 00	206 STO 00	

259 X>Y?	312 "DU	364 RCL 00	11 INT
260 ``0"	MOGELST"	365 STO 13	12 -2
261 ARCL Y	313 AVIEW	366 E	13 MOD
262 ``:"	314 TONE 6	367 -	14 SIGN
263 SF 29	315 GTO Y	368 STO 08	15 ST* 06
264 ARCL 00	316*LBL 01	369 5	16*LBL "W"
265 AVIEW	317 FC?C 07	370 STO 07	17 RCL 06
266 TONE IND 00	318 GTO 02	371 GTO "V"	18 ST+ 12
267 CF 29	319 E1	372*LBL 01	19 RCL 12
268 ``"	320 X<> 06	373 6	20 X<0?
269 RTN	321 ABS	374 FS? 03	21 GTO "P"
270*LBL 03	322 ST/ 06	375 7	22 E1
271 E1	323 RCL 13	376 22	23 /
272 XROM "#"	324 E	377 XEQ 09	24 LASTX
273 9	325 -	378 SF 03	25 X<=Y?
274 +	326 STO 08	379 GTO "W"	26 GTO "P"
275 SEEKPT	327 GTO 03	380*LBL "J"	27 RCL 06
276 GETREC	328*LBL 02	381 FS? 03	28 ABS
277 TONE 0	329 - E	382 GTO 16	29 X=Y?
278 AVIEW	330 ST* 06	383 DSE 08	30 GTO 00
279 FS? 04	331 SF 07	384 GTO "W"	31 RCL Z
280 GTO "M"	332*LBL 03	385 RCL 13	32 INT
281 GTO 98	333 RCL 11	386 E	33 RCL 11
282*LBL 04	334 STO 12	387 -	34 E1
283 E	335 FS? 03	388 STO 08	35 /
284 RCL 00	336 CF 07	389 RCL 11	36 INT
285 X=Y?	337 GTO "W"	390 RCL 06	37 X#Y?
286 GTO 01	338*LBL 04	391 +	38 GTO "P"
287 FC? 04	339 5	392 STO 12	39*LBL 00
288 GTO 04	340 22	393 GTO 16	40 XROM "?"
289 RCL 13	341 XEQ 09	394*LBL 09	41 X#0?
290 X=Y?	342 GTO 00	395 STO [42 GTO "P"
291 GTO 05	343*LBL 05	396 RDN	43 GTO "K"
292 "NA SO WAS"	344 CF 07	397 XROM "#"	44*LBL "#"
293 AVIEW	345 DSE 08	398 RCL [45 TIME
294 TONE 2	346 GTO 01	399 +	46 E6
295 TONE 5	347 3	400 SEEKPT	47 *
296 TONE 0	348 19	401 GETREC	48 X<>Y
297 CLA	349 XEQ 09	402 AVIEW	49 MOD
298 ARCL 09	350 CF 03	403 TONE 9	50 INT
299 RCL 12	351 CF 04	404 END	51 RTN
300 XTOA	352 RCL 11		52*LBL "?"
301 ASTO 09	353 STO 12		53 XROM "N"
302 GTO "M"	354 CLA	01*LBL "V"	54 E1
303*LBL 01	355 ARCL 09	02 RCL 12	55 *
304 4	356 ATOX	03 STO 11	56 RCL IND 01
305 23	357 X=0?	04 2	57 X<>Y
306 XEQ 09	358 GTO 98	05 XROM "#"	58 /
307 FC? 04	359 STO 12	06 10^X	59 FRC
308 GTO 98	360 ASTO 09	07 STO 06	60 E1
309*LBL "M"	361 XROM "?"	08 TIME	61 *
310 DSE 07	362*LBL 00	09 E4	62 INT
311 GTO 01	363 SF 04	10 *	63 STO 00

64 RTN	68 /	72 FRC	76 10^X
<u>65*LBL "N"</u>	69 17	73 E1	77 STO 10
66 RCL 12	70 +	74 *	78 END
67 E1	71 STO 01	75 INT	
<u>01*LBL "CNAV"</u>	25 APPREC	49 APPREC	
02 "LOADING..."	26 "TREIBHOLZ"	50 "WRACK AHOI"	
03 AVIEW	27 APPREC	51 APPREC	
04 "CNAV"	28 "L.M.A.A."	52 "HA-HA"	
05 SF 25	29 APPREC	53 APPREC	
06 PURFL	30 "SCHWUND"	54 "TOLL"	
07 CF 25	31 APPREC	55 APPREC	
08 37	32 "A***LOCH"	56 "DANKE"	
09 CRFLAS	33 APPREC	57 APPREC	
10 "NICHTS"	34 "MIST"	58 "PRIMA"	
11 APPREC	35 APPREC	59 APPREC	
12 "WASSER"	36 "GANOVE"	60 "KLASSE, BABY"	
13 APPREC	37 APPREC	61 APPREC	
14 "SEETANG"	38 "HALUNKE"	62 "WEITER SO"	
15 APPREC	39 APPREC	63 APPREC	
16 "HERINGE"	40 "NA GUT"	64 "AHA"	
17 APPREC	41 APPREC	65 APPREC	
18 "PLANKTON"	42 "FRUST"	66 "WAR MIR KLAR"	
19 APPREC	43 APPREC	67 APPREC	
20 "PLATSCH"	44 "SCH*****"	68 "ICH BEGINNE: "	
21 APPREC	45 APPREC	69 APPREC	
22 "DENKSTE"	46 "#-!"#\$%&'("	70 "DONE"	
23 APPREC	47 APPREC	71 AVIEW	
24 "VON WEGEN"	48 "TSCHUESS"	72 END	

Sub Hunt

HP Co. – Games Pac



You are the commander of a destroyer with orders to search out and destroy an enemy submarine. The submarine is trapped somewhere in a bay having dimensions of 10 by 10 leagues. Your destroyer is equipped with sonar having a range of 2.5 leagues. Since your destroyer has been recently restocked, you have a seemingly unlimited supply of depth charges. To find the submarine, input the position (row and column) of your destroyer. Then send out a sonar signal (a beep) by pressing [R/S] or [E]. If the signal is reflected back (indicated by a second beep), the submarine is within 2.5 leagues of your destroyer. The tone of the beep is related to the distance to the submarine.

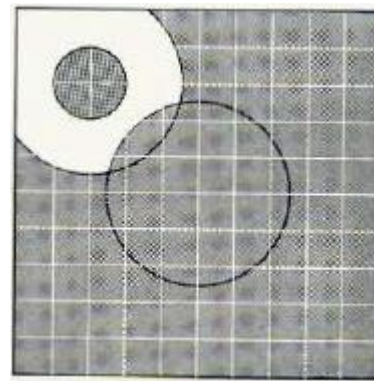
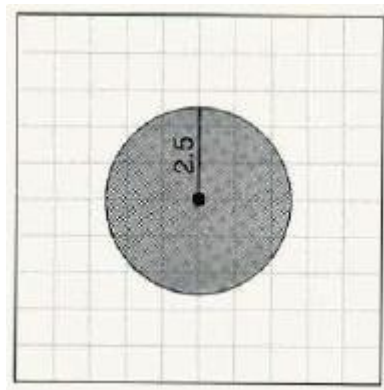
The higher the tone, the closer the submarine is. If the reflected tone is the same as the initial signal, the submarine is within 0.25 leagues of the destroyer. When you think that you have located the submarine, attempt to move your destroyer directly over it for an attack by inputting the new coordinates. Then drop a depth charge by pressing 0. The closer you are to the submarine, the greater your chances for a hit. If you are more than 1.5 leagues from the submarine, there is no chance for a hit. A successful attack is indicated by **"BOOM"** appearing in the display. Otherwise, **"MISSED"** will be displayed and the submarine will have moved. Its new location will be within a radius of one league from its last position.

When you have begun the game, you will be prompted with **"HARD? Y/N"**. If you select **"Y"** (yes) the game is more Challenging because the submarine is allowed to move after each sonar scan as well as after each depth charge miss. (All of these movements are restricted to within a radius of 1 league). Playing the easier game, try to destroy the submarine using no more than ten sonar readings and one depth charge. Anytime the prompt **"ORDERS"** appears in the display you can check your present score by pressing [C]. The display will show the number of depth charges (**CANS**) dropped, followed by the number of sonar readings (**SCANS**) taken.

Let's play an "easy" game:

Keystrokes (SIZE ≥ 009)	Display
[XEQ] [ALPHA] SUBHUNT [ALPHA]	SEED ?
45.6 [R/S]	HARD? Y/N
N [R/S]	ORDERS
First move:	
5 [ENTER] [R/S]	ORDERS

There was no echo, thus the shaded locations (left figure below) can be eliminated:



Second move:

Look for the sub in the upper left hand corner.

2 **ENTER** 8 **R/S**

ORDERS

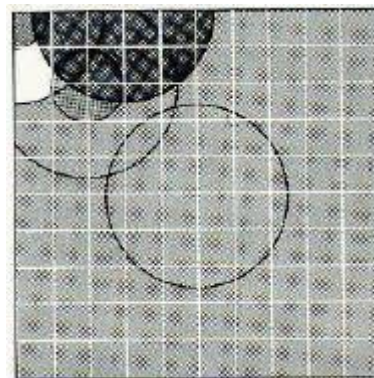
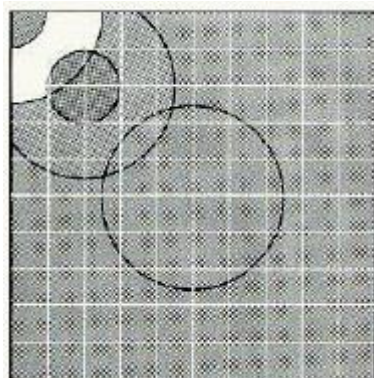
We hear a low tone echo. Since it was a low tone the area very close to the destroyer can be eliminated. Also eliminated is the area outside a 2.5 league radius from 2,8 (figure on the right above).

Third move: continue the search.

0 **ENTER** 10 **R/S**

ORDERS

We hear a low tone echo. The area close to 0,10 can be eliminated because it was a low tone. Also eliminated is the area outside of a 2.5 radius from 0, 10 (below left figure):

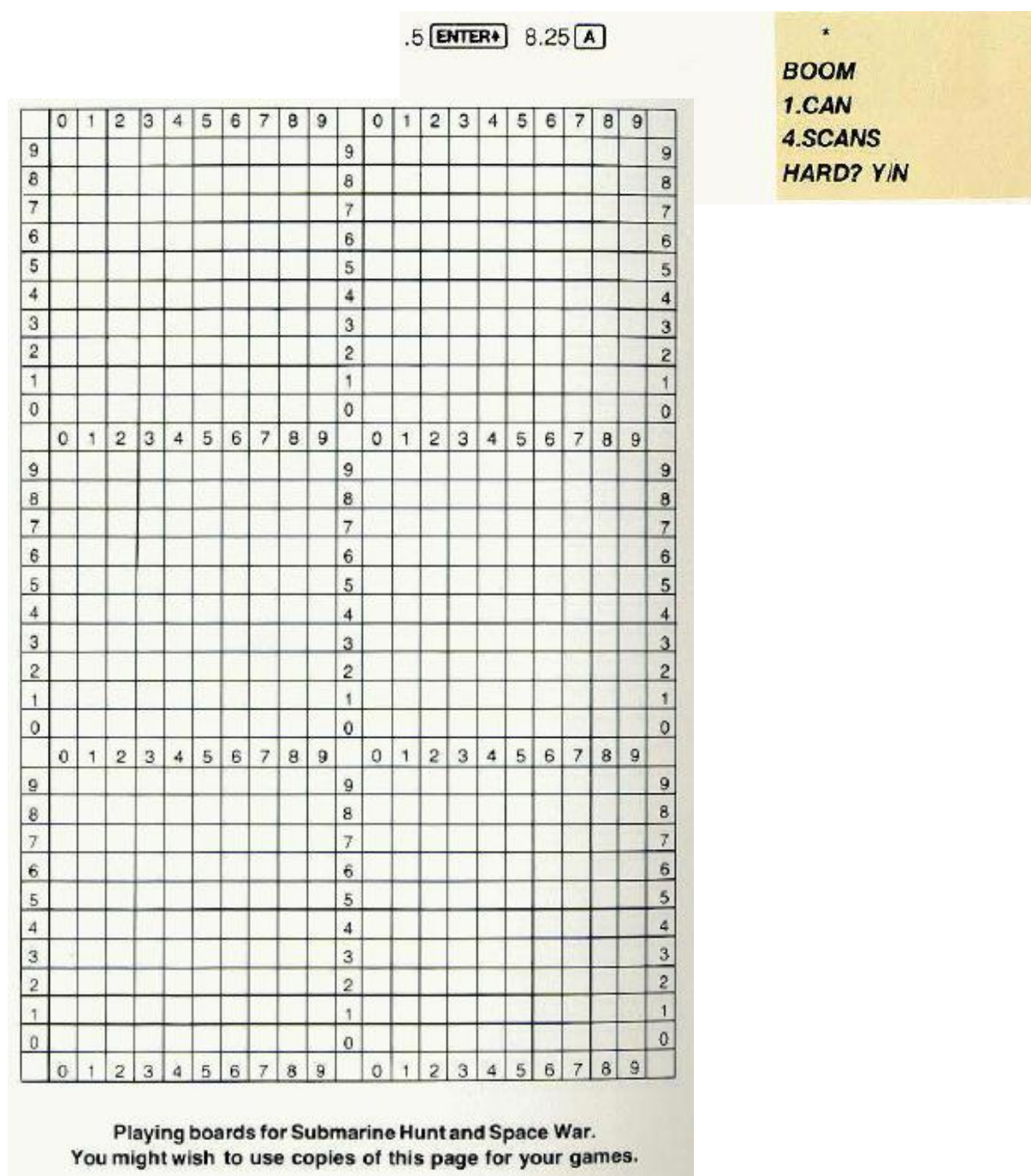


Try location 3, 10 to eliminate more area.

3 **ENTER** 10 **R/S**

ORDERS

We hear no echo. Since the area has been narrowed down quite a bit, a depth charge is now dropped in the center of that area (above right figure):



Program listing:

01*LBL "SUBHUNT"

02 9	11 XEQ 18	21 ASTO X
03 XROM "INIT"	12 STO 01	22 X=Y?
04 SF 27	13 XEQ 18	23 GTO 20
05 XROM "SEED"	14 STO 02	24 SF 00
06 STO 00	15 "N"	25*LBL 20
07 FC? 55	16 ASTO Y	26 RCL 05
08 CF 21	17 "HARD? Y/N"	27 RCL 03
09*LBL 23	18 AON	28 "ORDERS"
10 CF 00	19 PROMPT	29 FIX 2
	20 AOFF	30 PROMPT

31 GTO E	82 STO 08	133 LASTX
32*LBL 18	83 GTO 23	134 MOD
33 XROM "RND0"	84*LBL 01	135 180
34 10	85 XEQ a	136 XROM "RND0"
35 *	86 XEQ 24	137 *
36 RTN	87 TONE 0	138 +
37*LBL E	88 "MISSED"	139*LBL 03
38 1	89 AVIEW	140 XROM "RND0"
39 ST+ 08	90 PSE	141 P-R
40 RDN	91 GTO 20	142 ST+ 02
41 XEQ c	92*LBL C	143 RDN
42 TONE 9	93 XEQ 25	144 ST+ 01
43 FS? 00	94 GTO 20	145 1
44 XEQ a	95*LBL 25	146 XEQ b
45 CF 05	96 FIX 0	147 2
46 RCL 04	97 SF 03	148 XEQ b
47 2.5	98 RCL 07	149 RTN
48 X>Y?	99 XEQ 21	150*LBL b
49 GTO 01	100 RCL 08	151 10
50 GTO 20	101 XEQ 21	152 RCL IND Y
51*LBL 01	102 RTN	153 X<Y?
52 /	103*LBL 21	154 GTO 01
53 10	104 CLA	155 20
54 *	105 ARCL X	156 -
55 INT	106 "` "	157*LBL 01
56 9	107 FC?C 03	158 ABS
57 X<>Y	108 "S"	159 STO IND Z
58 -	109 "CAN"	160 RTN
59 TONE IND X	110 1	161*LBL 02
60 CLX	111 X#Y?	162 XROM "RND0"
61 GTO 20	112 "S"	163 GTO 03
62*LBL A	113 AVIEW	164*LBL c
63 1	114 PSE	165 X<>Y
64 ST+ 07	115 RTN	166 STO 05
65 RDN	116*LBL a	167 RCL 01
66 XEQ c	117 RCL 01	168 -
67 .25	118 RCL 05	169 X<>Y
68 -	119 -	170 STO 03
69 .8	120 RCL 02	171 RCL 02
70 *	121 RCL 03	172 -
71 XROM "RND0"	122 -	173 R-P
72 X<=Y?	123 R-P	174 STO 04
73 GTO 01	124 2.5	175 RTN
74 XEQ 24	125 X<=Y?	176*LBL 24
75 "BOOM"	126 GTO 02	177 CLA
76 XROM "BOOM"	127 RDN	178 " * "
77 AVIEW	128 X<>Y	179 "` "
78 PSE	129 90	180 AVIEW
79 XEQ 25	130 -	181 END
80 CLX	131 360	
81 STO 07	132 +	

Sub Hunt, v1.

[Charles Campbell - PPC V7N4 p15; \(May 1980\)](#)

This is modification of the Sub Hunt-Five Sub which appears in "65 NOTES, V4, N6, page 34". The Sub Hunt-Five Sub program was a modification of an HP-65 Sub Hunt program written by Jacob R. Jacobs, (PPC 99), and to him should go the credit. Without his fine contribution, this program would not have been done.

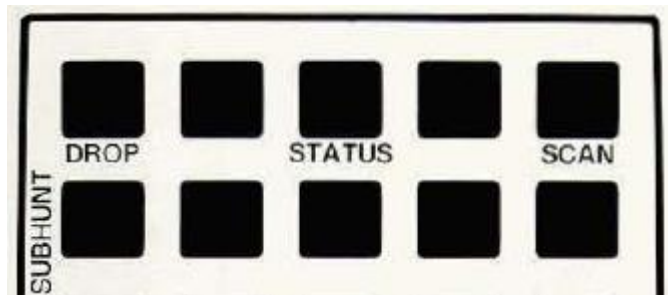
The game is played on a nine by nine grid, with rows and columns numbered one through nine. A 9x9 grid could be drawn on paper and placed in a 8 1/2 x 11 inch acetate paper holder made for a 3 ring binder to keep clues there and with a grease pencil, which can be erased with a rag. Any number of subs to be found and sunk (from one through twelve may be selected). When sunk the program removes them from the grid, so that it will not mask your radar when firing at adjoining squares. Two players may play by choosing an odd number of subs to find and sink. When a player sinks a sub(s), he receives an extra shot for each sub sunk. There can be more than one sub in a square. The player who sinks the majority of the subs, WINS. A single player will find it, choosing to sink the subs in the least number of shots. (The player will have his own record of the minimum number of shots to sink x number of subs). See "user instructions" for more details.

USER INSTRUCTIONS

1. Load Program and select "User" mode
2. Select number of subs to be sunk from 1 through 12: "Shift A" (Default is 5 subs).
3. (Optional) Input SEED: any number $0 < S < 1$ and hide subs: "Shift E"
3. OR start by hiding subs: "Shift E" (no input)
4. Input a trial coordinate as "RC", where R = row number and C = column number, then FIRE: "A". If 0 appears: no sub(s) in the square nor in the adjoining squares, including the diagonal squares. If 1 appears: a sub(s) not already sunk is in an adjoining square(s), including the diagonal squares. If 2 or 4 or 6 or 8 etc., appears: direct hit on 1 or 2 or 3 or 4 etc., subs within the square fired upon and the sub(s) are sunk.
5. (Optional) To see number of shots fired: "C"
6. (Optional) To see last RC fired upon: "D"
7. (Optional) To see the number of subs sunk and the remaining subs to sink: "B" and see SS.RR where SS is the number sunk and RR remains to be sunk. If a" of the subs are sunk, the R.C location of each of the subs will be displayed during a pause, then the program stops with the number of shots fired, displayed.
8. Repeat step 4 until all subs are found and sunk.
9. To start a new game, go to step 2 or 3.

NOTE: "OOPS?" will be displayed if R/S is pushed after steps 2,3,4, 5,6 or 7. Continue on by pushing a legal key, for there has been no damage done to the game.

41C SUB HUNT		
SIZE 18 NO RAMS REQUIRED		
R0	Index & Temp.	L SH (Global)
R1	Sub	LD & 13 Last shot fired
R2	Sub	Le seed? & Hide subs
R3	Sub	La # of subs to find
R4	Sub	LA FIRE
R5	Sub	LB # sunk
R6	Sub	LC # Shots fired
R7	Sub	L0 through L18 used
R8	Sub	Flag 00 & 22 used.
R9	Sub	
R10	Sub	
R11	Sub	
R12	Sub	
R13	# of subs	
R14	last shot	
R15	# sunk	
R16	seed	
R17	# shots fired	



Program listing:

01*LBL "SH"	24 STO 14	47 X<>Y
02*LBL 13	25*LBL 17	48 PSE
03*LBL D	26 XEQ 11	49 PSE
04 RCL 14	27 FIX 0	50 DSE 00
05 CF 22	28 RTN	51 GTO 09
06 E1	29 GTO 14	52*LBL C
07 *	30*LBL B	53 RCL 17
08 FIX 0	31 XEQ 11	54 FIX 0
09 RTN	32 RCL 15	55 CF 22
10 GTO 14	33 X#Y?	56 RTN
11*LBL e	34 GTO 18	57 GTO 14
12 FS?C 22	35 SCI 1	58*LBL 00
13 GTO 08	36 XEQ 00	59 XEQ 11
14*LBL 03	37 -1	60 E-5
15 XEQ 00	38 ENTER^	61 +
16*LBL 01	39*LBL 09	62 STO 00
17 XEQ 02	40 RDN	63 RTN
18 STO IND 00	41 RCL IND 00	64*LBL 02
19 DSE 00	42 X<>Y	65 RCL 16
20 GTO 01	43 E1	66 PI
21 ,	44 *	67 +
22 STO 17	45 *	68 X^2
23 STO 15	46 LASTX	69 FRC

70 STO 16	112 E	154 X<>Y
71 89	113 ST+ 17	155 RTN
72 *	114 RDN	156*LBL 08
73 INT	115 CF 22	157 E
74 ,1	116 FIX 0	158 X>Y?
75 *	117 RTN	159 GTO 16
76 1.1	118 GTO 14	160 RDN
77 +	119*LBL 05	161 RTN
78 INT	120 RCL IND 00	162 GTO 14
79 LASTX	121 RCL 14	163*LBL 16
80 X=Y?	122 X=Y?	164 RDN
81 GTO 02	123 GTO 06	165 STO 16
82 RTN	124 -	166 GTO 03
83*LBL 11	125 ABS	167*LBL a
84 5	126 .1	168 FIX 0
85 FS? 00	127 X<>Y	169 RND
86 RCL 13	128 X=Y?	170 X=0?
87 RTN	129 GTO 07	171 GTO 14
88*LBL A	130 E	172 12
89 FIX 0	131 -	173 X<>Y
90 STO 00	132 ABS	174 X>Y?
91 E1	133 X<=Y?	175 GTO 14
92 /	134 GTO 07	176 STO 13
93 STO 14	135 RDN	177 SF 00
94 FRC	136 RDN	178 CF 22
95 X=0?	137 RTN	179 RTN
96 GTO 13	138*LBL 06	180 GTO 14
97 RCL 00	139 -1	181*LBL 18
98 11	140 ST* IND 00	182 FIX 2
99 X>Y?	141 ABS	183 -
100 GTO 13	142 R^	184 LASTX
101 CLX	143 2	185 X<>Y
102 99	144 X<=Y?	186 E2
103 -	145 +	187 /
104 X>0?	146 E	188 +
105 GTO 13	147 ST+ 15	189 RTN
106 XEQ 00	148 RDN	190*LBL 14
107 CLX	149 RTN	191 "OOPS?"
108*LBL 04	150*LBL 07	192 ASTO X
109 XEQ 05	151 E	193 END
110 DSE 00	152 R^	
111 GTO 04	153 X<=Y?	

SubHunt, v2.

[James R. Merrill - PPCCJ V8N2 p17; \(Mar/Apr 1981\)](#)

SUBHUNT, a new simulation game for your HP-41C with 896 bytes of available memory, exactly 8 sides for those with the 82104A card reader. Fast being adapted by the Navy for its submarine warfare school programs, you, as the captain of the fleet boat SS-410, are sent by COMSUBPAC to intercept an enemy convoy of up to 7 ships with 35 torpedoes on board and the latest model of the Hewlett Packard 4100 Series TDC (Target Data Computer). The HP 4100 TDC incorporates sonar, ship-wide repair, evasive, and forward advance capabilities as well as actual fire control over your torpedoes.

As a result you are the only crew member on the SS-410 since the 4100 TDC is so complete. But you remain in absolute control over the eventual task that the 4100 TDC performs. There is, as you might have suspected, a catch. First, you are constrained to chasing and sinking one enemy convoy ship at a time. Secondly, there are always enemy destroyers escorting the convoy and your 4100 TDC unfortunately cannot handle these "tincans". Speaking of cans, these destroyers are loaded down with depth charges and can't wait to try them out on the arrogant Yankee submarine lurking beneath the surface and menacing the convoy. So you must inevitably contend with these dangerous depth bombs as well. As the game is described below, it is strongly suggested that the prospective captain read the descriptions before playing the game the first time to avoid frustration that usually accompanies such game simulations.

INITIALIZATION. A minimum of 21 data registers is necessary for the simulation to execute properly. Key "XEQ 'SUB'" and the HP-41C with a SEED? prompt asks for a seed. This is the only point, not including the end of the simulation, that the program actually halts and during the game, the necessary inputs are given within the PAUSE data entry feature as needed. The ~ame title is shown (*SUBHUNT*) and a status report (described under command # 5 below) and is followed by a CMD? (COMMAND?) message and the command should be selected from the list below and entered during the Pause that occurs (uses the data entry flag 22). Any illegal commands are screened out and control returns to the CMD? prompt. Again, do not stop the program.

CMD_#	Function
1	ADVANCE
2	FIRE
3	EVADE
4	REPAIR
5	STATUS

A comprehensive description of each follows.

Command # 1. ADVANCE.

This command advances your boat a random distance, reduced by exploding depth charges and the total damage to the boat. A minimum distance of 1200 yards is incorporated in the 4100 TDC, and if you go closer (or try) a message appears and control returns to the CMD?

prompt. Sorry, but no provision exists for retreating as this would be cheating yourself of valuable combat experience.

Command # 2. FIRE.

This command fires torpedoes. It is inoperative when the total damage to the boat is greater than 80 (of a possible 100) units. If so, the total damage is shown and control returns to the CMD? . If the boat is okay, depth and speed prompts occur during which you enter your desired settings during the pause that occurs (without stopping the game) and the settings from the previous firings will remain (unless they are outside the limit imposed by the 4100 TDC of a minimum 7 foot draft and a 45 maximum knots speed setting). Angle on the bow, ship course, bearing and speed are computed instantly by the 4100 TDC, making things easy.

Next the 4100 TDC asks you for the number to be fired on this shot, and you should enter a digit from 1 to 4 on the FIRE? prompt during the pause, again not stopping the game. The maximum is four and the default value is four, and the number fired on the last shot is not retained by the 4100 TDC. Please don't enter zero. In any event, the number fired on this shot will be confirmed. At this point you might be depth charged but the torpedoes on their way are unaffected. Then, if your depth setting is too deep or your speed is too slow, a relevant message appears, all the torpedoes on this shot miss, you might be depth charged, and control returns to the CMD? prompt. On the other hand, if the torpedoes have a chance of readily reaching the target ship, anything might happen. You could be bombed again (?I?). You might have a dud (damn the Bureau of Ordnance) or you might find that the enemy ship has evaded your torpedo. Finally, a hit may occur, which reduces the enemy's evasive ability. If not sunk on this shot, control returns to CMD? If sunk, any torpedoes remaining on this shot are lost and a message *SHIP SUNK* appears, followed by the tonnage of that ship, and the 4100 TDC moves relentlessly on for a new setup on the next enemy ship with a STATUS report (command #5). If you run out of torpedoes, an end of game display occurs with a comforting "NICE TRY" from COMSUBPAC, neither losing or winning the game. If you sink all the ships, the related total tonnage appears and a YOU WIN appears in the display of the 4100 TDC. End of patrol.

Command #3: EVADE.

The evade command allows the 4100 TDC to evade one depth charge and is active only when that depth charge is dropped. So it is suggested that this command be used once prior to an advance or fire situation when the total damage seems to be getting untenable.

Command #4. REPAIR.

Here, the 4100 TDC will implement repairs throughout the boat on a random basis, detracting, not below zero, from the cumulative damage inflicted by the depth charges. An advantage is that the 4100 TDC is ultra-quiet here and consequently, the 4100 and the SS-410 can't be bombed during this command.

Command #5. STATUS.

Again, a non-bombable (?) command, it shows the number of torpedoes left, the ships remaining to be sunk, and the current total damage from enemy depthcharges, and finally the current range in yards to the enemy ship the 4100 TDC is currently locked onto. Control returns to the CMD? prompt.

Depth Charge. Not in user input command, they can be dropped on you within the advance or firing commands by the Ubiquitous, elusive enemy destroyers, Noted by the DEPTH CHARGE message, a random amount of damage is inflicted (reduced when evasive action is taken) and it also slows the advance to the enemy ship but does not interfere with outgoing torpedoes. It is denoted by the DMG=dd message (dd=current damage). If the total damage is greater than 80, your fire control mechanisms do not work. If the total is over 100 units, your boat is sunk and you receive a posthumous Congressional Medal of Honor for "valor& guts" by executive order. A YOU LOSE appears and the end of game routine described under #2 above is shown.

General Notes.

- Thanks to John Rausch (88) for his Star Trek which provided some hints. (V7N2 p40-45)
- This program does not need the PPC ROM, Black Boxes, byte jumpers, etc., just a "regular'olde" HP-41C with 896 bytes of RAM (8card sides) Not ' . .,~ optimized for the printer, card reader, or wand.
- Requires registers 00-20 (SIZE 021), Flags 01-09, and numeric labels 00-27.

James R. Merrill (1625)

Program listing:

10:09AM 05/23	18 STO 10	36 STO 18	54 X<=0?
01*LBL "SUBT"	19*LBL 15	37 RCL 10	55 GTO 17
02 CLRG	20 E4	38 STO 09	56 5
03 FIX 0	21 XEQ 20	39 RCL IND 09	57 X<>Y
04 CF 29	22 2 E3	40 STO 11	58 X>Y?
05 "SEED?"	23 +	41 23	59 GTO 17
06 PROMPT	24 STO IND 09	42 /	60 22
07 ABS	25 DSE 09	43 SQRT	61 +
08 SQRT	26 GTO 15	44 INT	62 GTO IND X
09 STO 00	27 35	45 STO 12	63*LBL 23
10 "*SUBHUNT*"	28 STO 08	46 XEQ 27	64 FS? 06
11 AVIEW	29*LBL 16	47*LBL 17	65 GTO 02
12 XEQ 09	30 E2	48 CF 22	66 XROM "RND0"
13 6	31 XEQ 20	49 "CMD?"	67 .2
14 XEQ 20	32 39	50 AVIEW	68 X<Y?
15 2	33 *	51 PSE	69 XEQ 19
16 +	34 2 E3	52 FC? 22	70 25 E2
17 STO 09	35 +	53 GTO 17	71 FS?C 07

72 16 E2	125 STO 13	178 X>Y?	231 AVIEW
73 XEQ 20	126 "DEPTH="	179 XEQ 19	232 PSE
74 4 E2	127 ARCL X	180*LBL 18	233 GTO 04
75 +	128 "` FT"	181 XROM "RND0"	234*LBL 07
76 RCL 20	129 AVIEW	182 .7	235 "TOO "
77 %	130 PSE	183 X<Y?	236 ARCL L
78 -	131 RCL 14	184 XEQ 19	237 AVIEW
79 RND	132 "SPEED?"	185 SF IND 15	238 PSE
80 STO 19	133 AVIEW	186 "NO. "	239 XEQ 19
81 ST- 18	134 PSE	187 ARCL 15	240 GTO 17
82 RCL 18	135 45	188 AVIEW	241*LBL 08
83 12 E2	136 X<>Y	189 PSE	242 "*SHIP
84 X>Y?	137 X>Y?	190 XROM "RND0"	SUNK*"
85 XEQ 01	138 X<>Y	191 .9	243 AVIEW
86 RCL 19	139 STO 14	192 X<Y?	244 XEQ 09
87 X=0?	140 "SPEED="	193 GTO 05	245 RCL 11
88 GTO 02	141 ARCL X	194 CLX	246 ISG 16
89 "ADV="	142 "` KTS"	195 9	247 STO X
90 ARCL 19	143 AVIEW	196 RCL 18	248 ST+ 17
91 >" YDS"	144 PSE	197 D-R	249 "TONS="
92 AVIEW	145 4	198 /	250 ARCL X
93 PSE	146 "FIRE?"	199 FC? 09	251 AVIEW
94*LBL 00	147 AVIEW	200 GTO 03	252 PSE
95 "RNG="	148 PSE	201 3	253 DSE 10
96 ARCL 18	149 4	202 /	254 GTO 16
97 >" YDS"	150 X<>Y	203*LBL 03	255 "WIN"
98 AVIEW	151 X>Y?	204 X>Y?	256 ASTO L
99 PSE	152 X<>Y	205 GTO 06	257 GTO 14
100 GTO 17	153 "FIRED "	206 "` *HIT*"	258*LBL 09
101*LBL 01	154 ARCL X	207 AVIEW	259 9
102 STO 18	155 AVIEW	208 PSE	260*LBL 10
103 -	156 STO 15	209 SF 09	261 CF IND X
104 ST+ 19	157 ST- 08	210 9	262 DSE X
105 RTN	158 XROM "RND0"	211 ST/ IND 09	263 GTO 10
106*LBL 02	159 .4	212 E	264 RTN
107 "TOO CLOSE"	160 X>Y?	213 RCL IND 09	265*LBL 19
108 AVIEW	161 XEQ 19	214 X<Y?	266 SF 07
109 PSE	162 RCL 18	215 GTO 08	267 XROM "RND0"
110 SF 06	163 RCL 14	216*LBL 04	268 .3
111 GTO 00	164 /	217 CF IND 15	269 FS? 05
112*LBL 24	165 "FAR"	218 DSE 15	270 SQRT
113 RCL 20	166 ASTO L	219 GTO 18	271 X>Y?
114 80	167 80	220 RCL 08	272 RTN
115 X<=Y?	168 X<Y?	221 X<=0?	273 17
116 GTO 12	169 GTO 07	222 GTO 13	274 XEQ 20
117 RCL 13	170 RCL 12	223 GTO 17	275 3
118 "DEPTH?"	171 RCL 13	224*LBL 05	276 FS?C 05
119 AVIEW	172 "DEEP"	225 "` *DUD*"	277 CHS
120 PSE	173 ASTO L	226 AVIEW	278 +
121 7	174 X>Y?	227 PSE	279 X<=0?
122 X<>Y	175 GTO 07	228 GTO 04	280 RTN
123 X<Y?	176 XROM "RND0"	229*LBL 06	281 "DEPTH
124 X<>Y	177 .3	230 >" *EVADED*"	CHARGE"

282 AVIEW	307 2	332*LBL 27	357 AVIEW
283 PSE	308 +	333 "STATUS"	358 PSE
284 ST+ 20	309 ST- 20	334 AVIEW	359 SF 08
285 "DMG="	310 RCL 20	335 PSE	360*LBL 14
286 ARCL X	311 X>0?	336 "TORPS="	361 "END OF GAME"
287 AVIEW	312 GTO 11	337 ARCL 08	362 AVIEW
288 PSE	313 +	338 AVIEW	363 PSE
289 E2	314 0	339 PSE	364 "ΣSHIPS="
290 RCL 20	315 STO 20	340 "SHIPS="	365 ARCL 16
291 SF 08	316*LBL 11	341 ARCL 10	366 AVIEW
292 XEQ 12	317 "`="	342 AVIEW	367 PSE
293 X<=Y?	318 ARCL Y	343 PSE	368 "ΣTONS="
294 RTN	319 AVIEW	344*LBL 12	369 ARCL 17
295 "SUNK..."	320 PSE	345 "ΣDMG="	370 AVIEW
296 AVIEW	321 GTO 12	346 ARCL 20	371 PSE
297 PSE	322*LBL 20	347 AVIEW	372 FS?C 08
298 "LOSE"	323 SF 08	348 PSE	373 STOP
299 ASTO L	324 XROM "RND0"	349 FC?C 08	374 "YOU "
300 GTO 14	325 RTN	350 GTO 00	375 ARCL L
301*LBL 26	326*LBL 25	351 RTN	376 AVIEW
302 "REPAIR"	327 "EVADE"	352*LBL 13	377 STOP
303 AVIEW	328 AVIEW	353 "NO TORPS"	378 END
304 PSE	329 PSE	354 AVIEW	
305 21	330 SF 05	355 PSE	
306 XEQ 20	331 GTO 17	356 "NICE TRY"	

Submarine Hunt, v3.

[Brian Steel - DataFile V2N3 p7 ; \(Jun/Aug 1983\)](#)

Sub-Hunt. Game Description.

This is a simple game of finding and sinking a submarine hidden in 10000 square miles of sea, divided into a 100 x 100 grid. The submarine is destroyed if a shot lands in the square mile in which it is located. If the shot misses the sub by less than 3 miles (i.e. a shot in any adjacent square) then the sub is damaged. The sub can withstand three such near misses before the Captain heads for base to undertake repairs. Subsequently the sub returns to its patrol area but at a different location.

The co-ordinates for the shot are entered in the form XX.VV (i.e. the square 10,90 is entered as 10.9, the square 5,5 is entered as 5.05) and the 41 will tell you how close your shot was. You have sixteen shots to start with. Good luck!

Synthetic Program Lines:-

Line	SP Line	HEX	Instructions	Alternative to SP
09	TONE 0	9F,5A	(RCL IND 31,COS)	Ordinary TONE 0
24	TONE 2	9F,48	(RCL IND 31,Σ-)	'' TONE 2
66	TONE 9	9F,59	(RCL IND 31,SIN)	'' TONE 9
72	E2	1B,12		100
112	Text	-----	See note below	-----
144	TONE 0	-----	As line 09 above	-----
161	E2	-----	As line 72 above	-----

Lines 09, 24, 66 and 144 Synthetic tones

Enter the following PRGM instructions:-

RCL IND 31, COS, then BST twice and byte-grab (BG). Delete the resulting alpha string and SST to see TONE 0. This is a synthetic tone of lower pitch and longer duration than the standard HP-41 tones. The procedure is similar for the other SP tones.

Lines 72 and 161 - Short exponentials These are entered in a different manner to the SP tones. They are formed as follows:-

Key in EEX followed by 2. Do not key in 1 EEX then 2. BST then ENTER" and BG. Delete the resulting text string and the ENTER^. SST to see E2 Line 112 Text string 'YOU·VE SUNK U·

Enter a text string of same length as the string you require, e.g. 'YOUXRE SUNK U', with a 'dummy' character, X in this example, in the position of the apostrophe. Then key in the following instructions:- BST, ENTER^, BG, SST (to the PRGM instruction E^X-1, ~, and RCL 07 BST to ENTER^, and BG again. SST, ~ (x3), and SST to see the synthetic text line "YOU'VE SUNK U"

Program listing:

01*LBL "SUB"	46 2	92 X^2	137 ARCL X
02 ΣREG 04	47 RCL 05	93 +	138 AVIEW
03 CLs	48 X>Y?	94 SQRT	139 PSE
04 49	49 XEQ 10	95 STO 06	140 0
05 STO 08	50*LBL 04	96 4.24	141 STO 07
06*LBL 00	51 FIX 0	97 X>Y?	142 GTO 00
07 "U-BOAT"	52 1	98 GTO 02	143*LBL 01
08 AVIEW	53 ST+ 07	99 GTO 03	144 "YOU LOSE"
09 TONE 0	54 16	100*LBL 06	145 AVIEW
10 CF 29	55 RCL 07	101 1	146 PSE
11*LBL 01	56 X>Y?	102 STO 08	147 TONE 0
12 XEQ 09	57 GTO 01	103 SF 00	148 "SUB WAS IN
13 STO 01	58 "GUESS NO_"	104 "DIRECT HIT"	SQ"
14 XEQ 09	59 ARCL 07	105 AVIEW	149 AVIEW
15 STO 02	60 AVIEW	106 TONE 1	150 PSE
16 GTO 04	61 CF 01	107 TONE 1	151 "_X"
17*LBL 02	62 CF 02	108 TONE 1	152 ARCL 01
18 SF 05	63 CF 03	109 TONE 2	153 "_Y:"
19 1	64 CF 04	110 TONE 1	154 ARCL 02
20 ST+ 05	65 CF 05	111 TONE 2	155 AVIEW
21 "NEAR HIT	66 "ENTER XX.YY"	112 TONE 3	156 PSE
NO_"	67 TONE 9	113 "YOU'VE SUNK	157 GTO 07
22 ARCL 05	68 PROMPT	U"	158*LBL 09
23 AVIEW	69 INT	114 ARCL 08	159 RNG
24 TONE 2	70 STO 03	115 XEQ 09	160 STO 00
25*LBL 03	71 LASTX	116 "IN_"	161 E2
26 FIX 1	72 FRC	117 1	162 *
27 "YOU MISSED"	73 E2	118 RCL 07	163 INT
28 AVIEW	74 *	119 ARCL 07	164 RTN
29 FS?C 05	75 STO 04	120 "" GO"	165*LBL 10
30 GTO 01	76 RCL 01	121 X>Y?	166 0
31 TONE 2	77 RCL 03	122 ""ES"	167 STO 05
32 TONE 0	78 X#Y?	123 AVIEW	168 XEQ 09
33*LBL 01	79 GTO 05	124 TONE 3	169 STO 01
34 "_BY_"	80 RCL 02	125 TONE 3	170 XEQ 09
35 RCL 06	81 RCL 04	126*LBL 07	171 STO 02
36 30	82 X=Y?	127 "_GAME"	172 "SUB BEING"
37 X>Y?	83 XEQ 06	128 ASTO X	173 AVIEW
38 ARCL 06	84*LBL 05	129 FS?C 00	174 PSE
39 RCL 06	85 RCL 02	130 GTO 07	175 "REPAIRED"
40 ""_MILE"	86 RCL 04	131*LBL 08	176 AVIEW
41 1	87 -	132 "ANOTHER"	177 PSE
42 X<Y?	88 X^2	133 ARCL X	178 END
43 ""S"	89 RCL 01	134 PROMPT	
44 AVIEW	90 RCL 03	135*LBL 07	
45 PSE	91 -	136 "NEW"	

Submarine Hunt, v4.

Gary Goodman – UPL #02864C

This program is a further development of the HP's Users' Library program 41-00539-4, "Search and Destroy (w/out Wand)", by Richard Altman. One memory module is required on the basic HP-41C.

You are the captain of a destroyer with orders to seek out and destroy enemy submarines. The destroyer maneuvers on a 10 x 10 grid and searches for the submarine via sonar. The closer you are to the submarine the higher will be the pitch of the sonar's echo and the greater chance you'll have of sinking it when you drop your depth charge. However, if you are inept, the submarine might torpedo you. Two levels of play are available.

The game is played on a 10 x 10 grid numbered as shown in the grid below. The submarine is hiding somewhere within the grid's outer boundaries. The captain may maneuver his destroyer to the center of each box by inputting the grid number (0-99) when requested with ORDERS. Each time the destroyer moves, it sends out a sonar pulse with a range of 2½ units. If the sub is within range of the sonar an echo will be returned; the closer the sub, the higher pitched will be the echo.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

```

0==0-1-2-3-4-5-6-7-8-9
10==0-1-2-3-4-5-6-7-8-9
20==0-1-2-3-4-5-6-7-8-9
30==0-1-2-3-4-5-6-7-8-9
40==0-1-2-3-4-5-6-7-8-9
50==0-1-2-3-4-5-6-7-8-9
60==0-1-2-3-4-5-6-7-8-9
70==0-1-2-3-4-5-6-7-8-9
80==0-1-2-3-4-5-6-7-8-9
90==0-1-2-3-4-5-6-7-8-9

```

An auxiliary program, GRID, is included which will print practice grids as shown above.

The highest pitch (Tone 9) indicates that the sub is within 1/2 unit. The captain attacks by moving the destroyer as close over the sub as he can then dropping a depth charge. The probability of a kill decreases with the distance that the sub is from the destroyer, and is zero if the sub is more than 1 unit away (Tone 1, 3, or 5). A kill is indicated by BOOM appearing on the display. Otherwise MISSED will be displayed and the sub will move to a new location within a radius of 1 unit from its previous position.

Warning: As long as the destroyer is close enough to the sub to receive an echo, the sub also hears the sonar blips. The closer the destroyer comes to the sub and the longer it "hangs around", the more annoyed the sub captain becomes with the destroyer's presence until at last he counter attacks if he can do so without endangering his own ship. For the sub captain to be able to use his

torpedoes he must be at least 1 unit away but closer than $2\frac{1}{2}$ units. The torpedo's accuracy also decreases with distance. Torpedos are very powerful! If the torpedo misses, the sub again moves as described previously.

The object of the game is to kill each sub with as few scans and depth charges (cans) as possible. At any time the status of the game may be obtained and after each game the best, worst and average scores may be obtained.

The advanced level of play is the same except that the captain has only 1 second to decide each move after he is prompted with ORDERS, and the probability of a kill with the depth charges is reduced. The advanced play is activated by setting Flag 01 before starting the game.

Sample problem:

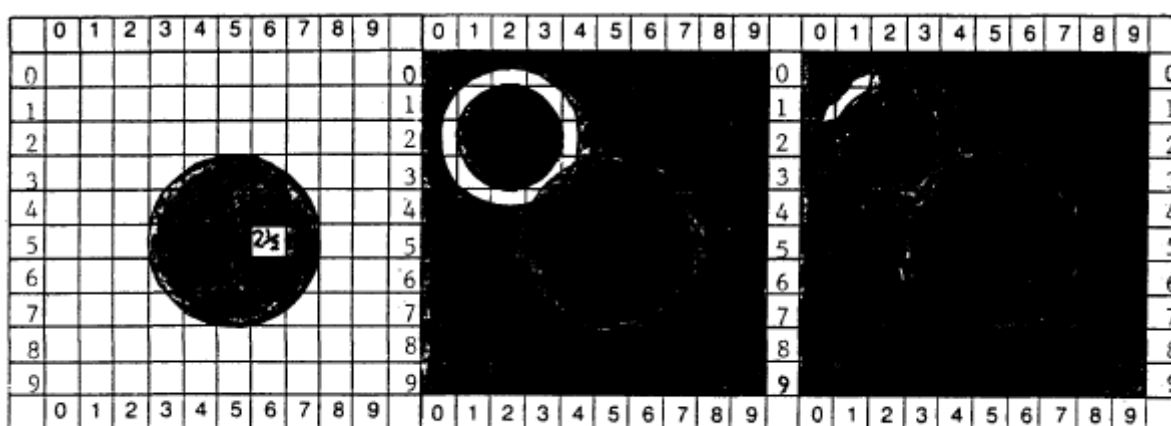


Figure 1

Figure 2

Figure 3

Solution:

Input	Display	Comment
CF 0`		Choose beginner's level
XEQ "SCAN"	SEED?	Asks for the RNG seed
73, R/S	ORDERS: 0	Destroyer starts at location 00
55, R/S	ORDERS: 55 --SCANNING--	your first move since it was no echo the sub isn't within range (figure 1).
22, R/S	ORDERS: 22 --SCANNING-- --CONTACT-- ORDERS: 22	your second move A low pitch echo (TONE 3) indicates that the sub's range is $1\frac{1}{2}$ – 2 units away. Areas closer to or further from the destroyer can be eliminated (figure 2).
0, R/S	ORDERS: 0- --SCANNING-- --CONTACT--	your third move A middle-pitch echo (TONE 5) indicates that the sub is closer, $1-1\frac{1}{2}$ units away. Eliminating the areas closer and further away leaves only a small strip (figure 3).
10, R/S	ORDERS: 10 --SCANNING-- --CONTACT--	your 4 th . move. A high pitch echo (TONE 7) indicates that the sub is between $\frac{1}{2}$ - 1 unit away, close enough to use depth charges.
[A]	BOOM	Dropped a depth charge


```

1 CAN
4 SCANS
GOOD JOB
NEW GAME? YES

```

Important: If when you drop the depth charge the sub is MISSED, then it moves up 1 unit from its previous position.

Optional: Display STATUS

```

[C]      X CAN5
          Y SCANS
          ORDERS: nn

```

Optional: to display TOTALS

```

[D]      NO. GAMES = nn
          WORST GAME =
          x CAN5, y SCANS
          BEST GAME =
          x CAN5, y SCANS
          AVERAGE =
          x.x CAN5
          y,y SCANS

```

When the advanced level of play is selected, the player has only 1 second to respond after ORDERS is displayed. Otherwise, the previous orders will be re-executed. The player does NOT press [R/S] after giving his orders.

Program listing:

1 <u>LBL "SCAN"</u>	21 STO 05	42 AVIEW
2 LBL 15	22 STO 06	43 STO 07
3 SF 27	23 STO 07	44 ISG 06
4 CF 29	24 STO 10	45 CLX
5 CLX	25 FIX 0	46 XEQ 02
6 STO 00	26 10	47 ST+ X
7 STO 01	27 XEQ 01	48 INT
8 STO 08	28 STO 02	49 ST+ X
9 E9	29 10	50 9
10 STO 09	30 XEQ 01	51 X<>Y
11 DEG	31 STO 03	52 -
12 "SEED?"	32 LBL 16	53 X<=0?
13 PROMPT	33 "ORDERS: "	54 GTO 16
14 SIN	34 RCL 07	55 "-CONTACT-"
15 ABS	35 ARCL X	56 TONE IND X
16 STO 04	36 AVIEW	57 AVIEW
17 LBL E	37 FS? 01	58 ST+ 10
18 SF 08	38 PSE	59 RCL 10
19 CF 21	39 FC? 01	60 XEQ 01
20 CLX	40 STOP	61 15
	41 "-SCANNING-"	62 X>Y?

63	GTO 16	116	TONE5	169	FTO 16
64	3	117	TONE4	170	LBL 01
65	R^	118	ISG 00	171	RCL 04
66	X>Y?	119	CLX	172	9821
67	GTO 16	120	RCL 08	173	*
68	" ALERT"	121	RCL 06	174	,2211327
69	AVIEW	122	RCL 05	175	+
70	TONE 8	123	E5	176	FRC
71	TONE 8	124	/	177	STO 04
72	TONE 8	125	+	178	*
73	TONE 8	126	ST+ 01	179	RTN
74	TONE 8	127	X>Y?	180	LBL 02
75	"TORPEDO ATTACK"	128	STO 08	181	RCL 07
76	AVIEW	129	RCL 09	182	E1
77	9	130	X<>Y	183	/
78	XEQ 01	131	X<=Y?	184	INT
79	PSE	132	STO 09	185	RCL 03
80	X>Y?	133	XEQ 04	186	-
81	GTO 18	134	12	187	RCL 07
82	" * KABLAM *"	135	RCL 06	188	E1
83	AVIEW	136	X>Y?	189	MOD
84	PSE	137	GTO 17	190	RCL 02
85	" GLUB GLUB"	138	"GOOD JOB:	191	-
86	AVIEW	139	AVIEW	192	,5
87	TONE 8	140	BEEP	193	ST+ Z
88	TONE 6	141	PSE	194	+
89	TONE 4	142	LBL 17	195	R-P
90	TONE 2	143	"NEW GAME? <E>"	196	RTN
91	TONE 0	144	AVIEW	197	LBL 03
92	CF 08	145	TONE 8	198	E1
93	SF 11	146	TONE 5	199	X>Y?
94	PSE	147	CF 08	200	X<>Y
95	OFF	148	STOP	201	X<0?
96	GTO 15	149	GTO E	202	CLX
97	LBL A	150	LBL 18	203	RTN
98	FC? 08	151	"MISSED"	204	LBL C
99	GTO 17	152	AVIEW	205	FIX 0
100	" * "	153	TONE 0	206	XEQ 04
101	>" "	154	TONE 0	207	FS? 08
102	AVIEW	155	360	208	GTO 16
103	ISG 05	156	XEQ 01	209	GTO 17
104	CLX	157	ENTER^	210	LBL 04
105	XEQ 02	158	FRC	211	SF 07
106	FC? 01	159	P-R	212	RCL 05
107	X^2	160	ST+ 02	213	XEQ 05
108	E	161	X<>Y	214	RCL 06
109	XEQ 01	162	RCL 03	215	LBL 05
110	X<=Y?	163	+	216	CLA
111	GTO 19	164	XEQ 03	217	ARCL X
112	"BOOOM"	165	STO 03	218	>" "
113	AVIEW	166	RCL 02	219	FC?C 07
114	TONE 5	167	XEQ 03	220	>"S"
115	TONE 5	168	STO 02	221	>"CAN"

222 E	251 /	280 TONE 6
223 X#Y?	252 XEQ 07	281 PSE
224 >"S"	253 GTO 17	282 END
225 AVIEW	254 LBL 06	
226 PSE	255 >"ST GAME="	1 <u>LBL "GRID"</u>
227 PSE	256 AVIEW	2 CF 12
228 RTN	257 INT	3 SF 21
229 LBL D	258 LASTX	4 CF 29
230 FS? 08	259 FRC	5 FIX 0
231 GTO 16	260 LBL 07	6 ,0901
232 FIX 0	261 E5	7 LBL 01
233 "NO./GAMES="	262 *	8 " "
234 ARCL 00	263 " "	9 INT
235 XEQ 08	264 ARCL X	10 X=0?
236 "WOR"	265 >"CAN"	11 >" "
237 RCL 08	266 E	12 ARCL X
238 XEQ 06	267 X#Y?	13 LASTX
239 "BE"	268 >"S"	14 >"==0-1-2-3-4-5"
240 RCL 09	269 >" "	15 >"-6-7-8-9"
241 XEQ 06	270 RCL Z	16 PRA
242 "AVERAGE="	271 ARCL X	17 ISG X
243 AVIEW	272 >SCAN"	18 GTO 01
244 FIX 1	273 X#Y?	19 ADV
245 RCL 01	274 >"S"	20 ADV
246 INT	275 >" "	21 ADV
247 LASTX	276 PSE	22 ADV
248 FRC	277 LBL 08	23 ADV
249 RCL 00	278 AVIEW	24 END
250 ST/ Z	279 TONE 8	

Submarine Hunt, v5.

Wodunit – MoHP Disks

Another apocryphal version from the Museum of HP Calculators Disks... And of course undocumented again, which is a shame – the listing shows a rich feedback and a pretty elaborate scheme. As always, feel free to replace those global labels with mute ones.

Program listing:

1 <u>LBL "SUBS"</u>	43 AVIEW	85 RND
2 LBL e	44 PSE	86 STO 11
3 RCL 00	45 LBL C	87 LBL 01
4 CLRG	46 "BEARING"	88 RCL 08
5 STO 00	47 RCL 05	89 INT
6 FIX 0	48 RCL 04	90 RCL 11
7 CF 29	49 R-P	91 X=Y?
8 "YOUR NAME?"	50 X<>Y	92 GTO 02
9 AON	51 10	93 X<>Y
10 PROMPT	52 /	94 4
11 ASTO 01	53 RND	95 /
12 AOFF	54 10	96 INT
13 ,4	55 *	97 LASTX
14 STO 02	56 ARCL X	98 X=Y?
15 4	57 AVIEW	99 GTO 03
16 STO 03	58 PSE	100 RCL 08
17 <u>LBL "START"</u>	59 "RANGE="	101 INT
18 XEQ "RNG"	60 X<>Y	102 12
19 5 E3	61 3	103 –
20 *	62 /	104 X<0?
21 2 E3	63 E2	105 GTO 04
22 +	64 /	106 >"+"
23 STO 04	65 RND	107 GTO 05
24 XEQ "RNG"	66 E2	108 LBL 02
25 5 E3	67 *	109 >"a"
26 *	68 ARCL X	110 GTO 05
27 2 E3	69 AVIEW	111 LBL 03
28 +	70 RTN	112 3
29 STO 05	71 LBL D	113 –
30 XEQ "RNG"	72 "PEAKING"	114 ABS
31 10	73 AVIEW	115 ARCL X
32 *	74 CLA	116 GTO 05
33 RCL 06	75 1,023	117 LBL 04
34 –	76 STO 08	118 >"-"
35 STO 06	77 RCL 05	119 LBL 05
36 XEQ "RNG"	78 RCL 04	120 ISG 08
37 10	79 R-P	121 GTO 01
38 *	80 X<>Y	122 AVIEW
39 STO 07	81 ,4	123 30
40 LBL G	82 *	124 LBL F
41 "SPEED="	83 12	125 STO 08
42 ARCL 10	84 +	126 RCL 10

127 *	180 GTO 06	231 "NICE SHOOTING,
128 ST- 04	181 "THEY OPEN FIRE"	"
129 RCL 08	182 AVIEW	232 ARCL 01
130 GTO "TIME"	183 R^	233 AVIEW
131 <u>LBL "RNG"</u>	184 SF 06	234 PSE
132 RCL 00	185 XEQ F	235 GTO "START"
133 9821	186 XRQ "RNG"	236 LBL A
134 *	187 10	237 STO 08
135 ,2211327	188 *	238 RCL 10
136 +	189 INT	239 +
137 FRC	190 STO 11	240 45
138 STO 00	191 LBL 07	241 X<Y?
139 RTN	192 SF 06	242 GTO G
140 LBL E	193 30	243 RCL 08
141 "FIRING 1-4"	194 XEQ F	244 2
142 AVIEW	195 XEQ "RNG"	245 /
143 RCL 07	196 "BOOOOM"	246 ABS
144 RCL 06	197 AVIEW	247 STO 11
145 P-R	198 RCL 04	248 RCL 10
146 CHS	199 *	249 +
147 65	200 2 E3	250 RCL 11
148 +	201 X<=Y?	251 *
149 X<>Y	202 GTO 08	252 RCL 05
150 /	203 "POWIE YOU'RE	253 RCL 04
151 RCL 05	HIT"	254 P-R
152 RCL 04	204 > , "	255 RCL Z
153 P-R	205 ARCL 01	256 -
154 RDN	206 AVIEW	257 R-P
155 *	207 PSE	258 STO 04
156 CHS	208 "DOWN YOU GO"	259 X<>Y
157 R^	209 0	260 STO 05
158 -	210 STO 10	261 RCL 08
159 65	211 AVIEW	262 ST+ 10
160 RCL 10	212 PSE	263 RCL 11
161 -	213 DSE 03	264 <u>LBL "TIME"</u>
162 /	214 GTO "START"	265 ST+ 09
163 STO 11	215 GTO "DOWN"	266 RCL 07
164 ,003	216 LBL 08	267 RCL 06
165 99	217 "SPLASH"	268 RCL Z
166 RCL 06	218 AVIEW	269 *
167 /	219 DSE 11	270 P-R
168 ABS	220 GTO 07	271 X<>Y
169 LBL 06	221 GTO C	272 RCL 05
170 30	222 LBL 09	273 RCL 04
171 RCL Z	223 SF 06	274 P-R
172 INT	224 LASTX	275 R^
173 *	225 XEQ F	276 +
174 RCL 11	226 CF 06	277 RDN
175 ABS	227 "THEY SINK"	278 +
176 X<=Y?	228 AVIEW	279 R^
177 GTO 09	229 PSE	280 R-P
178 RDN	230 ISG 02	281 STO 04
179 ISG Y		282 X<>Y

283 STO 05
284 FC? 06
285 GTO C
286 RTN
287 LBL B
288 FC?C 22
289 CLX
290 STO 08
291 "TOO SLOW"
292 RCL 10
293 10
294 X>Y?
295 PROMPT
296 *
297 3
298 *
299 3
300 *
301 P-R
302 X<>Y
303 CHS
304 RCL 05
305 RCL 04
306 P-R
01

307 R^
308 -
309 RDN
310 +
311 R^
312 R-P
313 STO 04
314 X<>Y
315 RCL 08
316 -
317 STO 05
318 RCL 08
319 STO 07
320 RCL 10
321 30
322 *
323 PI
324 *
325 RCL 08
326 *
327 RCL 10
328 /
329 180
330 /

331 ABS
332 GTO "TIME"
333 LBL "DOWN"
334 "LAST SUB GONE"
335 AVIEW
336 PSE
337 "SCORE= "
338 RCL 02
339 INT
340 ARCL X
341 AVIEW
342 PSE
343 "TIME= "
344 RCL 09
345 3600
346 /
347 HMS
348 FIX 4
349 ARCL X
350 AVIEW
351 SF 29
352 FIX 3
353 END

NFL American Football

Whodunit – Swap Disks

This seems to be a very detailed and comprehensive simulation, but unfortunately no documentation is available. A real shame... :-(

Program listing:

1. LBL "FB3"	38. XEQ 98	77. SF 17	115. .3
2. LBL 81	39. SF 09	78. <u>LBL 00</u>	116. ST+ 08
3. CLRG	40. <u>LBL 17</u>	79. RCL 08	117. 0
4. SF 27	41. CF IND 02	80. 15	118. FS?C 06
5. TIME	42. FC?C 00	81. X>Y?	119. -5
6. 40	43. SF 00	82. GTO 01	120. FS?C 10
7. /	44. <u>LBL 01</u>	83. RCL 09	121. -2
8. FRC	45. XEQ 19	84. 2	122. FS?C 11
9. STO 00	46. XEQ 21	85. X#Y?	123. 6
10. CF 29	47. XEQ 89	86. GTO 00	124. STO 01
11. CF 16	48. <u>LBL 16</u>	87. CF 00	125. XEQ 97
12. CF 17	49. SF IND 02	88. "GAME OVER"	126. 97
13. FIX 00	50. 5	89. XEQ 98	127. X<=Y?
14. CF 06	51. RCL 02	90. STOP	128. GTO 40
15. CF 07	52. X<Y?	91. GTO 81	129. RDN
16. CF 10	53. GTO 27	92. LBL 00	130. 7
17. CF 11	54. XEQ 95	93. 1	131. /
18. "COIN FLIP"	55. XEQ 89	94. ST+ 09	132. 3
19. XEQ 98	56. SF IND 02	95. TONE 05	133. -
20. 1	57. <u>LBL 27</u>	96. 0	134. RCL 01
21. STO 02	58. "BALL ON "	97. STO 08	135. +
22. STO 09	59. ARCL 06	98. <u>LBL 01</u>	136. STO 07
23. CF 00	60. XEQ 98	99. FS? 17	137. GTO 90
24. CF 01	61. ARCL 05	100. XEQ 84	138. LBL B
25. CF 02	62. ">" YDS TO GO"	101. FC? 07	139. XEQ 76
26. CF 03	63. XEQ 98	102. GTO 01	140. LBL 62
27. CF 04	64. FS?C 08	103. FS? 00	141. "DRAW"
28. XEQ 97	65. RTN	104. GTO 18	142. XEQ 98
29. 50	66. <u>LBL 15</u>	105. <u>LBL 01</u>	143. .3
30. X>Y?	67. 1	106. SF 16	144. ST+ 08
31. GTO 26	68. RCL 09	107. "PLAY ?"	145. -2
32. <u>LBL 24</u>	69. X=Y?	108. PROMPT	146. FS?C 06
33. "HOME KICKS"	70. GTO 00	109. GTO 15	147. -9
34. XEQ 98	71. 3	110. LBL A	148. FS?C 10
35. GTO 01	72. X=Y?	111. XEQ 76	149. 7
36. <u>LBL 26</u>	73. GTO 00	112. <u>LBL 61</u>	150. FS?C 11
37. "VIS KICKS"	74. RCL 08	113. "UP MIDDLE"	151. 4
	75. 13	114. XEQ 98	152. STO 01
	76. X<=Y?		153. XEQ 97
			154. 95

155. X<=Y?	207. XEQ 97	260. 61	312. GTO 45
156. GTO 40	208. 45	261. X<=Y?	313. RDN
157. RDN	209. X<=Y?	262. GTO 60	314. 37
158. 1.6	210. GTO 03	263. LBL 03	315. X<=Y?
159. Y^X	211. RDN	264. XEQ 97	316. GTO 29
160. 30	212. 10	265. .2	317. RDN
161. /	213. X<=Y?	266. *	318. 9
162. 10	214. GTO 60	267. 3	319. X<=Y?
163. -	215. RDN	268. -	320. GTO 60
164. RCL 01	216. 4	269. RCL 01	321. GTO 50
165. +	217. X<=Y?	270. +	322. LBL 01
166. STO 07	218. GTO 50	271. STO 07	323. 65
167. GTO 90	219. GTO 45	272. RCL 06	324. X<=Y?
168. LBL C	220. LBL 00	273. 93	325. GTO 29
169. XEQ 76	221. 1	274. X>Y?	326. RDN
170. LBL 63	222. STO 01	275. GTO 00	327. 55
171. "SWEEP"	223. XEQ 97	276. 3	328. X<=Y?
172. XEQ 98	224. 90	277. ST- 07	329. GTO 50
173. .3	225. X<=Y?	278. LBL 00	330. RDN
174. ST+ 08	226. GTO 45	279. .1	331. 27
175. 2	227. RDN	280. ST+ 08	332. X<=Y?
176. FS?C 06	228. 85	281. RCL 07	333. GTO 45
177. -4	229. X<=Y?	282. GTO 90	334. GTO 60
178. FS?C 10	230. GTO 50	283. LBL E	335. LBL 02
179. -10	231. RDN	284. XEQ 76	336. 77
180. FS?C 11	232. 60	285. LBL 65	337. X<=Y?
181. 7	233. X<=Y?	286. "LONG PASS"	338. GTO 50
182. STO 01	234. GTO 60	287. XEQ 98	339. RDN
183. XEQ 97	235. GTO 03	288. .3	340. 75
184. 3	236. LBL 01	289. ST+ 08	341. X<=Y?
185. /	237. -1	290. XEQ 97	342. GTO 45
186. 8.8	238. STO 01	291. FS?C 06	343. RDN
187. -	239. XEQ 97	292. GTO 00	344. 23
188. RCL 01	240. 70	293. FS?C 10	345. X<=Y?
189. +	241. X<=Y?	294. GTO 01	346. GTO 60
190. STO 07	242. GTO 60	295. FS?C 11	347. LBL 29
191. GTO 90	243. RDN	296. GTO 02	348. XEQ 97
192. LBL D	244. 20	297. 67	349. .8
193. XEQ 76	245. X<=Y?	298. X<=Y?	350. *
194. LBL 64	246. GTO 03	299. GTO 60	351. RCL 06
195. "SHORT PASS"	247. RDN	300. RDN	352. .6
196. XEQ 98	248. 12	301. 54	353. *
197. .2	249. X<=Y?	302. X<=Y?	354. -
198. ST+ 08	250. GTO 50	303. GTO 45	355. 8
199. FS?C 06	251. GTO 45	304. RDN	356. X>Y?
200. GTO 00	252. LBL 02	305. 42	357. GTO 29
201. FS?C 10	253. -1	306. X<=Y?	358. RDN
202. GTO 01	254. STO 01	307. GTO 50	359. STO 07
203. FS?C 11	255. XEQ 97	308. GTO 29	360. .2
204. GTO 02	256. 91	309. LBL 00	361. ST+ 08
205. 0	257. X<=Y?	310. 84	362. RCL 07
206. STO 01	258. GTO 50	311. X<=Y?	363. GTO 90
	259. RDN		364. LBL F

365. "PUNT"	417. GTO 14	468. FS?C 11	520. CF 06
366. XEQ 98	418. RDN	469. 4	521. CF 10
367. .2	419. 27	470. STO 01	522. 1
368. ST+ 08	420. X>Y?	471. XEQ 97	523. FS?C 11
369. XEQ 97	421. GTO 14	472. 95	524. .5
370. 97	422. RDN	473. X<=Y?	525. STO 01
371. X<=Y?	423. ARCL X	474. GTO 40	526. .3
372. GTO 70	424. >" YDS"	475. RDN	527. ST+ 08
373. RDN	425. XEQ 98	476. 85	528. XEQ 97
374. .3	426. RCL 06	477. X<=Y?	529. 98
375. *	427. +	478. GTO 50	530. X<=Y?
376. 32	428. 110	479. RDN	531. GTO 30
377. +	429. X>Y?	480. 65	532. RDN
378. RND	430. GTO 01	481. X<=Y?	533. 90
379. ARCL X	431. XEQ 97	482. GTO 60	534. X<=Y?
380. >" YDS"	432. 80	483. XEQ 97	535. GTO 50
381. XEQ 98	433. X<=Y?	484. 3	536. RDN
382. RCL 06	434. GTO 00	485. /	537. 65
383. +	435. "GOOD"	486. 8	538. X<=Y?
384. E2	436. XEQ 98	487. -	539. GTO 40
385. X<=Y?	437. TONE 09	488. RCL 01	540. XEQ 97
386. GTO 55	438. 3	489. +	541. X^2
387. RDN	439. XEQ 99	490. STO 07	542. 90
388. E2	440. XEQ 34	491. GTO 90	543. /
389. -	441. GTO 17	492. LBL 01	544. 20
390. ABS	442. LBL 00	493. XEQ 97	545. -
391. STO 06	443. "WIDE"	494. 97	546. RCL 01
392. "BALL ON "	444. XEQ 98	495. X<=Y?	547. *
393. ARCL X	445. LBL 01	496. GTO 40	548. STO 07
394. XEQ 98	446. "NO	497. RDN	549. GTO 90
395. FC?C 00	GOOD"	498. 87	550. LBL 33
396. SF 00	447. XEQ 98	499. X<=Y?	551. "QTR. = "
397. 3	448. TONE 00	500. GTO 50	552. ARCL 09
398. RCL 02	449. RCL 06	501. RDN	553. XEQ 98
399. X>Y?	450. 80	502. 72	554. GTO 00
400. XEQ 21	451. X<=Y?	503. X<=Y?	555. LBL 84
401. CF IND 02	452. GTO 55	504. GTO 60	556. SF 08
402. XEQ 89	453. XEQ 95	505. RDN	557. LBL 00
403. GTO 16	454. XEQ 89	506. 67	558. 15
404. LBL G	455. GTO 16	507. X<=Y?	559. RCL 08
405. "F. G.	456. LBL H	508. GTO 45	560. -
ATTEMPT"	457. XEQ 76	509. XEQ 97	561. INT
406. XEQ 98	458. LBL 66	510. 2.5	562. "TIME= "
407. ,2	459. "SCREEN	511. /	563. ARCL X
408. ST+ 08	PASS"	512. 7	564. >:"
409. LBL 14	460. XEQ 98	513. -	565. LASTX
410. XEQ 97	461. .3	514. STO 07	566. FRC
411. 95	462. ST+ 08	515. GTO 90	567. HMS
412. X<=Y?	463. FS?C 10	516. LBL I	568. 100
413. GTO 70	464. GTO 01	517. XEQ 76	569. *
414. RDN	465. 0	518. "TRICK	570. INT
415. 55	466. FS?C 06	PLAY"	571. ARCL X
416. X<=Y?	467. -4	519. XEQ 98	572. XEQ 98

573. FS?C 08	625. RCL 02	677. XEQ 95	726. LBL 89
574. RTN	626. X#0?	678. XEQ 21	727. 90
575. GTO 15	627. CF IND 02	679. XEQ 89	728. RCL 06
576. LBL 34	628. 1	680. GTO 16	729. X<=Y?
577. "V= "	629. ST+ 02	681. LBL 55	730. GTO 00
578. ARCL 03	630. GTO 16	682. "TOUCHBA	731. 100
579. ">" H="	631. LBL 01	CK"	732. RCL 06
580. ARCL 04	632. "LOSE"	683. XEQ 98	733. -
581. XEQ 98	633. XEQ 98	684. XEQ 95	734. STO 05
582. RTN	634. RCL 02	685. XEQ 89	735. GTO 01
583. LBL J	635. X=0?	686. 20	736. LBL 00
584. XEQ 34	636. 1	687. STO 06	737. 10
585. SF 08	637. STO 02	688. FS?C 00	738. STO 05
586. XEQ 27	638. RCL 06	689. SF 00	739. LBL 01
587. SF 08	639. 100	690. GTO 16	740. CF IND 02
588. XEQ 33	640. -	691. LBL 60	741. 1
589. GTO 15	641. ABS	692. "INCOMPL	742. STO 02
590. LBL 30	642. STO 06	ETE"	743. RTN
591. "*TOUCHD	643. XEQ 89	693. XEQ 98	744. LBL 90
OWN*"	644. FC?C 00	694. CF IND 02	745. RND
592. XEQ 98	645. SF 00	695. 1	746. X<=0?
593. BEEP	646. GTO 16	696. ST+ 02	747. GTO 00
594. XEQ 80	647. LBL 45	697. GTO 16	748. 1
595. GTO 17	648. "SACK"	698. LBL 70	749. X<>Y
596. LBL 35	649. XEQ 98	699. "BLOCKED"	750. X<=Y?
597. "SAFETY"	650. XEQ 97	700. XEQ 98	751. 0
598. XEQ 98	651. 10	701. XEQ 95	752. LBL 00
599. TONE 00	652. /	702. XEQ 89	753. STO 07
600. 2	653. INT	703. GTO 16	754. ST+ 06
601. FS? 00	654. CHS	704. LBL 80	755. RCL 05
602. GTO 00	655. STO 07	705. "X. P.	756. RCL 07
603. ST+ 04	656. GTO 90	ATTEMPT"	757. -
604. GTO 01	657. LBL 50	706. XEQ 98	758. X>0?
605. LBL 00	658. "INTERCEP	707. XEQ 97	759. GTO 01
606. ST+ 03	TION"	708. 80	760. 90
607. LBL 01	659. XEQ 98	709. X>Y?	761. RCL 06
608. XEQ 34	660. XEQ 97	710. GTO 01	762. X<=Y?
609. GTO 17	661. .3	711. "NO	763. GTO 00
610. LBL 41	662. *	GOOD"	764. 100
611. CF IND 02	663. 5	712. XEQ 98	765. RCL 06
612. 0	664. -	713. TONE 00	766. -
613. STO 02	665. RND	714. 6	767. STO 05
614. 10	666. RCL 06	715. XEQ 99	768. GTO 07
615. STO 05	667. +	716. XEQ 34	769. LBL 00
616. LBL 40	668. 100	717. RTN	770. 10
617. "FUMBLE"	669. X<=Y?	718. LBL 01	771. STO 05
618. XEQ 98	670. GTO 55	719. "GOOD"	772. LBL 07
619. XEQ 97	671. RDN	720. XEQ 98	773. CF IND 02
620. 50	672. STO 06	721. TONE 09	774. 1
621. X<=Y?	673. "AT "	722. 7	775. STO 02
622. GTO 01	674. ARCL X	723. XEQ 99	776. GTO 02
623. "KEEP"	675. ">" YDL"	724. XEQ 34	777. LBL 01
624. XEQ 98	676. XEQ 98	725. RTN	778. STO 05

779. CF IND 02	828. "BALL ON "	876. GTO 01	925. X>Y?
780. 1	829. ARCL 06	877. XEQ 97	926. GTO F
781. ST+ 02	830. >"YL"	878. 60	927. LBL 00
782. LBL 02	831. XEQ 98	879. X<=Y?	928. CF 22
783. ARCL 07	832. RTN	880. GTO 91	929. "DEF. 0, 1,
784. >" YDS"	833. LBL 21	881. RDN	2, 3"
785. XEQ 98	834. "RETURN"	882. 40	930. AVIEW
786. RCL 06	835. XEQ 98	883. X<=Y?	931. LBL 87
787. 100	836. .2	884. GTO 92	932. PSE
788. X<=Y?	837. ST+ 08	885. RDN	933. FC?C 22
789. GTO 30	838. XEQ 97	886. 20	934. GTO 87
790. 0	839. 98	887. X<=Y?	935. 4
791. RCL 06	840. X<=Y?	888. GTO 93	936. X<=Y?
792. X<=Y?	841. GTO 30	889. GTO 94	937. GTO 18
793. GTO 35	842. RDN	890. LBL 00	938. RDN
794. GTO 16	843. 94	891. XEQ 97	939. 91
795. LBL 95	844. X<=Y?	892. 66	940. +
796. FC?C 00	845. GTO 41	893. X<=Y?	941. XEQ IND X
797. SF 00	846. RDN	894. GTO 93	942. RTN
798. RCL 06	847. X^2	895. RDN	943. LBL 91
799. 100	848. ,005	896. 33	944. "4 - 3"
800. -	849. *	897. X<=Y?	945. XEQ 98
801. ABS	850. RND	898. GTO 94	946. RTN
802. STO 06	851. X<0?	899. GTO 91	947. LBL 92
803. RTN	852. 0	900. LBL 01	948. "STACK"
804. LBL 98	853. ARCL X	901. XEQ 97	949. XEQ 98
805. AVIEW	854. >"YD	902. 50	950. SF 06
806. PSE	RUNBACK"	903. X<=Y?	951. RTN
807. CLA	855. XEQ 98	904. GTO 91	952. LBL 93
808. RTN	856. RCL 06	905. GTO 92	953. "BLITZ"
809. LBL 99	857. +	906. LBL 18	954. XEQ 98
810. FS? 00	858. 100	907. FS?C 16	955. SF 10
811. GTO 00	859. X<=Y?	908. GTO 00	956. RTN
812. ST+ 03	860. GTO 30	909. 4	957. LBL 94
813. GTO 01	861. RDN	910. RCL 02	958. "PREVENT"
814. LBL 00	862. STO 06	911. X#Y?	959. XEQ 98
815. ST+ 04	863. RTN	912. GTO 00	960. SF 11
816. LBL 01	864. LBL 76	913. 1	961. RTN
817. RTN	865. FC? 07	914. RCL 05	962. LBL 97
818. LBL 19	866. GTO 18	915. X=Y?	963. RCL 00
819. "KICKOFF"	867. RCL 02	916. GTO 01	964. 997
820. XEQ 98	868. RCL 05	917. RCL 06	965. *
821. ,2	869. +	918. 59	966. FRC
822. ST+ 08	870. 12	919. X<=Y?	967. STO 00
823. XEQ 97	871. X<=Y?	920. GTO G	968. E2
824. ,2	872. GTO 00	921. GTO F	969. *
825. *	873. RDN	922. LBL 01	970. INT
826. RND	874. 7	923. RCL 06	971. END
827. STO 06	875. X>Y?	924. 42	

Car Racing.

Martin Meyer, PRISMA 89/3 p29

Even though I'm not a friend of computer games, it's been a hell of a long time. that such "useful" stuff in PRISMA showed up for the last time.

For the computer or programming engineer however, such programs are quite useful in order to be able to enter the abilities of his new acquisition better, and to become "playfully" familiar with the new matter.

The following program acts as follows it is a quite simple version of Pocket-size car race, in which anis fixed.

I've recorded this one, more or less winding route, in which the "steep curves" with two lines, dotted and dashed, marked are: -. -.-.-

These so-called "steep bends" leave of course at much higher speeds than the normal ones. The goal of the game consists of to keep the course as high as possible average speed, without flying out of a bend.

You will see to stay on track isn't that easy, you have to do something similar like normal life with gears, the throttle and the brake, a certain inertia of the vehicle's reaction belongs to lifelike simulation, so hectic reactions are pointless!

I have discarded adding any acoustic background, this is usually rather disturbing.

Now to the operation of this work of art:

The program is loaded into the memory and started. first the message "TEMPO:0KM/H" appears. one second later then "POSITION 0.0".

After two seconds viewing pause the display shows the prompt "GANG =", here begins the actual Input of the vehicle data, the respective can take the values 1-5. The upper 5 keys of the HP41 are used here by is assigned the numbers 1-5, i.e. A=1, B=2, ... E=5. All other keys cause the entry of 0.

The calculator waits until I press a key the confirmation will appear in form of "GANG=2" when I press the key B had pressed.

Now I'm prompted with "GAS=" for the input the accelerator pedal to stop me now press the E key (5), i.e. I'm going full throttle. The display now shows short "GAS=".

Last but not least the calculator wants to use know if I intend to brake, in which The display shows "BREMSE=". I press any other key, e.g. the ENTER key, appears briefly "BRAKE=O."

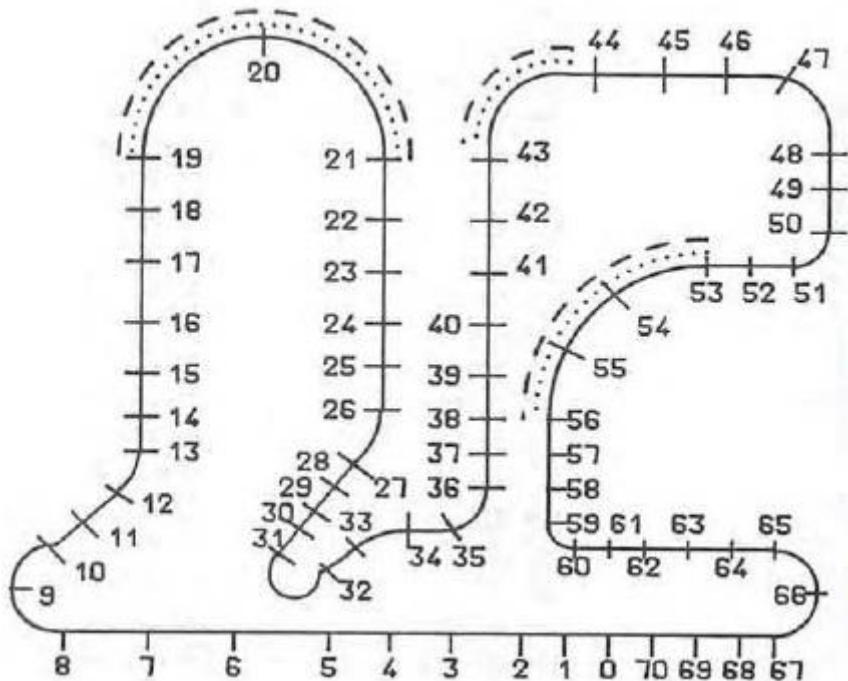
After a short pause for reflection we get we shared the consequences of our wishes:

TEMPO:54KM/H
POSITION:0.8

After a short pause of about two seconds, it follows again the renewed request for the Driver's wishes:

GEAR=3 [C]
 GAS=5 [E]
 BRAKE=0 [ENTER^]

The result is: TEMPO:162KM/H ; POSFTION:3.0



As you can see, the vehicle can be accelerated well, so be careful with the Gas. Because if we're not careful now, then we'll go in the upcoming Curve discreetly straight ahead ...

The next entries:

GEAR=4
 GAS=3
 BRAKE=0

have resulted into: TEMPO:151KM/H ; POSITION:5.1

Now we're going full throttle for fun out of the curve:

GEAR=4
 GAS=5
 BRAKE=0

The consequences of this reckless driving are immediately presented, the display shows
****CRASH****

This is nothing more than the sad news, that we are at position 8.1 from the I'm sorry I threw you into a bend. I finally wanted not to demonstrate how to optimize the course masters, everybody has to try it for himself.

The race starts after this mishap again from position 0.0, the end of the nervous strains is only for the complete driving through of the race course so you don't have to pinch.

If somebody should manage the course after a line of any length, it will be successful by the display AVERAGE SPEED; one second pause: XXXKM/H rewarded. Fanfares to the award ceremony gives none of them can make up their own words.

Should anyone think about the lines 250 and 251 wonder, they serve only the time the display to indicate the flashing of the the --CRASH-- display.

Have fun driving!

Program listing:

01*LBL "RACE"	38 AVIEW	75*LBL 01
02*LBL 06	39*LBL a	76 RCL 06
03 sREG 08	40 GETKEY	77 RCL 07
04 CLs	41 X=0?	78 *
05 sREG 00	42 GTO a	79 10.8
06 CLs	43 XEQ A	80 *
07 SIZE?	44 X<=0?	81 RCL 08
08 12	45 E	82 X>Y?
09 X>Y?	46 STO 07	83 XEQ 02
10 PSIZE	47 "GAS="	84 X<>Y
11 RCLFLAG	48 AVIEW	85 RCL 10
12 STO 11	49*LBL b	86 10.8
13 5.012	50 GETKEY	87 *
14*LBL 13	51 X=0?	88 -
15 CF IND X	52 GTO b	89 X<=0?
16 ISG X	53 XEQ A	90 0
17 GTO 13	54 STO 06	91 STO 08
18 CF 29	55 "BREAKS="	92 s+
19 FIX 0	56 AVIEW	93 RCL 08
20 E	57*LBL c	94 72
21 STO 07	58 GETKEY	95 /
22*LBL J	59 X=0?	96 ST+ 09
23 "TEMPO:"	60 GTO c	97 RCL 09
24 ARCL 08	61 XEQ A	98 71
25 "" KM/H"	62 STO 10	99 X<=Y?
26 AVIEW	63 XEQ 01	100 GTO 11
27 PSE	64 GTO J	101 X<>Y
28 PSE	65*LBL A	102 65
29 FIX 1	66 5	103 X<Y?
30 "POSITON "	67 X<>Y	104 GTO 08
31 ARCL 09	68 10	105 X<>Y
32 AVIEW	69 -	106 59
33 PSE	70 X>Y?	107 X<Y?
34 PSE	71 CLX	108 GTO 03
35 PSE	72 ARCL X	109 X<>Y
36 FIX 0	73 AVIEW	110 53
37 "GEAR="	74 RTN	111 X<Y?

112 GTO 10	161 ENTER^	210*LBL 09
113 X<>Y	162 RTN	211 FS? 10
114 50	163*LBL 03	212 RTN
115 X<Y?	164 FS? 05	213 SF 10
116 GTO 03	165 RTN	214 CF 09
117 X<>Y	166 SF 05	215 CF 05
118 47	167 RCL 08	216 RCL 08
119 X<Y?	168 70	217 190
120 GTO 08	169 X<Y?	218 X<Y?
121 X<>Y	170 GTO 14	219 GTO 14
122 43	171 RTN	220 RTN
123 X<Y?	172*LBL 04	221*LBL 10
124 GTO 09	173 FS? 06	222 FS? 12
125 X<>Y	174 RTN	223 RTN
126 35	175 SF 06	224 SF 12
127 X<Y?	176 RCL 08	225 CF 05
128 GTO 08	177 150	226 CF 09
129 X<>Y	178 X<Y?	227 RCL 08
130 33	179 GTO 14	228 220
131 X<Y?	180 RTN	229 X<Y?
132 GTO 04	181*LBL 05	230 GTO 14
133 X<>Y	182 FS? 07	231 RTN
134 31	183 RTN	232*LBL 11
135 X<Y?	184 SF 07	233 MEAN
136 GTO 07	185 CF 06	234 "AVRG. TEMPO"
137 X<>Y	186 RCL 08	235 AVIEW
138 26	187 110	236 PSE
139 X<Y?	188 X<Y?	237 CLA
140 GTO 04	189 GTO 14	238 ARCL X
141 X<>Y	190 RTN	239 "" KM/H"
142 19	191*LBL 07	240 RCL 11
143 X<Y?	192 FS? 08	241 STOFLAG
144 GTO 05	193 RTN	242 PROMPT
145 X<>Y	194 SF 08	243*LBL 14
146 12	195 CF 06	244 6
147 X<Y?	196 RCL 08	245 FIX 1
148 GTO 04	197 40	246*LBL 12
149 X<>Y	198 X<Y?	247 CLA
150 8	199 GTO 14	248 AVIEW
151 X<Y?	200 RTN	249 HMS
152 GTO 03	201*LBL 08	250 HR
153 RTN	202 FS? 09	251 "***CRASH***"
154*LBL 02	203 RTN	252 ARCL 09
155 X<>Y	204 SF 09	253 AVIEW
156 -	205 RCL 08	254 DSE X
157 3	206 90	255 GTO 12
158 /	207 X<Y?	256 GTO 06
159 ST- 08	208 GTO 14	257 END
160 RCL 08	209 RTN	

ZCAR, an Action Game

Cary E. Reinstein, PPCCJ V11N1 p20; (Jan/Feb 1984)

"ZCAR" is an action game that simulates driving a car at high speed through five laps of a slalom race course. Each lap has eight turns, all of which are randomly selected except the first one which is always straight. In order to "drive" the car you must shift gears appropriately as well as use the gas pedal and brakes. The display always shows the turns as seen through the windshield (see details below). The object is to finish the course as well as score as many points as possible though these aims are occasionally at variance because points scored depend upon how fast the car is driven and the faster the car is driven the less likely it is that the course will be finished. There are four possible outcomes to the game: the first is that all laps will be finished and a score will be displayed; second, the engine will be over-revved in an attempt to reach top speed or due to a gear-shifting error; third, engine revs will drop too low and the engine will stall; and lastly, a steering error, failure to reduce speed on an ess turn or to brake for a warning flag will cause a crash. Pressing any undefined key will cause the car to crash and terminate the game. Whenever the game ends, for whatever reason, the score is displayed along with the percent of the course finished.

An attempt was made to program the game to run as fast as possible (if the word "fast" applies at all to an HP-41) by minimizing the use of alpha strings and numeric constants and using Valentín Albillo's pseudorandom number generator, R-D FRC, an excellent one for games.

Start the game with [XEQ] "ZCAR". If a Time Module is not present the display will prompt for a random number seed. The input seed must be a fraction or mixed number. After inputting this seed press [R/S]. Don't press [R/S] at any other time.

GAME RULES :

Pressing any key not shown on the overlay causes the car to crash. When a turn appears in the display, press the top row key that will negotiate the turn; for example, to ,steer a hard left, press the [S+] key –see the display table and keyboard diagrams below for reference.

Do not rev the engine higher than 6000 RPM nor low enough in any gear to stall. 600 RPM or less will cause a stall in first gear and the other gear stall points vary according to the gear ratio.

The faster the car is driven the more paints will be scored to a maximum of 210 per lap.

To begin the race you must shift into first and then begin to rev the engine through its gears until the desired speed is achieved. You don't have to steer until you have reached the desired engine speed and gear. Decision time can be extended by pressing the key representing the gear you are already in which also causes the speedometer and tachometer to be shown.

Gears must be shifted in the natural order; attempting to jump a gear will result in a blown transmission.

At a random point in each lap two race officials will drop a warning flag -- You must brake to avoid a crash. Tapping the brakes or downshifting will also cause a crash and terminate the race.

You must lower your speed on the ess turns to avoid a crash. Driving in fourth gear will lose points on these turns.

TECHNICAL DETAILS:

Point scoring:

Each lap gives a maximum of 210 points, 5 turns at 28 points each and two ess turns at 35 points each. Maximum points are scored for- an ess turn at 4000 rpm in third gear and the greatest penalty is a loss of 28 points at 6000 rpm in third to 21 points at the same rpm in fourth.

Pedal pressure and RPM:

After using the gas pedal or brakes the rpm will vary between 47 and 400 rpm each turn depending upon the pedal pressure applied. There are three pedal pressures, tap, medium and stamp which alter the rpm respectively by 14 - 1350 rpm, 21 - 2140 rpm and 35 - 3500 rpm depending upon the size of the random number when the pedals are used. The tone pitch anticipates the random number change. If the pitch is high the pedal pressure will cause a higher percentage of rpm change.

Gear Ratios:

First, 3.5; second, 2.14; third, 1.36; fourth, 1.0
(gear ratios are similar to a Datsun Turbo 280ZX).

System requirements:

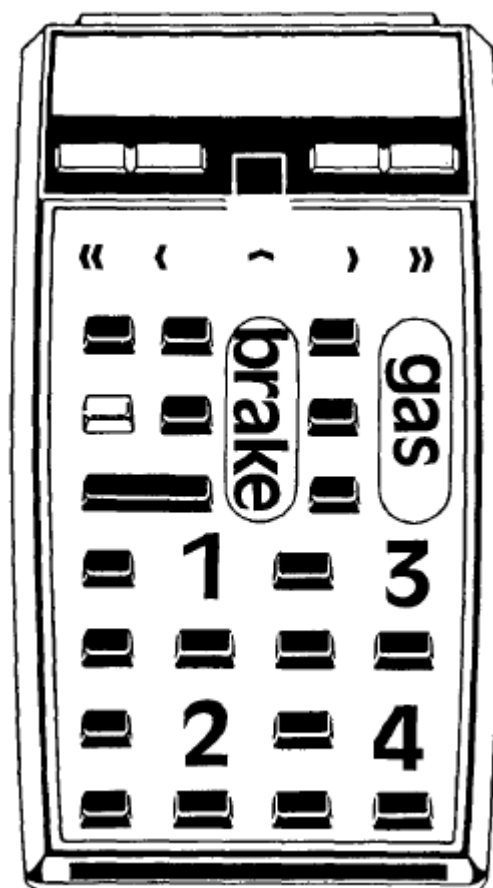
127 registers, 704 bytes of program memory and 26 registers for data. Extended Functions Module. Optionally, a Time Module. Lines 32, 35 and 36 can be omitted if a Time Module will not be used. Lines 32, 34, 36 and 37 can be omitted if the module will always be used.

Synthetic text and number strings used:

14	245, 49, 50, 92, 92, 92
16	245, 49, 51, 33, 33, 33
27	246, 51, 51, 1, 45, 45, 1
40	27, 19
151	27, 18
173	27, 17
197	28, 27
281	27, 19
307	241, 40
309	244, 127, 41, 32, 40
318	242, 127, 41
366.7	159, 28

DISPLAYS:

L L L	Hard Left	[Σ+]
\ \ \	Moderate Left	[1/X]
 	Straight	[SQRT]
/ / /	Moderate Right	[LOG]
7 7 7	Hard Right	[LN]
/ / 7 7	Right ess	[LOG],[LN]
S S S	Harpiness	[Σ+],[LN]
# - - #	Warning flag	[STO] or [CHS]
(nnn) (nnn)	Thacometer	Speedometer

KEYS USED :

Steering:	See Displays [Σ+] – [LN] above		
Brakes:	Tap [SIN],	Medium [STO],	Stomp [CHS]
Gas pedal:	Tap [TAN],	Medium [SST],	Stomp [←]
Gears:	First [7],	Second[1],	Third [9], Fourth [3]

Program listing:

01*LBL "ZCAR"	19 ASTO 13	37 *
02 27	20 ASTO 10	38 LASTX
03 XROM "INIT"	21 "14///"	39 +
04 PI	22 ASTO 14	40 STO 08
05 3.5	23 "15 777"	41 XEQ 09
06 STO 01	24 ASTO 15	42 CF 09
07 SIGN	25 "29//77"	43 5
08 STO 04	26 ASTO 16	44 STO 25
09 -	27 "26SSS"	45 CLX
10 STO 02	28 ASTO 17	46 STO 06
11 1.36	29 "33--"	47 STO 20
12 STO 03	30 ASTO 18	48 STO 24
13 X<>F	31 " "	49 STO 26
14 "11LLL"	32 ASTO 19	50 BEEP
15 ASTO 11	33 CF 21	<u>51*LBL 05</u>
16 "12\\\\"	34 RNG	52 CF 06
17 ASTO 12	35 STO 00	53 CF 08
18 "13!!!"	36 E3	54 CLA

55 ARCL IND 09	<u>108*LBL 05</u>	161 GETKEY
56 29	109 RCL 24	162 RCL 00
57 ANUM	110 *	163 R-D
58 STO 07	<u>111*LBL 08</u>	164 FRC
59 X=Y?	112 FC?C 07	165 E1
60 SF 05	113 ST+ 20	166 *
61 X=Y?	114 5	167 TONE IND X
62 SF 06	115 SQRT	168 X<>Y
63 X>Y?	116 ST+ 26	169 XEQ IND X
64 SF 07	117 RCL 23	170 FC? 25
65 X<>Y	118 4 E2	171 GTO 16
66 3	119 *	172 FS? 00
67 -	120 RCL 06	173 RTN
68 X=Y?	121 /	174 FS?C 09
69 SF 05	122 ST+ 08	175 GTO 06
70 ATOX	123 ISG 09	176 RTN
71 ATOX	124 GTO 05	<u>177*LBL 11</u>
72 ARCL 19	125 "LAP "	<u>178*LBL 12</u>
73 -4	126 RCL 25	<u>179*LBL 13</u>
74 AROT	127 CHS	<u>180*LBL 14</u>
75 ARCL 19	128 7	<u>181*LBL 15</u>
76 AVIEW	129 +	182 FS? 00
77 CLX	130 ARCL X	183 GTO 17
78 STO 05	131 >"..."	184 ST+ 05
79 XEQ 06	132 RCL 20	185 13
80 FS? 00	133 LASTX	186 X<>Y
81 GTO 07	134 *	187 -
82 FS? 05	135 ARCL X	188 FS? 06
83 XEQ 06	136 DSE 25	189 - E
84 FS? 00	137 GTO 08	190 AROT
85 GTO 07	138 BEEP	191 AROT
86 RCL 05	<u>139*LBL 07</u>	192 AVIEW
87 RCL 07	140 "*" "	193 FC? 05
88 X#Y?	141 RCL 20	194 RTN
89 XEQ 16	142 7	195 FC? 08
90 FS? 00	143 *	196 GTO 16
91 GTO 07	144 ARCL X	197 RTN
92 RCL 08	145 "' * "	<u>198*LBL 52</u>
93 6 E3	146 E2	199 RCL 04
94 /	147 RCL 26	200 FC? 03
95 STO 23	148 INT	201 FS? 04
96 FC?C 05	149 ARCL X	202 GTO 18
97 GTO 05	150 X<Y?	203 GTO 07
98 RCL 08	151 "'%"	<u>204*LBL 72</u>
99 -45 E-4	152 FIX 2	205 2
100 *	153 AVIEW	206 FS? 04
101 RCL 24	154 RTN	207 GTO 18
102 +	<u>155*LBL 08</u>	208 GTO 07
103 20	156 AVIEW	<u>209*LBL 54</u>
104 +	157 XEQ 09	210 3
105 6	158 GTO 05	211 FS? 01
106 MOD	<u>159*LBL 06</u>	212 GTO 18
107 GTO 08	160 SF 25	213 GTO 07

<u>214*LBL 74</u>	265 FC? 05	316 CLA
215 4	266 FS? 46	317 ARCL 21
216 FC? 01	267 GTO 16	318 ARCL 22
217 FS? 02	<u>268*LBL 12</u>	319 20
218 GTO 18	269 FS? 00	<u>320*LBL 20</u>
<u>219*LBL 07</u>	270 GTO 17	321 DSE X
220 X<> 24	271 RCL 00	322 GTO 20
221 X#0?	272 R-D	323 AVIEW
222 CF IND X	273 FRC	324 RTN
223 SF IND 24	274 STO 00	<u>325*LBL 09</u>
224 RCL IND 24	275 RCL IND 06	326 8
225 RCL IND Y	276 *	327 STO 05
226 X<Y?	277 E3	328 STO 09
227 SF 08	278 *	<u>329*LBL 10</u>
228 SF 09	279 RCL 06	330 RCL 00
229 FS?C 00	280 SIGN	331 R-D
230 GTO 08	281 *	332 FRC
231 /	282 ST+ 08	333 STO 00
232 ST* 08	<u>283*LBL 07</u>	334 RCL 09
233 GTO 07	284 FS? 07	335 *
<u>234*LBL 23</u>	285 RTN	336 11
235 FS? 07	286 SF 09	337 ST+ Y
236 GTO 16	<u>287*LBL 08</u>	338 DSE 05
237 3	288 RCL 01	339 ""
238 GTO 11	289 RCL IND 24	340 RCL 05
239*LBL 33	290 /	341 +
240 2	291 6 E2	342 RCL IND X
241 GTO 11	292 *	343 X<> IND Z
<u>242*LBL 42</u>	293 RCL 08	344 STO IND Y
243 SIGN	294 X<Y?	345 DSE 09
<u>244*LBL 11</u>	295 GTO 17	346 GTO 10
245 CHS	296 6 E3	347 10.018
246 STO 06	297 X<>Y	348 STO 09
247 33	298 X>Y?	349 RTN
248 FS? 07	299 GTO 18	<u>350*LBL 16</u>
249 ST+ 05	300 ASTO 21	351 "CRASH"
250 SF 08	301 ASHF	352 GTO 07
251 GTO 12	302 ASTO 22	<u>353*LBL 17</u>
<u>252*LBL 25</u>	303 "("	354 "STALLED "
253 3	304 ARCL X	355 ARCL 08
254 GTO 11	305 ">") ("	356 GTO 07
<u>255*LBL 35</u>	306 E3	<u>357*LBL 18</u>
256 2	307 /	358 "REDLINE "
257 GTO 11	308 RCL IND 24	359 ARCL 08
<u>258*LBL 44</u>	309 1/X	<u>360*LBL 07</u>
259 SIGN	310 21	361 AVIEW
<u>260*LBL 11</u>	311 *	362 TONE 8
261 STO 06	312 *	363 TONE 8
262 ENTER^	313 ARCL X	364 SF 00
263 SIGN	314 ">")"	365 END
264 X=Y?	315 AVIEW	

Truck Routes

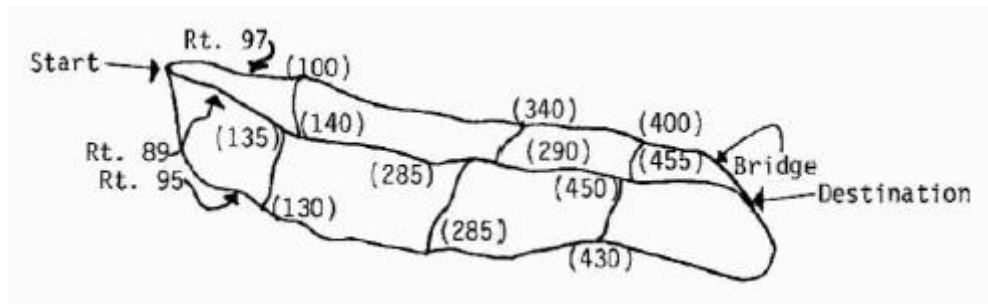
Kenneth Sharp - Games Solutions Book

Breaker – Breaker 19 --- All you would-be eighteen wheelers!!!

You must get your shipment delivered within ten hours or be charged a penalty. Smokies are patrolling the roads and obstructing on the road slows you down. Listen closely to your CB for messages. Any of three roads (95, 89, or 97) will get you to your destination, but you may need to change routes several times during your run. Route changes are permitted only at designated exits. Smokies will fine you if they catch you speeding, wreckers will charge you for repairs if they must pull you from a crash. Get into your rig and put the pedal to the metal. See you on the flip-flop.

Sample Problem.

You must drive from 'Start' to 'Destination' in less than 10 hours. You can change roads only at exits. A perfect run will net you \$100.00



Route 95 is 725 miles long. It is heavily patrolled by "smokies"

Route 89 is 650 miles long. It is a mountain road plagued by rock slides.

Route 97 is 500 miles long. It has a curve that gets slick when wet, rock lides and a bridge that perpetually falls down.

Display	Input	Function	Comments
	[XEQ]	"SIZE" 01	
	[XEQ]	"TRUCK"	
SEED?	.987654321	[R/S]	Enter seed for random number
ROUTE?	89	[R/S]	Let's try route 89
SPEED?	55	[R/S]	Let's start at 55 MPH
ROUTE NO. 89			
SPEED = 55MPH			
MI. POST 0			
CH. SPEED/RT?	[N]	[R/S]	No need to make changes yet
BREAKER 19			
BRIDGE OUT			

```

M.P. 490 ON 97
CLOSED 4 HR
ROUTE NO. 89
SPEED = 55MPH
MI. POST 55
CH. SPEED/RT.7      [N]   [R/S]      (one hour has passed)
BREAKER 19          doesn't affect us – no changes
WET ROAD AT
M.P. 165 ON 97
SPEED LMT. 35
ROUTE 89
SPEED = 55MPH
MI. POST 110
CH. SPEED/RT.7      [N]   [R/S]      Route 89 is still clear
BREAKER 19
SMOKEY AT
M.P. 575 ON 95
ROUTE NO. 89
MI. POST 275
CH. SPEED/RT.7      [R/S]      Let's try another road (R/S = yes)
SPEED--R/RT.--J
DRIVING TIME        ,1622 [R/S]      16 min. 22 sec. should put us at
                                     the destination
etc...

```

Status: SIZE 019, FIX 0, USER Mode ON, Total 1,640 program bytes.

Data Registers:

00-03	Time Smokey will remain	(Route #95)
04-06	Time to clear slide	(Route #89)
07	Time for road to dry	
08	Time to clear Slide	(Route #97)
09	Time to repair bridge	
10	Random number	
11	Route	
12	Speed	
13	Previous position	
14	Present Position	
15	Money	
16	Present position or HMS conversion	
17	Trip time	
18	Driving time	

Flags Used:

00-03	Set: Smokey Present	Clear: Road Clear	(Route #95)
04-06	Set: Rock Slide	Clear: Road Clear	(Route #89)

07	Set: Wet Road	Clear: Road Clear	
08	Set: Rock Slide	Clear: Road Clear	(Route #97)
09	Set: Bridge Out	Clear: Road Clear	
27	Set: USER mode	Clear: USER Mode off	
29	Set: Digit Grouping	Clear: No digit grouping	

Program listing:

01*LBL "TRUCK"	41 PROMPT	82 RCL 12	123 X<=0?
02 SF 27	42 X=0?	83 X<=0?	124 RTN
03 FIX 0	43 GTO A	84 XEQ 10	125 "EEE...."
04 CF 29	44 ABS	85 "BREAKER 19"	126 AVIEW
05 CF 21	45 STO 12	86 AVIEW	127 TONE 8
06 FS? 55	46 85	87 BEEP	128 TONE 9
07 SF 21	47 -	88 XEQ 62	129 TONE 8
08*LBL a	48 X>0?	89 SF IND X	130 TONE 9
09 0	49 GTO A	90 XEQ IND X	131 TONE 8
10 X<>F	50*LBL 10	91 RTN	132 TONE 9
11 CF 07	51 SF 28	92*LBL 62	133 "SMOKEY"
12 CF 08	52 "ROUTE NO. "	93 RNG	134 AVIEW
13 CF 09	53 ARCL 11	94 STO 10	135 PSE
14 E-2	54 AVIEW	95 E1	136 2
15 CLRGX	55 PSE	96 *	137 *
16 0	56 "SPEED="	97 INT	138 ST- 15
17 STO 14	57 ARCL 12	98 RTN	139 "FINE OF \$"
18 RNG	58 >" MPH"	99*LBL 95	140 ARCL X
19 STO 10	59 AVIEW	100 14	141 AVIEW
20 E2	60 PSE	101 STO 18	142 PSE
21 STO 15	61 "MI. POST "	102 135	143 .25
22 E1	62 ARCL 14	103 STO 16	144 ST- 17
23 STO 17	63 AVIEW	104 FS? 00	145 RCL 16
24*LBL 12	64 PSE	105 XEQ 19	146 STO 14
25 "RT.?"	65 "CHANGES?"	106 290	147 GTO 10
89/95/97"	66 PROMPT	107 STO 16	148*LBL 22
26 PROMPT	67 X=Y?	108 FS? 01	149 RCL 11
27 STO 11	68 GTO 13	109 XEQ 19	150 95
28 97	69 "SP.-A/RT.-B"	110 445	151 X=Y?
29 X=Y?	70 PROMPT	111 STO 16	152 GTO 52
30 GTO A	71*LBL 13	112 FS? 02	153 RCL 11
31 CLX	72 XEQ 60	113 XEQ 19	154 89
32 95	73 E	114 575	155 X=Y?
33 X=Y?	74 ST- 17	115 STO 16	156 GTO 53
34 GTO A	75 RCL 14	116 FS? 03	157 RCL 14
35 CLX	76 STO 13	117 XEQ 19	158 500
36 89	77 RCL 12	118 RTN	159 STO 18
37 X#Y?	78 ST+ 14	119*LBL 14	160 -
38 GTO 12	79 XEQ IND 11	120 RCL 12	161 X<0?
39*LBL A	80 GTO 22	121 55	162 GTO 51
40 "SPEED=?"	81*LBL 60	122 -	163*LBL 54

164 RCL 12	217 RTN	268 FS? 07	320*LBL 01
165 /	218 CF IND X	269 XEQ 19	321 3
166 ST+ 17	219 RTN	270 16	322 ST+ 01
167 RCL 18	220*LBL 89	271 STO 18	323*LBL 24
168 STO 14	221 16	272 405	324 XEQ 33
169 RCL 17	222 STO 18	273 STO 16	325 "M.P. 290 ON
170 X<0?	223 145	274 FS? 08	95"
171 GTO 55	224 STO 16	275 XEQ 19	326 AVIEW
172 "ON TIME"	225 FS? 04	276 20	327 PSE
173 AVIEW	226 XEQ 19	277 STO 18	328 RTN
174 PSE	227 300	278 490	329*LBL 02
175*LBL 56	228 STO 16	279 STO 16	330 3
176 "MONEY=\$"	229 FS? 05	280 FS? 09	331 ST+ 02
177 ARCL 15	230 XEQ 19	281 XEQ 19	332*LBL 25
178 AVIEW	231 465	282 RTN	333 XEQ 33
179 0	232 STO 16	283*LBL 19	334 "M.P. 445 ON
180 STO 12	233 FS? 06	284 RCL 16	95"
181 STOP	234 XEQ 19	285 RCL 13	335 AVIEW
182 GTO D	235 RTN	286 -	336 PSE
183*LBL 55	236*LBL 16	287 X<=0?	337 RTN
184 "LATE"	237 "*ROCKS*"	288 RTN	338*LBL 03
185 AVIEW	238 AVIEW	289 RCL 14	339 3
186 PSE	239 PSE	290 RCL 16	340 ST+ 03
187 25	240*LBL 11	291 -	341*LBL 26
188 *	241	292 X<=0?	342 XEQ 33
189 ST+ 15	***CRASH***	293 RTN	343 "M.P. 575 ON
190 GTO 56	242*LBL 21	294 GTO IND 18	95"
191*LBL 52	243 AVIEW	295*LBL 20	344 AVIEW
192 RCL 14	244 PSE	296 "SPLASH"	345 PSE
193 725	245 "CALL	297 AVIEW	346 RTN
194 STO 18	WRECKER"	298 PSE	347*LBL 33
195 -	246 AVIEW	299 "WET FEET"	348 "SMOKEY AT"
196 X<0?	247 PSE	300 GTO 21	349 AVIEW
197 GTO 51	248 RCL 16	301*LBL 18	350 PSE
198 GTO 54	249 STO 14	302 RCL 12	351 RTN
199*LBL 53	250 XEQ 62	303 35	352*LBL 04
200 RCL 14	251 8	304 -	353 3
201 550	252 *	305 X<=0?	354 ST+ 04
202 STO 18	253 5	306 RTN	355 RDN
203 -	254 +	307 "-+SKID+--"	356*LBL 27
204 X>0?	255 ST- 15	308 AVIEW	357 XEQ 17
205 GTO 54	256 "REPAIRS-\$"	309 PSE	358 "M.P. 145 ON
206*LBL 51	257 ARCL X	310 GTO 11	89"
207 .009	258 AVIEW	311*LBL 00	359 AVIEW
208*LBL 57	259 PSE	312 3	360 PSE
209 XEQ 41	260 2.5	313 ST+ 00	361 GTO 15
210 ISG X	261 ST- 17	314*LBL 23	362*LBL 05
211 GTO 57	262 GTO 10	315 XEQ 33	363 3
212 GTO 10	263*LBL 97	316 "M.P. 135 ON	364 ST+ 05
213*LBL 41	264 18	95"	365 RDN
214 FC? IND X	265 STO 18	317 AVIEW	366*LBL 28
215 RTN	266 165	318 PSE	367 XEQ 17
216 DSE IND X	267 STO 16	319 RTN	

368 "M.P. 300 ON 89"	415*LBL 32	465 XEQ 63	518 RCL 14
369 AVIEW	416 "BRIDGE OUT"	466 RCL 11	519 -
370 PSE	417 AVIEW	467 95	520 RND
371 GTO 15	418 PSE	468 X=Y?	521 X=0?
372*LBL 06	419 "M.P. 490 ON 97"	469 GTO 50	522 GTO IND Y
373 4	420 AVIEW	470 RCL 11	523 RTN
374 ST+ 06	421 PSE	471 89	524*LBL 50
375 RDN	422*LBL 15	472 X=Y?	525 40
376*LBL 29	423 "CLOSED "	473 GTO 34	526 ENTER^
377 XEQ 17	424 ARCL IND X	474 35	527 130
378 "M.P. 465 ON 89"	425 ">" HR."	475 ENTER^	528 XEQ 38
379 AVIEW	426 AVIEW	476 E2	529 41
380 PSE	427 PSE	477 XEQ 38	530 ENTER^
381 GTO 15	428 RTN	478 36	531 285
382*LBL 17	429*LBL B	479 ENTER^	532 XEQ 38
383 "ROCK SLIDE"	430 RCL 12	480 340	533 42
384 AVIEW	431 X#0?	481 XEQ 38	534 ENTER^
385 PSE	432 GTO 58	482 37	535 430
386 RTN	433 TONE 0	483 ENTER^	536 XEQ 38
387*LBL 07	434 "NOT MOVING"	484 400	537 GTO 59
388 3	435 AVIEW	485 XEQ 38	538*LBL 34
389 ST+ 07	436 PSE	486*LBL 59	539 43
390*LBL 30	437 GTO 10	487 "NO EXIT"	540 ENTER^
391 "WET ROAD AT"	438*LBL 58	488 AVIEW	541 135
392 AVIEW	439 RCL 14	489 PSE	542 XEQ 38
393 PSE	440 STO 16	490 GTO 22	543 44
394 "M.P. 165 ON 97"	441 "DRIVING TIME"	491*LBL 63	544 ENTER^
395 AVIEW	442 PROMPT	492 X<> 13	545 285
396 PSE	443 HR	493 X<> 14	546 XEQ 38
397 "SPEED LMT. 35"	444 STO 18	494 X<> 13	547 45
398 AVIEW	445 INT	495 RTN	548 ENTER^
399 PSE	446 X#0?	496*LBL 35	549 450
400 RTN	447 GTO B	497 89	550 XEQ 38
401*LBL 08	448 XEQ 60	498 STO 11	551 46
402 4	449 RCL 18	499 140	552 ENTER^
403 ST+ 08	450 ABS	500 STO 14	553 140
404 RDN	451 ST- 17	501 GTO 39	554 XEQ 38
405*LBL 31	452 RCL 16	502*LBL 36	555 47
406 XEQ 17	453 STO 13	503 89	556 ENTER^
407 "M.P. 405 ON 97"	454 RCL 12	504 STO 11	557 290
408 AVIEW	455 RCL 18	505 290	558 XEQ 38
409 PSE	456 *	506 STO 14	559 48
410 GTO 15	457 ST+ 14	507 GTO 39	560 ENTER^
411*LBL 09	458 RCL 18	508*LBL 37	561 455
412 4	459 X<0?	509 89	562 XEQ 38
413 ST+ 09	460 CF 28	510 STO 11	563 GTO 59
414 RDN	461 X<0?	511 455	564*LBL 40
	462 XEQ 63	512 STO 14	565 89
	463 XEQ IND 11	513*LBL 39	566 STO 11
	464 FC? 28	514 .25	567 135
		515 ST- 17	568 STO 14
		516 GTO 10	569 GTO 39
		517*LBL 38	570*LBL 41

571 89	596 STO 11	621 X<0?	646 INT
572 STO 11	597 430	622 >" OVER"	647 ARCL X
573 285	598 STO 14	623 X>0?	648 >"SC."
574 STO 14	599 GTO 39	624 >" TO GO"	649 AVIEW
575 GTO 39	600*LBL 46	625 AVIEW	650 STOP
576*LBL 42	601 97	626 PSE	651 GTO 10
577 89	602 STO 11	627 HMS	652*LBL C
578 STO 11	603 E2	628 ABS	653 "STATUS:"
579 450	604 STO 14	629 STO 16	654 AVIEW
580 STO 14	605 GTO 39	630 INT	655 23.032
581 GTO 39	606*LBL 47	631 CLA	656 ENTER^
582*LBL 43	607 97	632 ARCL X	657 .01
583 95	608 STO 11	633 >"HR."	658*LBL 49
584 STO 11	609 340	634 RCL 16	659 FS? IND X
585 130	610 STO 14	635 FRC	660 TONE IND X
586 STO 14	611 GTO 39	636 E2	661 FS? IND X
587 GTO 39	612*LBL 48	637 *	662 XEQ IND Y
588*LBL 44	613 97	638 STO 16	663 ISG X
589 95	614 STO 11	639 INT	664 ISG Y
590 STO 11	615 400	640 ARCL X	665 GTO 49
591 285	616 STO 14	641 ""MN."	666 GTO 10
592 STO 14	617 GTO 39	642 RCL 16	667 END
593 GTO 39	618*LBL D	643 FRC	
594*LBL 45	619 "TIME:"	644 E2	
595 95	620 RCL 17	645 *	

Pilot (Flying)

Whodunit –Swap Disks

Get what you can from the program listing, no documentation exists!

Program listing:

1. LBL "PILOTE"	44. P-R	87. LBL 11
2. LBL A	45. ST+ 03	88. ST+ 03
3. CF 21	46. "DIS.="	89. "DIS.="
4. FIX 00	47. ARCL 03	90. ARCL 03
5. CF 29	48. >" KM"	91. AVIEW
6. CLX	49. AVIEW	92. DSE 06
7. STO 03	50. PSE	93. GTO 11
8. STO 04	51. X<>Y	94. RTN
9. "2000 ...KM"	52. 1 E3	95. LBL 05
10. AVIEW	53. *	96. 1
11. PSE	54. ST+ 04	97. RCL 01
12. "PILOTE AUTO."	55. "HAU.="	98. ABS
13. AVIEW	56. ARCL 04	99. X>Y?
14. PSE	57. >" M"	100. GTO 09
15. "HAUTEUR ?(M)"	58. AVIEW	101. RCL 03
16. PROMPT	59. RCL 04	102. 2000
17. STO 05	60. RCL 05	103. -
18. "VIT. ?(KM/H)"	61. X<=Y?	104. ABS
19. PROMPT	62. XEQ 01	105. 5
20. 60	63. PSE	106. X<=Y?
21. /	64. RCL 04	107. GTO 10
22. STO 02	65. X<=0?	108. "POSE.."
23. "DIST. ?(KM)"	66. GTO 05	109. AVIEW
24. PROMPT	67. DSE 00	110. PSE
25. RCL 02	68. GTO 00	111. "BRAVO ..."
26. /	69. RCL 03	112. PROMPT
27. 10	70. 2000	113. GTO A
28. /	71. X<=Y?	114. LBL 09
29. INT	72. GTO 12	115. "CRASH ..."
30. STO 06	73. GTO 03	116. PROMPT
31. LBL 03	74. LBL 01	117. GTO 09
32. "MIN<)"	75. RCL 03	118. LBL 10
33. PROMPT	76. 2000	119. "AUX VACHES..."
34. STO 01	77. X<=Y?	120. PROMPT
35. X<>Y	78. RTN	121. GTO 10
36. INT	79. RCL 05	122. LBL 12
37. STO 00	80. STO 04	123. "PLUS DE JUS,"
38. LBL 00	81. CLX	124. AVIEW
39. RCL 01	82. STO 01	125. PSE
40. RCL 02	83. STO 00	126. GTO 09
41. RCL 01	84. RCL 02	127. END
42. %	85. 10	
43. -	86. *	

The Eighth Passenger(HP-67)

By 'Archilog' <https://www.hpmuseum.org/forum/thread-12909.html>

From some memories of a French friend about a game for the TI-58, I wrote an ALIEN program for the HP-67/97. The point here is that the keyboard, like for the 41 (with armed flag 22), can receive direct key inputs - I don't know if it is possible on a 42 to use the keyboard like a pad.

The goal is to escape (in a limited time) from the alien on a 8x8 square which represents the Nostromo whose walls are infected with toxic substances - yeah, the beast left its progeny behind, what did you expect? You have to reach point 0.0 (the airlock) to win the game before the vessel explodes.

What is amazing:

- you know the distance from/to the evil, but NOT WHERE it is located;
- The beast is always moving in YOUR direction.

You can find this game there: <http://www.silicium.org/forum/viewtopic...28#p519028>

I shall post it here in the forum in a few weeks. Enjoy.

PS: oh yes, it's a big program; not sure you would accept to type it in every time you want to play on the 42, but it should be okay on the awesome Free42/DM42.

This thread, as sticky as the saliva of the most repulsive monster in the universe, marks the return of....ALIEN, the eighth passenger! I don't know about you, but with me, it always has a little effect...

Because finally, we thought we had said everything about this abject creature, but a series of surprising clues rekindled the debate on his presence at the in this forum itself! what am I saying, in this forum? maybe even everything near you, in a drawer of your desk, in your pocket, on your bedside table!!!!

Have you noticed that the handheld computer whose the programming has allowed this fateful regeneration is coloured from disturbing greenish shades to... his keys? How to explain this color while the entire HP calculator collection oscillates between austere and classic grey and black, and orange and blue yellows and blues more dashing? (You'll tell me, and the HP-27, huh, huh, and I'll answer you, yes, yes, yes, - I have arguments too - but the HP-27 ALSO is a creature of the alien! And tac!

But there is something more worrying: the initials H.P. themselves, which we believe have always been those of Mr. Hewlett and Mr. Packard, referring to the bonhomie of two good guys with reassuring boy scout reels Californians who have dedicated their lives to their business, their wives and their families, would in fact be those of the.... Eighth Passenger! On this point, Mr. Capelo is clear: "I'm positive."



Photograph of a Calculator taken right before User has been devoured by the Ugliest Monster of the Universe - Courtesy of Internet/=

Back to 2122, exactly 60 seconds before the final destruction of the Nostromo ship.... And if you don't believe me, here's the soundtrack.

You are the only survivor of the crew and have just triggered the self-destruct process of the Nostromo, fleeing the hideous xenomorphic that you hope to get rid of forever. To do this, you have an HP-67 that has undergone all required decontamination tests, and numeric keys on the keyboard to guide you in the maze of the ship. Remember: you can move from one place to another square in all directions of this 8x8 checkerboard, but cannot touch the alienated walls without a certain death... and of course, your predator in the unknown position is heading towards you without error: Alien you feel, Alien follows you, Alien is here! Find the exit (0.0) as soon as possible, the countdown is on!

A game can be played alone or with several players, in a tournament. Here is a typical handle:-

[7][7][7] : a number between 1 and 999 to place the actors according to the generation of the pseudo-random number; the most interesting parts take place with a seed towards the middle of the range.

[A] : starts the program;

4.4 : display of your position;
 4,243: distance from the ALIEN.
 4.243_____59 : distance from the ALIEN and time remaining before destruction of the Nostromo. The monster is obviously between you and the exit!

You can use the keyboard and its keys directly as a directional keypad to move you, without pressing anything else, thanks to a trick that 'badaze' allowed me to discover.

[7] [8] [9]
 [4] [5] [6]
 [1] [2] [3]

[7] : you choose to move sideways (-1, +1);

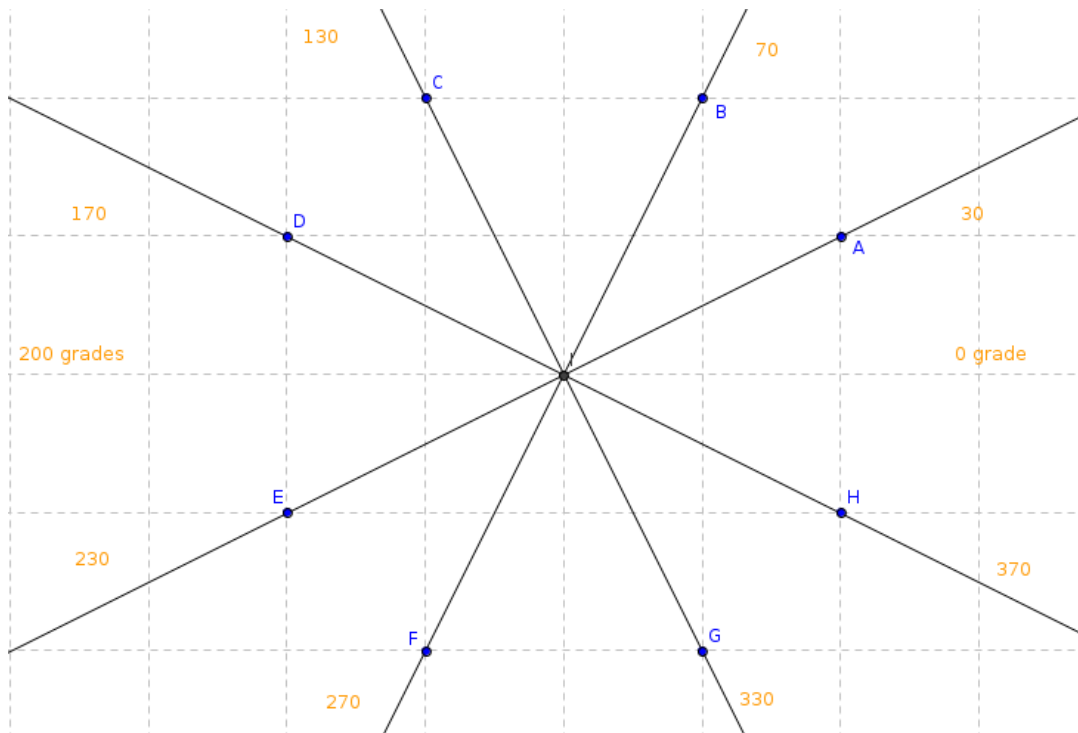
3.5: your new position;

4,472: the distance between you and the filthy;
(but the alien's response is not long in coming...)

3,606: the xenomorph has moved, and is rushing towards you!

3.606_____58: Displaying your time... what to do? Don't hang around here!

The Alien is always moving towards you following this angular pattern. The angles measured in grades are used to save a few bytes:



There is a solution, if not more... For my part, with this deal I got out of there in less than thirty seconds. The proof:



The original program was prodigiously shortened by the subtle pir2 routine that allows the detection of the key pressed in a record number of steps. The space freed up allows to include the countdown that will be modified in step 3, if necessary, for a maximum limit of 99 seconds.

The draw of the pseudo-random number has also been modified and has no longer been used, a lot of randomness! Inspired by a "jxano" algorithm, it allows challenges much more interesting, especially if... your speed depends on the outcome of the game.

Of course, the HP-67 computer does not have an internal stopwatch, so it was simulated using the display of the exponents of 10 on the right of the screen; and no more than in fiction, the elapsed time is real. But the objective is achieved and you too will lose your socks!

This program will probably benefit from some final modifications: it should be easy to homogenize the presentation of the distance for example. The following version at least has the merit to work.

```

Code:
001 LBL A      011 CF 3      021 3      031 RND      041 GTO 9      051 +      061 STO-3      071 1      081 7      091 LBL 8
002 STO 5      012 GSB E      022 /      032 1      042 LBL 8      052 1      062 RCL 4      072 3      082 GSB 4      092 GSB 2
003 6          013 1          023 ENTER      033 -          043 GSB 1      053 0      063 2      073 GSB 3      083 STO-2      093 GTO 9
004 0          014 -          024 INT      034 STO+0      044 GSB 2      054 /          064 3      074 2      084 RCL 4      094 LBL 3
005 STO I      015 X<0?      025 1      035 X<>Y      045 X<>Y      055 STO 4      065 GSB 3      075 7      085 3      095 X<=Y?
006 GRAD      016 GTO 9      026 -          036 STO+1      046 4          056 3          066 3          076 GSB 4      086 3          096 SF 0
007 GSB 0      017 9          027 X<>Y      037 GTO 8      047 0          057 GSB 3      067 7          077 STO+2      087 X>Y?      097 RCL 4
008 GSB 1      018 X<=Y?      028 FRAC      038 LBL 9      048 0          058 1          068 GSB 4      078 RCL 4      088 GTO 8      098 RTN
009 GSB 2      019 GTO 9      029 3          039 DSP 0      049 X<>Y      059 7          069 STO+3      079 0          089 1          099 LBL 4
010 LBL 9      020 X<>Y      030 *          040 5          050 X<0?      060 GSB 4      070 RCL 4      080 GSB 3      090 STO-2      100 X>Y?

101 SF 1      111 FS? 1      121 FIX      131 1      141 LBL 2      151 DSP 3      161 *          171 PI          181 FRAC      191 -
102 0          112 GTO 5      122 DSP 1      132 0          142 RCL 3      152 FIX      162 GSB 6      172 X^2      182 8          192 1/X
103 GSB 5      113 CF 0      123 8          133 /          143 RCL 1      153 PAUSE      163 STO 3      173 e^x      183 *          193 LBL E
104 RTN      114 RTN      124 RCL 0      134 +          144 -          154 RTN      164 RCL 5      174 *          184 INT      194 SCI
105 LBL 5      115 LBL 5      125 GSB 7      135 PAUSE      145 RCL 2      155 LBL 0      165 PI          175 GSB 6      185 RTN      195 DSP 3
106 FS? 0      116 CF 0      126 8          136 X#0?      146 RCL 0      156 RCL 5      166 e^x      176 STO 1      186 LBL 7      196 FS? 3
107 GTO 5      117 CF 1      127 RCL 1      137 RTN      147 -          157 GSB 6      167 *          177 RTN      187 X<0?      197 RTN
108 CF 1      118 1          128 GSB 7      138 FIX      148 ->P      158 STO 2      168 GSB 6      178 LBL 6      188 SQR x      198 DSZ I
109 RTN      119 RTN      129 X<>Y      139 - x -          149 X=0?      159 RCL 5      169 STO 0      179 8          189 X#Y?      199 GTO 0
110 LBL 5      120 LBL 1      130 Rd      140 R/S      150 1/X      160 PI          170 RCL 5      180 /          190 RTN      200 RCL I

201 1/x      211 GTO E
202 LBL 0
203 RCL I
204 10^x
205 *
206 PAUSE
207 FS? 3
208 RTN
209 LAST x
210 /
-----
| Reg. I : temps (s.) | LBL 0 : Routine d'affectation positions J & A, entre autres
| Reg. 0 : Xj         | LBL 1 : Routine de visualisation position J
| Reg. 1 : Yj         | LBL 2 : Routine de visualisation distance AJ
| Reg. 2 : Xa         | LBL 3 & 4 : Routines d'économie (voir LBL 8)
| Reg. 3 : Ya         | LBL 5 : Routines de déplacement A
| Reg. 4 : angle      | LBL 6 : Routine d'économie (voir LBL 0)
| Reg. 5 : graine     | LBL 7 : Routine d'économie (voir LBL 1)
|                     | LBL 8 : Routine de détermination angulaire
|                     | LBL 9 : Routine d'interprétation clavier
|                     | LBL E : Routine d'interfaçage clavier
|-----

```

6:49PM 05/18

Galaxis Game.

By 'Heimchr' <https://www.hpmuseum.org/forum/thread-6063-post-54168.html>

This HP-41 program allows you to play this game.

The goal of the game is to find 4 space ships in as few attempts as possible. The game board is a grid of 9 columns by 7 rows, i.e. coordinates A1 through G9. If there is no ship on the coordinates you entered, you'll get as answer the number of ships visible from that location along the lines of the grid (horizontal, vertical, diagonal). You cannot look 'behind' a ship. For example, if the ships are on locations G1, F2, A5 and D7, and you shoot at D4, the answer will be 2 - not 3 because from D4 you can see the ship at F2 but not the ship behind it at G1, so only the ships at F2 and D7 are visible.

I wrote this program in the mid-80's when I first got my 41CV. Looking at it now, I see lots of opportunities to make it faster and shorter so maybe I'll post an update later.

To start, simply XEQ "GALAXIS". You'll be asked whether you want sound or not, and then you can start guessing. Enter coordinates as a letter between A and G followed by a number 1 through 9 (since you're in Alpha mode you'll need to use the shift key to enter the number). The number of ships visible from your entered coordinates will be shown through the indicators for flags 0 through 4. When you've hit all 4 ships, you'll see the number of tries it took you and you'll be asked if you want to play again.

A blank game grid is attached - you can print it to play the game.

Let me know if you have any questions/issues!

Requirements:

27 data registers

885 program bytes (127 registers)

HP-41C with 2 memory modules or quad-memory module and X-functions module, or
HP-41CV with X-functions module; or HP-41CX

Registers:

R00: seed for random number generator

R01: work variable for counter 2.005 -> 5.005

R02: coordinates of ship 1

R03: coordinates of ship 2

R04: coordinates of ship 3

R05: coordinates of ship 4

R06: not used

R07: Entered coordinates

R08: Entered coordinates

R09: Entered coordinates

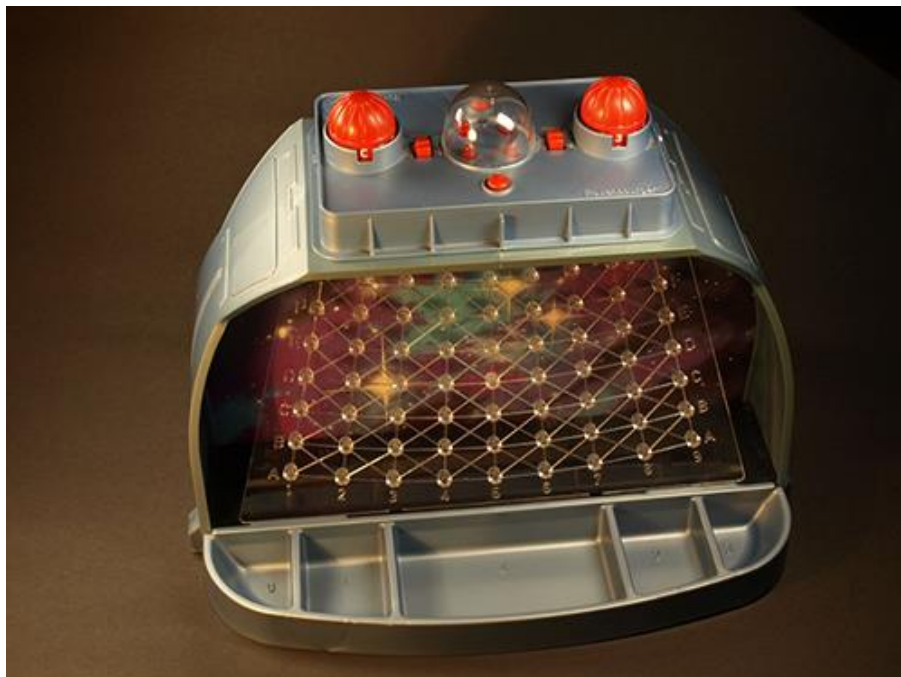
R10: ASCII value of entered X-coordinate

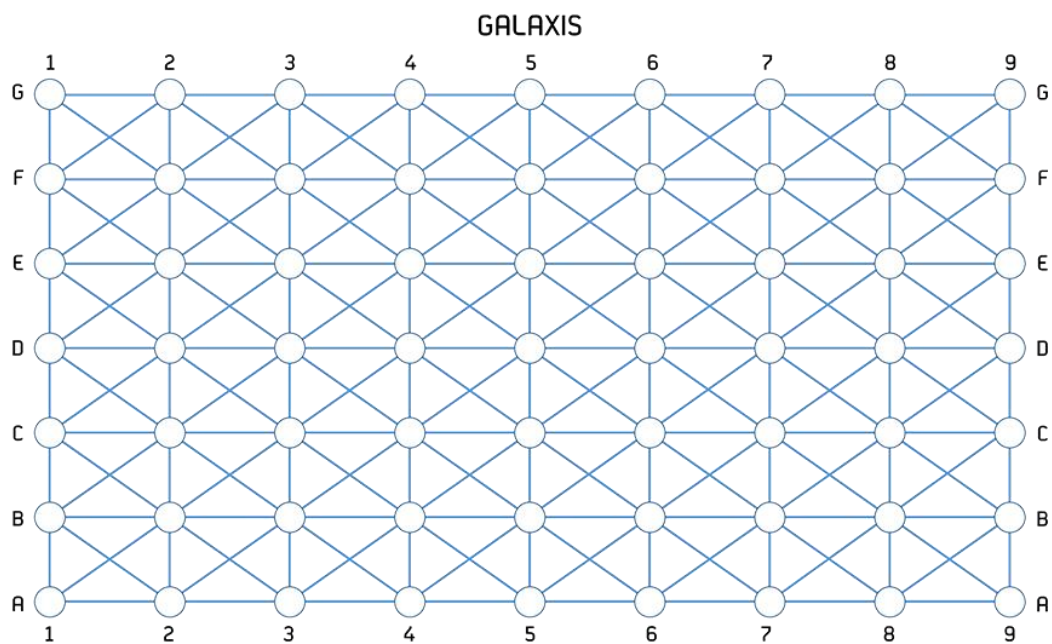
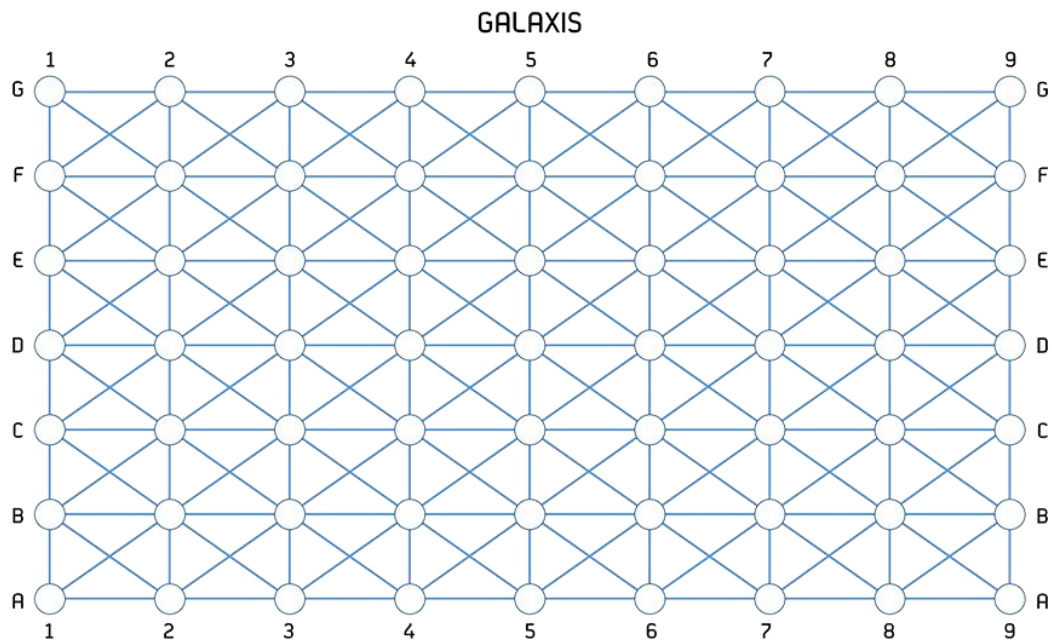
R11: entered Y-coordinate

R12: work variable used during check how many ships are visible
R13: work variable used during check how many ships are visible (coordinates of position currently being checked)
R14: work variable used during check how many ships are visible
R15: number of visible ships from entered coordinates
R16: number of ships already hit (during the game); Entered answer to the question "AGAIN? Y/N" (after the game)
R17: number of tries (during game); "Y" (after the game)
R18: not used
R19: work counter to determine if computer generated ship locations overlap
R20: work counter to determine if computer generated ship locations overlap
R21: 0 if ship 1 has not been hit, 1 if it has been hit
R22: 0 if ship 2 has not been hit, 1 if it has been hit
R23: 0 if ship 3 has not been hit, 1 if it has been hit
R24: 0 if ship 4 has not been hit, 1 if it has been hit
R25: work pointer to R21 through R24
R26: status of all flags 0-55 at start of game

Flags:

00: set if 0 ships visible from entered coordinates
01: set if 1 ship visible from entered coordinates
02: set if 2 ships visible from entered coordinates
03: set if 3 ships visible from entered coordinates
04: set if 4 ships visible from entered coordinates
26: turned on if playing with sound, off if playing without sound
29: turned off at the beginning of the game to suppress digit grouping





Program listing:

01*LBL "GALAXIS"	11 SF 26	21 STO 24	31 CLA
02 RCLFLAG	12 X=Y?	22*LBL 40	32 2.005
03 STO 26	13 CF 26	23 .9	33 STO 01
04 "SOUND? Y/N"	14 RCL 00	24 STO 15	34*LBL 01
05 AON	15 ABS	25 STO 17	35 XEQ 00
06 PROMPT	16 STO 00	26 .003	36 7
07 AOFF	17 0	27 STO 16	37 *
08 ASTO X	18 STO 21	28 FIX 0	38 INT
09 "N"	19 STO 22	29 CF 29	39 65
10 ASTO Y	20 STO 23	30 X<>F	40 +

41 XTOA	94 RCL IND 01	147 49	200 CLA
42 XEQ 00	95 X=Y?	148 X<>Y	201 XTOA
43 8	96 GTO 39	149 X<Y?	202 ARCL 12
44 *	97 ISG 01	150 GTO 45	203 ASTO 13
45 INT	98 GTO 37	151 57	204*LBL 06
46 E	99 GTO 33	152 X<Y?	205 RCL 13
47 +	100*LBL 39	153 GTO 45	206 RCL IND 01
48 ARCL X	101 19	154 ATOX	207 X=Y?
49 ASTO IND 01	102 RCL 01	155 X#0?	208 GTO 25
50 CLA	103 +	156 GTO 45	209 ISG 01
51 ISG 01	104 STO 25	157 ISG 17	210 GTO 06
52 GTO 01	105 RCL IND X	158 CLA	211 GTO 05
53 GTO 43	106 X>0?	159 ARCL 08	212*LBL 07
54*LBL 00	107 GTO 47	160 ATOX	213 RCL 10
55 RCL 00	108*LBL 46	161 STO 10	214 STO 12
56 RN#	109 E	162 ANUM	215*LBL 09
57 STO 00	110 ST+ IND 25	163 STO 11	216 2.005
58 RTN	111 ISG 17	164 STO 12	217 STO 01
59*LBL 43	112 ISG 16	165*LBL 35	218 E
60 2.004	113 GTO 38	166 2.005	219 ST+ 12
61 STO 19	114 GTO 34	167 STO 01	220 RCL 12
62 STO 20	115*LBL 45	168 E	221 72
63*LBL 41	116 TONE 0	169 ST- 12	222 X=Y?
64 RCL 19	117 "INVALID"	170 RCL 12	223 GTO 10
65 1.001	118 AVIEW	171 0	224 RCL 12
66 +	119 PSE	172 X=Y?	225 CLA
67 STO 20	120 GTO 02	173 GTO 04	226 XTOA
68*LBL 42	121*LBL 47	174 RCL 10	227 ARCL 11
69 RCL IND 19	122 TONE 0	175 CLA	228 ASTO 13
70 RCL IND 20	123 "ALREADY HIT"	176 XTOA	229*LBL 08
71 X=Y?	124 AVIEW	177 ARCL 12	230 RCL 13
72 GTO 40	125 PSE	178 ASTO 13	231 RCL IND 01
73 ISG 20	126 GTO 02	179*LBL 03	232 X=Y?
74 GTO 42	127*LBL 38	180 RCL 13	233 GTO 26
75 ISG 19	128 TONE C	181 RCL IND 01	234 ISG 01
76 GTO 41	129 TONE C	182 X=Y?	235 GTO 08
77*LBL 02	130 TONE C	183 GTO 24	236 GTO 09
78 TONE 9	131 "HIT"	184 ISG 01	237*LBL 10
79 "GUESS?"	132 AVIEW	185 GTO 03	238 RCL 10
80 .9	133 PSE	186 GTO 35	239 STO 12
81 STO 15	134 GTO 02	187*LBL 04	240*LBL 11
82 AON	135*LBL 33	188 RCL 11	241 2.005
83 PROMPT	136 CLA	189 STO 12	242 STO 01
84 AOFF	137 ARCL 08	190*LBL 05	243 E
85 0	138 ATOX	191 2.005	244 ST- 12
86 X<>F	139 65	192 STO 01	245 RCL 12
87 ASTO 07	140 X<>Y	193 E	246 64
88 ASTO 08	141 X<Y?	194 ST+ 12	247 X=Y?
89 ASTO 09	142 GTO 45	195 RCL 12	248 GTO 12
90 2.005	143 71	196 E1	249 RCL 12
91 STO 01	144 X<Y?	197 X=Y?	250 CLA
92*LBL 37	145 GTO 45	198 GTO 07	251 XTOA
93 RCL 07	146 ATOX	199 RCL 10	252 ARCL 11

253 ASTO 13	303 STO 01	353*LBL 36	403*LBL 26
254 ASTO 13	304 E	354 RCL 13	404 ISG 15
255*LBL 30	305 ST+ 12	355 RCL IND 01	405 GTO 10
256 RCL 13	306 RCL 12	356 X=Y?	406*LBL 27
257 RCL IND 01	307 72	357 GTO 31	407 ISG 15
258 X=Y?	308 X=Y?	358 ISG 01	408 GTO 12
259 GTO 27	309 GTO 18	359 GTO 36	409*LBL 28
260 ISG 01	310 E	360 GTO 19	410 ISG 15
261 GTO 30	311 ST+ 14	361*LBL 20	411 GTO 15
262 GTO 11	312 RCL 14	362 RCL 10	412*LBL 29
263*LBL 12	313 E1	363 STO 12	413 ISG 15
264 RCL 10	314 X=Y?	364 RCL 11	414 GTO 18
265 STO 12	315 GTO 18	365 STO 14	415*LBL 31
266 RCL 11	316 RCL 12	366*LBL 21	416 ISG 15
267 STO 14	317 CLA	367 2.005	417 GTO 20
268*LBL 14	318 XTOA	368 STO 01	418*LBL 32
269 2.005	319 ARCL 14	369 E	419 ISG 15
270 STO 01	320 ASTO 13	370 ST- 12	420 GTO 23
271 E	321*LBL 17	371 RCL 12	421*LBL 34
272 ST- 12	322 RCL 13	372 64	422 BEEP
273 RCL 12	323 RCL IND 01	373 X=Y?	423 "CONGRATS"
274 64	324 X=Y?	374 GTO 23	424 AVIEW
275 X=Y?	325 GTO 29	375 E	425 PSE
276 GTO 15	326 ISG 01	376 ST+ 14	426 "IN "
277 E	327 GTO 17	377 RCL 14	427 RCL 17
278 ST- 14	328 GTO 16	378 E1	428 INT
279 RCL 14	329*LBL 18	379 X=Y?	429 ARCL X
280 0	330 RCL 10	380 GTO 23	430 " TRIES"
281 X=Y?	331 STO 12	381 RCL 12	431 AVIEW
282 GTO 15	332 RCL 11	382 CLA	432 PSE
283 RCL 12	333 STO 14	383 XTOA	433 "AGAIN? Y/N"
284 CLA	334*LBL 19	384 ARCL 14	434 AON
285 XTOA	335 2.005	385 ASTO 13	435 PROMPT
286 ARCL 14	336 STO 01	386*LBL 22	436 AOFF
287 ASTO 13	337 E	387 RCL 13	437 ASTO 16
288*LBL 13	338 ST+ 12	388 RCL IND 01	438 "Y"
289 RCL 13	339 RCL 12	389 X=Y?	439 ASTO 17
290 RCL IND 01	340 72	390 GTO 32	440 RCL 16
291 X=Y?	341 X=Y?	391 ISG 01	441 RCL 17
292 GTO 28	342 GTO 20	392 GTO 22	442 X=Y?
293 ISG 01	343 E	393 GTO 21	443 GTO
294 GTO 13	344 ST- 14	394*LBL 23	"GALAXIS"
295 GTO 14	345 RCL 14	395 SF IND 15	444 RCL 26
296*LBL 15	346 X=0?	396 GTO 02	445 STOF LAG
297 RCL 10	347 GTO 20	397*LBL 24	446 CLX
298 STO 12	348 RCL 12	398 ISG 15	447 " BYE"
299 RCL 11	349 CLA	399 GTO 04	448 AVIEW
300 STO 14	350 XTOA	400*LBL 25	449 END
301*LBL 16	351 ARCL 14	401 ISG 15	
302 2.005	352 ASTO 13	402 GTO 07	

Space Invaders

[Ramón Cererols -Boletín Pont-Reyes N.1;\(October 1983\)](#)

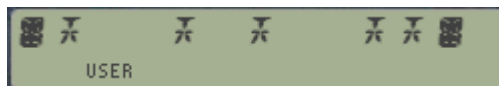
This program simulates a "Space Invaders" game that have become so popular, within the possibilities of the HP-41 platform. At any rate it is an interactive program relatively fast, and with its own attractiveness. The CX module is required for this version.

Program comments.

To play you need to start it using XEQ "ALIEN". The program greets you and uses the current time to produce a random seed that makes the game always different. This is followed with the message: "10 FOR 5", indicating that there are ten shots available to kill five invaders.



After a short pause the five invaders appear randomly spread between two starburst characters that demarcate the endings:



The ten places between the starburst chars are represented by numbers 0 to 9. When the invaders appear you have one PSE's worth time to shoot. This you accomplish by pressing a numeric key, without R/S (!), corresponding to the position occupied by an invader. Then three things may occur:

- if you hit it, one shot is used up and one fewer invader remain.
- if you miss, besides using one available shot a new invader is added to the string
- if you don't shoot within the allotted time you save the shot but the number of invaders is increased by two (!)



The process is repeated until all invaders are destroyed, or until the number of invaders exceeds the available shots. Be prepared to shoot, you'll need quick reflexes or else these little buggers proliferate fast!

The program informs who's won and asks for a new game. If no more games are requested it presents the final score and greets you b'bye. Multiple TONE instructions are executed during the action, - so get psyched!

Note; Lines 28 and 36 end with a blank space – there aren't any synthetic instructions.

Program listing:

01*LBL "ALIENS"

02 "GET READY"	46 XTOA	91 "AGAIN?"
03 AVIEW	47 X<> L	92 AON
04 7	48 DSE Y	93 PROMPT
05 PSIZE	49 GTO 03	94 AOFF
06 FIX 0	50 R-D	95 ATOX
07 CF 29	51 AROT	96 78
08 -2	52 FRC	97 X#Y?
09 STO 03	53 STO 00	98 GTO 00
10 E1	54 RCL 03	99 "SCORE: "
11 STO 04	55 XTOA	100 AVIEW
12 ,	56 XTOA	101 PSE
13 STO 05	57 SIGN	102 "ALIENS: "
14 STO 06	58 AROT	103 ARCL 05
15 TIME	59 CHS	104 ", YOU: "
16 HR	60 ENTER^	105 ARCL 06
17 24	61 CF 22	106 AVIEW
18 /	62 RASP	107 PSE
19 STO 00	63 AVIEW	108 BEEP
20*LBL 00	64 PSE	109 "SEE'YA"
21 5	65 PSE	110 PROMPT
22 STO 01	66 CLAXON	111*LBL 07
23 RCL 04	67 FC? 22	112 TONE 4
24 STO 02	68 GTO 08	113 TONE 6
25*LBL 01	69 +	114 DSE 01
26 CLA	70 AROT	115 GTO 04
27 ARCL 02	71 CLX	116 "YOU WON"
28 "` FOR "	72 ATOX	117 AVIEW
29 ARCL 01	73 X=Y?	118 BEEP
30 AVIEW	74 GTO 07	119 E
31 CLA	75 TONE 0	120 ST+ 06
32 RCL 04	76 TONE 0	121 GTO 06
33 RCL 01	77 SIGN	122*LBL 08
34 -	78 ST+ 01	123 TONE 9
35*LBL 02	79*LBL 04	124 TONE 9
36 "` "	80 DSE 02	125 RCL 03
37 DSE X	81 GTO 09	126 ST- 01
38 GTO 02	82*LBL 05	127*LBL 09
39 LASTX	83 "YOU LOST"	128 RCL 02
40 RCL 00	84 AVIEW	129 RCL 01
41*LBL 03	85 TONE 1	130 X>Y?
42 R-D	86 TONE 1	131 GTO 05
43 AROT	87 TONE 0	132 GTO 01
44 FRC	88*LBL 06	133 END
45 SIGN	89 E	
	90 ST+ 05	

Star Raiders

George Ruppert – PPCCJ V11N1 p14; (Jan/Feb 1981)

The enclosed game is similar to the Atari cassette Star Raiders. Just run and try it out. Some running information: when energy and number of Krylons are displayed, enter the number of the chosen Command or enter nothing.

Commands: #1 Galactic Map
 #2: Hyper-Warp
 #3: Attack Computer
 #4: Shields
 #5: Photons

Today I do not have time to give you more details, but I am sure that you can find out everything yourself or contact me...

George Ruppert (10819)
 Fohrenburgsstrasse 8
 A 6700 Bludenz
 Austria

Program listing:

01*LBL 99	25 ADV	46 "SE."	68 "5. PHOTONS"
02 RNG	26 ADV	47 ACA	69 PRA
03 STO 07	27 "YOUR	48 PRBUF	70 ADV
04 RTN	MISSION"	49 ADV	71 ADV
05*LBL "STAR"	28 ACA	50 ADV	72 XEQ 99
06 FIX 1	29 " IS TO"	51 "AVAILABLE "	73 4
07 SF 01	30 ACA	52 ACA	74 *
08 SF 04	31 6	53 "COMMANDS	75 INT
09 CF 00	32 SKPCHR	:"	76 STO 00
10 CF 02	33 "DESTROY THE	54 ACA	77 XEQ 99
11 CF 03	"	55 PRBUF	78 4
12 CF 05	34 ACA	56 ADV	79 *
13 CF 06	35 "KRYLON	57 "1. GALACTIC	80 INT
14 CF 07	WAR="	MAP"	81 E1
15 RCL 07	36 ACA	58 PRA	82 /
16 CLRG	37 PRBUF	59 "2.	83 ST+ 00
17 X=0?	38 "SHIPS THAT A"	HYPERWARP"	84 RCL 00
18 PI	39 ACA	60 PRA	85 STO 05
19 STO 07	40 "RE HEADED"	61 "3. ATTACK"	86 400
20 ADV	41 ACA	62 ACA	87 STO 02
21 ADV	42 3	63 " COMPUTER"	88 E1
22 SF 12	43 SKPCHR	64 ACA	89 STO 03
23 "STAR	44 "FOR YOUR	65 PRBUF	90 STO 04
RAIDERS"	STARBA"	66 "4. SHIELDS"	91*LBL 00
24 CF 12	45 ACA	67 PRA	92 XEQ 99

93 4	146 X#0?	199 E1	251 ADV
94 *	147 GTO 21	200 STO 04	252 CF 00
95 INT	148 XEQ 07	201 GTO 19	253 RCL 06
96 STO 06	149 PI	202*LBL 01	254 RCL 00
97 XEQ 99	150 CHS	203 ADV	255 X=Y?
98 4	151 STO 05	204 "GALACTIC	256 XEQ 06
99 *	152*LBL 19	MAP"	257 RCL 05
100 INT	153 1.5	205 PRA	258 X=Y?
101 E1	154 FS? 03	206 ADV	259 XEQ 07
102 /	155 ST- 02	207 "STARBASE :"	260 RTN
103 ST+ 06	156 ST- 02	208 ARCL 05	261*LBL 03
104 RCL 06	157 3	209 PRA	262 FC?C 03
105 RCL 05	158 FS? 02	210 "FIGHTER :"	263 SF 03
106 X=Y?	159 ST- 02	211 ARCL 00	264 "COMPUTER "
107 GTO 00	160 GTO 15	212 PRA	265 ACA
108 XEQ 01	161*LBL 18	213 "KRYLONS :"	266 FS? 03
109*LBL 15	162 RCL 00	214 ARCL 06	267 "ON"
110 CLD	163 RCL 06	215 PRA	268 FC? 03
111 RCL 02	164 X#Y?	216 ADV	269 "OFF"
112 X<0?	165 RTN	217 RTN	270 ACA
113 GTO 14	166 XEQ 99	218*LBL 02	271 PRBUF
114 RCL 03	167 .7	219 ADV	272 ADV
115 X=0?	168 X>Y?	220 "HYPERWARP"	273 RTN
116 GTO 16	169 RTN	221 PRA	274*LBL 04
117 "E"	170 SF 07	222 ADV	275 FS? 02
118 ARCL 02	171 TONE 8	223 "WHERE TO?"	276 RTN
119 "":K"	172 TONE 5	224 PROMPT	277 SF 02
120 ARCL 03	173 CF 21	225 3.3	278 E1
121 CF 21	174 "*:ATTACK:*	226 X<Y?	279 STO 01
122 TONE 9	175 AVIEW	227 RTN	280 "SHIELDS O.K."
123 AVIEW	176 SF 21	228 RDN	281 PRA
124 SF 21	177 RTN	229 X<> 00	282 ADV
125 17	178*LBL 20	230 CHS	283 RTN
126 PSE	179 DSE 04	231 RCL 00	284*LBL 05
127 XEQ IND X	180 GTO 19	232 +	285 FC? 07
128*LBL 17	181 SIGN	233 ABS	286 RTN
129 FC? 07	182 FS? 00	234 INT	287 "PHOTONS"
130 XEQ 18	183 ST+ 00	235 LASTX	288 PRA
131 RCL 05	184 ST+ 06	236 FRC	289*LBL 08
132 X<0?	185 XEQ 01	237 E1	290 E1
133 GTO 19	186 E1	238 *	291 "HOW
134 RCL 05	187 STO 04	239 +	MANY?"
135 INT	188 GTO 19	240 20	292 PROMPT
136 RCL 06	189*LBL 21	241 *	293 X<=0?
137 INT	190 DSE 04	242 RCL 02	294 GTO 08
138 -	191 GTO 19	243 X<Y?	295 X>Y?
139 X#0?	192 SIGN	244 RTN	296 GTO 08
140 GTO 20	193 E1	245 -	297 STO 08
141 RCL 05	194 /	246 CHS	298 STO [
142 FRC	195 FS? 00	247 STO 02	299 E
143 RCL 06	196 ST+ 00	248 "FIGHTER :"	300*LBL 09
144 FRC	197 ST+ 06	249 ARCL 00	301 E
145 -	198 XEQ 01	250 PRA	302 XEQ 99

303 -	335 *	366 PRA	398 "DESTROYED"
304 *	336 INT	367 ADV	399 ACA
305 TONE 9	337 ST- 01	368 ADV	400 PRBUF
306 TONE 9	338 E1	369 TONE 8	401 CF 12
307 DSE [339 *	370 TONE 5	402 ADV
308 GTO 09	340 ST- 02	371 TONE 8	403 ADV
309 E	341 RCL 01	372 TONE 5	404 ADV
310 -	342 X<0?	373 TONE 8	405 ADV
311 CHS	343 GTO 14	374 TONE 5	406 ADV
312 FC? 03	344 DSE X	375 CF 12	407 STOP
313 GTO 10	345 ""	376 RTN	408*LBL 16
314 .1	346 X>0?	377*LBL 07	409 BEEP
315 +	347 RTN	378 BEEP	410 SF 12
316*LBL 10	348 "SHIELDS	379 SF 01	411 ADV
317 "KRYLON "	LOW"	380 CF 02	412 FMT
318 .8	349 CF 21	381 XEQ 04	413 "SUPER"
319 X<Y?	350 TONE 2	382 CF 03	414 ACA
320 GTO 12	351 AVIEW	383 XEQ 03	415 PRBUF
321 "MISSED"	352 SF 21	384 SF 04	416 FMT
322 GTO 13	353 RTN	385 400	417 "YOU WON"
323*LBL 12	354*LBL 06	386 STO 02	418 ACA
324 "DESTROYED"	355 SF 00	387 BEEP	419 PRBUF
325 CF 07	356 ADV	388 RTN	420 CF 10
326 DSE 03	357 ADV	389*LBL 14	421 ADV
327 ""	358 SF 12	390 BEEP	422 ADV
328*LBL 13	359 "RED ALERT"	391 SF 12	423 ADV
329 PRA	360 TONE 8	392 ADV	424 ADV
330 ADV	361 TONE 5	393 FMT	425 ADV
331 FC? 02	362 TONE 8	394 "JUST"	426 STOP
332 GTO 14	363 TONE 5	395 ACA	427 END
333 XEQ 99	364 TONE 8	396 PRBUF	
334 RCL 08	365 TONE 5	397 FMT	

Space Wars Interactive

[Roger M. Stenerson – UPL #00655C](#)

This program was written to take advantage of something that I discovered the HP-41C could do. What I discovered was that it is possible to make the display scroll rapidly from left to right by using code similar to the following:

```
01 LBL 00
02 SE 25
03 AVIEW
04 0
05 1/X
06 "LBL 01
07 GTO 01
```

This will cause whatever is in the ALPHA register to scroll from left to right. The speed of the scrolling being determined by the number of instructions between lines 6 and 7. After I had figured out how to use this feature I wrote this program.

As the captain of a spacecraft, your mission is to destroy as many of the enemy as possible. There are three types of enemy spacecraft freighters battleships and fighters. Each time you destroy one of the enemy spacecraft you score a certain number of points based on the type of the craft you destroyed. They are denoted by and worth the following:

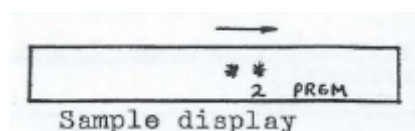
```
"  *" - fighter. worth 20 p01nts
"  ***" - battleship. worth 10 points
"****" - Freighter, worth 5 points
```

Your spacecraft can still function after two hits. On the third hit the spacecraft is destroyed and the game is over. The calculator will then display your score.

However it is possible to repair your spacecraft after the first or second hits. Periodically, a star base will appear denoted by "888". To dock with it you go through the same procedure that you do when shooting down the enemy; however instead of entering a "3" you enter a "2". If you are successful your ship will be repaired and as good as new. If on the other hand you are unsuccessful nothing happens and the game continues.

Destroying the base is not recommended, and results in your being relieved of your duties. It also ends the game. Equally bad is attempting to dock with the enemy. As you must turn off your defense fields to dock, you are defenseless, and the enemy can easily destroy your spacecraft which ends the game.

Of course the object of the game is to get as many points as possible. The calculator will also keep track of the highest score since the program was loaded. (high score stored in REG 02). During the game, fighters will come up 25% of the time, Battleships and freighters 35% and star bases 5% . Good luck!



Warning: It is very easy to spend much time playing this game.

Example:

Due to the randomness of the user pushing the R/S key, it is impossible to give an example that can be duplicated. However, I will show parts of a sample run.

Input	Function	Display	Comments
	XEQ"SPACE"	READY?"	Begin playing
	R/S	"**" ->	Spacecraft moving across the LCD
	R/S	0	Stop calculator
3	R/S	"YOU HAVE BEEN HIT"	missed him
		"*" ->	spacecraft moving
	R/S	0	stop calculator
3	R/S	"YOU SCORE A KILL"	got him!
		"***" ->	spacecraft moving
	R/S	0	stop calculator
3	R/S	"YOU HAVE BEEN HIT"	missed again
		"888" ->	star base moving
	R/S	0	stop calculator
2	R/S	"DOCKING COMPLETE"	successful docking
		"SHIP IS REPAIRED"	
,		"*" ->	spacecraft moving
	R/S	0	stop calculator
2	R/S	"YOU HAVE BEEN HIT"	
		"SHIP DESTROYED"	
		"YOUR SCORE IS 50"	
		"HIGH SCORE IS 50"	
		"AGAIN?"	

Program listing:

01*LBL "SPACE"	21 XEQ 10	41 GTO 00
02 ,	22 5	42*LBL 02
03 STO 02	23 -	43 7
04 "SEED=?"	24 X>0?	44 -
05 PROMPT	25 GTO 02	45 X>0?
06 STO 04	26 .01	46 GTO 02
07*LBL 05	27 STO 00	47 .009
08 FIX 0	28 *	48 STO 00
09 3	29 XEQ 14	49 "***"
10 STO 03	30 7	50 XEQ 14
11 CF 06	31 X=Y?	51 6
12 SF 02	32 GTO 12	52 X=Y?
13 0	33 8	53 GTO 06
14 STO 01	34 GTO 03	54 RDN
15 "READY?"	35*LBL 12	55 7
16 PROMPT	36 XEQ 11	56 X=Y?
17 CLA	37 TONE 9	57 GTO 06
18 AVIEW	38 AVIEW	58 6
19*LBL 00	39 20	59 GTO 03
20 20	40 ST+ 01	60*LBL 06

61 XEQ 11	113 X#Y?	165*LBL 04
62 TONE 7	114 GTO 02	166 RCL 00
63 AVIEW	115 "DOCKING COMPLET"	167 INT
64 10	116 ""E"	168 RTN
65 ST+ 01	117 AVIEW	169*LBL 01
66 GTO 00	118 "YOUR SHIP "	170 CF 22
67*LBL 02	119 ""IS REPAIRED"	171 SF 25
68 7	120 FS?C 06	172 0
69 -	121 AVIEW	173 AVIEW
70 X>0?	122 3	174 LN
71 GTO 02	123 STO 03	175*LBL 09
72 .008	124 GTO 00	176 FS? 22
73 STO 00	125*LBL 02	177 RTN
74 "****"	126 "YOU DESTROYED "	178 ISG 00
75 XEQ 14	127 ""YOUR BASE"	179 GTO 09
76 5	128 AVIEW	180 RTN
77 X<=Y?	129 PSE	181*LBL 03
78 GTO 13	130 "AND HAVE BEEN "	182 10
79 4	131 ""RELIEVED"	183 XEQ 10
80 GTO 03	132 AVIEW	184 X>Y?
81*LBL 13	133 PSE	185 GTO 00
82 RDN	134 "OF COMMAND"	186 SF 06
83 8	135 GTO 08	187 "YOU HAVE JUST "
84 X>Y?	136*LBL 10	188 ""BEEN HIT"
85 GTO 13	137 RCL 04	189 TONE 3
86 4	138 9821	190 TONE 3
87 GTO 03	139 *	191 AVIEW
88*LBL 13	140 .211327	192 DSE 03
89 XEQ 11	141 +	193 GTO 00
90 TONE 5	142 FRC	194 PSE
91 AVIEW	143 STO 04	195*LBL 07
92 5	144 *	196 "YOUR SHIP IS "
93 ST+ 01	145 RTN	197 ""DESTROYED"
94 GTO 00	146*LBL 11	198 BEEP
95*LBL 02	147 "YOU SCORED A "	199*LBL 08
96 .008	148 ""KILL"	200 AVIEW
97 STO 00	149 RTN	201 "YOUR SCORE IS "
98 "888"	150*LBL 14	202 ARCL 01
99 XEQ 01	151 XEQ 01	203 AVIEW
100 CLA	152 CLA	204 PSE
101 AVIEW	153 AVIEW	205 RCL 02
102 RCL 00	154 2	206 RCL 01
103 INT	155 X#Y?	207 X>Y?
104 5	156 GTO 04	208 STO 02
105 X>Y?	157 "YOU ATTEMPTED "	209 "HIGH SCORE IS "
106 GTO 00	158 ""TO DOCK"	210 ARCL 02
107 RDN	159 AVIEW	211 AVIEW
108 7	160 PSE	212 PSE
109 X<Y?	161 "WITH THE ENEMY"	213 "AGAIN?"
110 GTO 00	162 AVIEW	214 PROMPT
111 RCL Z	163 PSE	215 GTO 05
112 2	164 GTO 07	216 END

FOURS, Virtually Connected

David Kipling -DataFile V2N4 p5 ; (Sep/Oct 1983)

Anyone who has ever tried to play chess without a board solely by calling out the moves will realize the difficulty of it; chaos easily ensues if you lose your place, and without a referee keeping track arguments can surround a supposedly winning position.

With this in mind I devised a new 'virtual' game, based on a 'real' game where players drop colored counters into a 6*6 matrix with the intention of connecting an, four of their counters vertically, horizontally, or diagonally.

To play this 'virtual' version ('virtual' because there is no equipment and the players must remember the position in their heads) I labeled each of the columns with the letters A through F. Each player takes it in turn to callout a letter corresponding to the column that he is dropping his imaginary 'counter' into. This continues until one player thinks that he has won, or until the matrix is full (a draw). I quickly decided that my 41C would be an ideal referee to judge who had won; this was the birth of "FOURS".

To start play, XEQ 'FOURS'; the prompt "X?" appears. One player is assigned crosses, the other noughts. The 'X' player now presses the key with the blue letter that corresponds to the column he wants his counter dropping into (A, B, C, D, E, or F). The prompt "0?" now appears and the other player goes thru' the same routine. "X?" reappears and the cycle continues.

If a player thinks that he has won, the [TAN] key is pressed at a "X?" or "0?" prompt and the game stops. This shows the contents of column A; e.g. X000X (the left-hand side corresponds to the bottom of the column). By pressing R/S repeatedly, column B thru 'F' are shown in sequence.

Example:

This would be seen as the following set-up;

Key	Display	Column
TAN	X 0 0 0 X	A
R/S	X X 0	B
R/S	0 0 X	C
R/S	X X 0 X	D
R/S	X 0 0	E
R/S	0 X	F

X					
0			X		
0	0	X	0	0	
0	X	0	X	0	X
X	X	0	X	X	0
A	B	C	D	E	F

In this example, crosses have got a line.

My rules are that if the player who called victory was correct, even if the other player has a line as well, he wins; the other player should have noticed his line. If he was wrong, however, he automatically loses, even if the other player hasn't got a line either. It is a source of great amusement if your opponent wins, although you've had a line for the last 9 moves but didn't realize it!

After a game, press the [SIN] key to clear the state and start another game. Oh yes, and if you try to put a counter into a full column (6 counters), the program will pass over the entry and you'll have wasted a go. The game always starts with a "X?" prompt so we usually swap symbols each time.

Notes.

Columns A thru F correspond to REG's R00 thru R05. Flag 27 is set (USER on) to enable use of local labels; flag 00 is also used. "FOURS" will run on any HP-41 system with SIZE>0 006; no synthetics are used.

Anyway, have fun; I think you'll find this game a bit more taxing than the usual battleship or pontoon games for the 41C! My next job is (hopefully) to make a routine to scan the columns upon executing LBL J and to see if there are any connected counters. If I have any success I'll let you know in a future issue. If anyone has any ideas for this routine, or if anyone has devised any other "virtual" games for the 41C I'd be delighted to hear from them.

Program listing:

01*LBL "FOURS"	26 ARCL 01	51 XEQ 03
02*LBL H	27 XEQ 03	52 ASTO 05
03 SF 27	28 ASTO 01	53 GTO 02
04 CF 00	29 GTO 02	54*LBL 03
05 CLA	30*LBL C	55 FS? 00
06 ASTO 00	31 CLA	56 GTO 00
07 ASTO 01	32 ARCL 02	57 "X"
08 ASTO 02	33 XEQ 03	58 SF 00
09 ASTO 03	34 ASTO 02	59 RTN
10 ASTO 04	35 GTO 02	60*LBL 00
11 ASTO 05	36*LBL D	61 "0"
12*LBL 02	37 CLA	62 CF 00
13 "0?"	38 ARCL 03	63 RTN
14 FC? 00	39 XEQ 03	64*LBL J
15 "X?"	40 ASTO 03	65 ,
16 AVIEW	41 GTO 02	66*LBL 01
17 RTN	42*LBL E	67 RCL IND X
18*LBL A	43 CLA	68 STOP
19 CLA	44 ARCL 04	69 RDN
20 ARCL 00	45 XEQ 03	70 1
21 XEQ 03	46 ASTO 04	71 +
22 ASTO 00	47 GTO 02	72 GTO 01
23 GTO 02	48*LBL F	73 END
24*LBL B	49 CLA	
25 CLA	50 ARCL 05	

Flip-Flop for the HP-41C/CV/CX

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

Flip-Flop challenges you to change a string of 8 zeroes and 1 one (.000010000) to 1 zero and 8 ones (.111101111). Only positions containing ones can be specified for flipping. Flipping a one to a zero will automatically flip adjacent zeroes to ones and ones to zeroes. Flipping a one in either end position will flip the opposite end as well as the adjacent position.

Positions are: previous move, 123456789. Note that the position to the left of the comma always shows the last move unless the last move tried to flip a zero, at which time it will show zero.

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] FLIP	0,000010000
3	Key in position to flip.	x		(x),()
	Repeat step 3 until successful.			

Example

Keystrokes:	Display:
[XEQ] [ALPHA] SIZE	
[ALPHA] 013	
[XEQ]	
[ALPHA] FLIP [ALPHA]	0,000010000
5	5,000101000
6	6,000110100
5	5,000001100
.	.
.	.
.	.

Program listing:





01 LBL "FLIP"

02 CF 28	36 ST- 00	71 ST- 00
03 FIX 09	37 9	72 RCL 09
04 CLRG	38 RCL 12	73 CHS
05 9	39 X=Y?	74 STO 09
06 STO 12	40 GTO 09	75 ST- 00
07 LBL 00	41 1	76 GTO 01
08 10	42 X=Y?	77 LBL 05
09 RCL 12	43 GTO 04	78 CF 22
10 CHS	44 ISG 12	79 VIEW X
11 Y^X	45 LBL 10	80 LBL 06
12 STO IND 12	46 RCL IND 12	81 PSE
13 DSE 12	47 CHS	82 FS?C 22
14 GTO 00	48 STO IND 12	83 RTN
15 RCL 05	49 ST- 00	84 GTO 06
16 STO 00	50 DSE 12	85 LBL 07
17 CHS	51 DSE 12	86 RCL IND 12
18 STO 05	52 RCL IND 12	87 X<0?
19 LBL 01	53 CHS	88 RTN
20 RCL 00	54 STO IND 12	89 RCL 00
21 .111101111	55 ST- 00	90 VIEW X
22 X<>Y	56 GTO 01	91 XEQ 06
23 X=Y?	57 LBL 09	92 GTO 03
24 GTO 02	58 RCL 01	93 LBL 02
25 RCL 10	59 CHS	94 RCL 10
26 +	60 STO 01	95 +
27 XEQ 05	61 ST- 00	96 VIEW X
28 LBL 03	62 RCL 08	97 PSE
29 STO 12	63 CHS	98 CLA
30 STO 10	64 STO 08	99 FIX 00
31 XEQ 07	65 ST- 00	100 CF 29
32 ISG 11	66 GTO 01	101 SF 28
33 LBL 10	67 LBL 04	102 ARCL 11
34 CHS	68 RCL 02	103 "┌ FLIPS"
35 STO IND 12	69 CHS	104 AVIEW
	70 STO 02	105 END


Maze Construction & Play

Layne K. Johnson - UPL #00663C

The program makes a path through the maze. Path construction is governed by two things: One, it can't double back upon itself, and Two, the path can't exit through the bottom. Each square is represented by a number 0 – 5, and each number has its own meaning:

- 0 -> depends on difficulty
- 1 -> 
- 2 -> 
- 3 -> 
- 4 -> 
- 5 -> out of bounds.



The second half of the program allows the user to play the maze by interpreting the numbers. The program ANDs the four square's walls that surround you with the walls in your square.

For example: 
 1] 2] 4 -> produces: | |
 2]

If you exceed the boundaries of the maze you'll be in a square with four walls. Go back through the wall you just went thru.

User Instructions:

1. Enter the program, set status.
2. Input seed ($0 < x < 1$)
 The program will now generate the maze. Flag 1 indicator will come on as the program generates the path. Flag 2 will come on the programs completes the maze. When the maze is done the calculator will Beep and turn itself off.
3. Turn calculator ON, ser USER mode
4. Start the program – XEQ "MAZE", then pressing [A]
5. Select difficulty level, 0 – 3, where zero is the easiest and 3 the hardest.
 The calculator will show your current position as a string of 4 symbols denoting the Left, Top, Right, Bottom walls, represented as follows:

-  -> opening ,
-  -> wall

6. Input the direction you wish to go, as follows:

[B] -> Left		[C]
[C] -> Up	←	[B] [D]→
[D] -> Right		[E]
[E] -> Down		

7. Repeat step 6 until you exit the maze. Exit condition will be shown by "OUT"

Example:

Using the seed 0.2912576789, start the program: XEQ "MAZE".

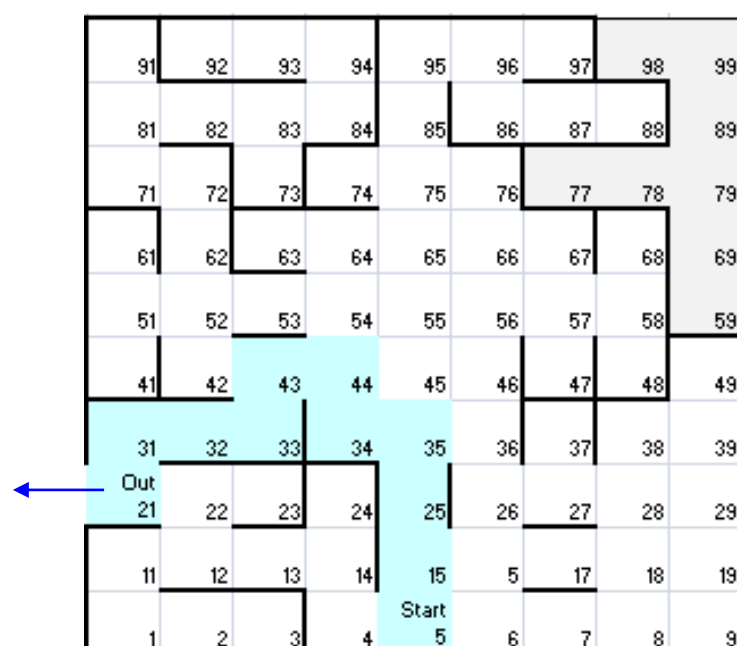
Wait for the calculator to construct the maze. On completion it'll switch itself off.

Turn the calculator ON, switch USER on and press [A]

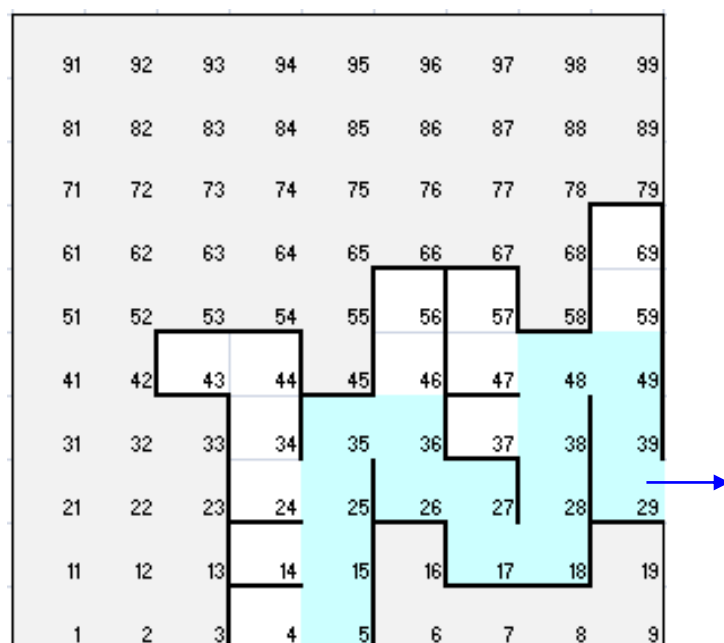
See	Where am I?	Press	Action
DIFFICULTY?	choose 1	1, R/S	
0000 -- 5	entrance square	[C], R/S	move up
0000 -- 15	next square up	[B], R/S	move left
0000 -- 14	next square left	[C], R/S	move up
0000 -- 24	next square up	[D], R/S	move right
0000 -- 25	etc.	[D], R/S	move right again
0000 -- 26		[E], R/S	move down
0000 -- 16		[E], R/S	down again
0000 -- 6		[D], R/S	move right
0000 -- 7		[D], R/S	right again
0000 -- 8		[D], R/S	and again
0000 -- 9	check exit on right	[C], R/S	move up
0000 -- 19		[C], R/S	and up again
0000 -- 29	oops, dead end!	[E], R/S	move down
0000 -- 19	back out	continue till exit square is found	
....
0000 -- . XX	last three moves	[B], R/S	move left
0000 -- XX		[B]	and left again
YOU'RE OUT	the exit!		

The sketches following below represent two other instances of mazes generated by the program, and the path used to exit them (it's much easier when you have the overview picture, isn't it?)

Note that the starting square is always #5, and that some parts of the maze may not be practicable (i.e. can't be accessible) depending on the random structure derived from your input values.



Seed = 0,8529637419 ; Difficulty – 3



Seed = 0,0356819427 ; Difficulty = 0

Program listing:

01*LBL "MAZES"

02 SIZE?	50 X#Y?	99 2	148 ST+ IND 25
03 26	51 GTO 00	100 X=Y?	149*LBL 00
04 X>Y?	52 RCL 25	101 ISG 25	150 RCL 21
05 PSIZE	53 19	102 RDN	151 10^X
06 RCL Z	54 X=Y?	103 3	152 RCL 21
07*LBL H	55 GTO 14	104 X#Y?	153 ST* 20
08 CLRG	56*LBL 00	105 GTO 00	154 RDN
09 STO 24	57 XEQ 19	106 RCL 21	155 RCL 20
10 -5	58 ST+ IND 25	107 ST/ 23	156 X#Y?
11 10^X	59 RCL 20	108 RTN	157 GTO 13
12 STO 23	60 XEQ 17	109*LBL 00	158 ISG 25
13 10	61 RCL IND 25	110 RDN	159 GTO 20
14 STO 21	62 RCL 23	111 4	160 CF 02
15 STO 25	63 /	112 X=Y?	161 CLST
16 SF 01	64 INT	113 DSE 25	162 GTO A
17*LBL 12	65 RCL 21	114 RTN	163*LBL 16
18 E	66 /	115*LBL 14	164 4.789
19 3	67 FRC	116 XEQ 19	165*LBL 18
20 RCL 25	68 X=0?	117 ST+ IND 25	166 RCL 24
21 10	69 GTO 00	118 RCL 25	167 9821
22 -	70 2	119 RCL 21	168 *
23 X#0?	71 RCL 20	120 *	169 .211327
24 RDN	72 X>Y?	121 RCL 23	170 +
25 +	73 CHS	122 LOG	171 FRC
26 XEQ 18	74 +	123 CHS	172 STO 24
27 E	75 ABS	124 E	173 *
28 +	76 XEQ 17	125 -	174 INT
29 STO 20	77 XEQ 19	126 +	175 RTN
30 3	78 ST- IND 25	127 STO 22	176*LBL 19
31 X#Y?	79 E	128 10.019	177 RCL 20
32 GTO 00	80 ST+ 22	129 STO 25	178 RCL 23
33 RCL 23	81 RCL 22	130 CF 01	179 *
34 -9	82 5	131 SF 02	180 RTN
35 10^X	83 X>Y?	132*LBL 20	181*LBL A
36 X=Y?	84 GTO 12	133 RCL 21	182 .0111
37 GTO 14	85 RCL 24	134*LBL 13	183 STO 01
38*LBL 00	86 GTO H	135 STO 20	184 XEQ 03
39 RCL 20	87*LBL 00	136 RCL IND 25	185 STO 04
40 E	88 STO 22	137 RCL 20	186 XEQ 03
41 X#Y?	89 GTO 12	138 *	187 STO 03
42 GTO 00	90*LBL 17	139 INT	188 XEQ 03
43 RCL 23	91 E	140 RCL 21	189 STO 02
44 .1	92 X#Y?	141 /	190 .2222
45 X=Y?	93 GTO 00	142 FRC	191 STO 05
46 GTO 14	94 RCL 21	143 X#0?	192 "DIFFICULTY?"
47*LBL 00	95 ST* 23	144 GTO 00	193 PROMPT
48 RCL 20	96 RTN	145 XEQ 16	194 3
49 2	97*LBL 00	146 RCL 20	195 X>Y?
	98 RDN	147 /	196 X<>Y

197 STO 07	239 *	281 99	323 FRC
198 4	240 INT	282 -	324 RCL 21
199 +	241 E3	283 RCL 06	325 *
200 XEQ IND X	242 /	284 +	326 9
201 104	243 ST+ 06	285 >"-"	327 X=Y?
202 STO 09	244 RCL 21	286 AIP	328 GTO 00
203 GTO 10	245 CHS	287 PROMPT	329 RDN
204*LBL 03	246 XEQ 01	288*LBL B (Left)	330 10^X
205 RCL 21	247 E2	289 E	331 RCL IND 25
206 *	248 *	290 ST- 09	332 *
207 FRC	249 INT	291 GTO 10	333 FRC
208 LASTX	250 RCL 21	292*LBL C (Up)	334 RCL 21
209 INT	251 /	293 RCL 21	335 *
210 4	252 FRC	294 ST+ 09	336 INT
211 10^X	253 E3	295 GTO 10	337 STO 25
212 /	254 /	296*LBL D (Right)	338 RCL IND 25
213 +	255 ST+ 06	297 E	339 RTN
214 RTN	256 5	298 ST+ 09	340*LBL 00
215*LBL 10	257 STO 25	299 GTO 10	341 RCL 05
216 0	258 " "	300*LBL E (Down)	342 RTN
217 XEQ 01	259*LBL 08	301 RCL 21	343*LBL 09
218 STO 06	260 DSE 25	302 ST- 09	344 "YOU'RE OUT"
219 -1	261 GTO 00	303 GTO 10	345 PROMPT
220 XEQ 01	262 GTO 02	304*LBL 01	346*LBL 07
221 E3	263*LBL 00	305 RCL 09	347 CLX
222 *	264 RCL 06	306 RCL 22	348 STO 00
223 INT	265 RCL 21	307 X=Y?	349 RTN
224 RCL 21	266 *	308 GTO 09	350*LBL 06
225 /	267 FRC	309 RDN	351 .101
226 FRC	268 STO 06	310 +	352 STO 00
227 ST+ 06	269 LASTX	311 2	353 RTN
228 RCL 21	270 INT	312 10^X	354*LBL 05
229 XEQ 01	271 2	313 X>Y?	355 XEQ 06
230 E3	272 X>Y?	314 GTO 00	356 XEQ 03
231 *	273 GTO 00	315 X<>Y	357 STO 00
232 FRC	274 >"0"	316 199	358 RTN
233 RCL 21	275 GTO 08	317 X<=Y?	359*LBL 04
234 /	276*LBL 00	318 GTO 00	360 .1111
235 ST+ 06	277 >"0"	319 RDN	361 STO 00
236 E	278 GTO 08	320 RCL 21	362 END
237 XEQ 01	279*LBL 02	321 /	
238 RCL 21	280 RCL 09	322 STO 25	

Mazes for the HP-41

JM Baillard –

The program hereunder generates a pseudo-random rectangular maze of dimensions $n \times m$. You place a random seed in register R00, n in register Y and m in register X and XEQ "MAZE"

The algorithm uses backtracking (cf reference [1]):

Starting at register R01, the HP41 successively finds unvisited neighbors and deletes the walls between them:

- When it becomes impossible, it backtracks until it finds an unvisited cell.
- When that also becomes impossible, the HP41 returns to register R01 and we have our maze.

Data Registers and User Instructions

Registers R01, R02,, Rmm are the 1st raw.
Rm+1, Rm+2,, R2m are the 2nd raw and so on...

STACK	INPUTS	OUTPUTS
Y	n	/
X	m	1.eee

Where n = number of rows, m = number of columns, and
1.eee = cntrol number of the maze with eee = $m.n$

Example:

Let's try with $n = 7$ and $m = 10$

If we choose $r = 1$ as the random seed: 1, STO 00

```
7 ENTER^
10 XEQ "MAZE" >>>> 1.070 ---Execution time = 5m11s---
```

Each register now contains a number of the form a.bcdef

$a = 0$ for an unvisited cell
 $a = 1$ for a visited cell

So, at the end, all the cells have been visited and $a = 1$

The walls $\begin{array}{c} \text{ } \\ \text{ } \\ \text{ } \end{array} \begin{array}{c} \text{ } \\ \text{ } \\ \text{ } \end{array} \begin{array}{c} \text{ } \\ \text{ } \\ \text{ } \end{array}$ are numbered this way. The walls $n^{\circ}3$ and $n^{\circ}4$ are in fact the walls

1 and 2 of other cells, so we only deal with walls 1 & 2

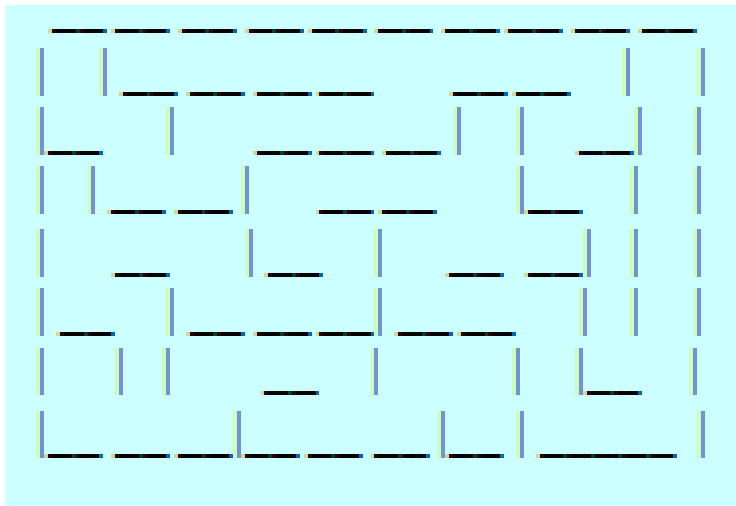
bc = 00 if the walls 1 & 2 are both deleted
 bc = 10 if the wall 2 only is deleted
 bc = 02 if the wall 1 only is deleted
 bc = 12 if the walls 1 & 2 are not deleted

The decimals d,e,f indicate the previous visited cell.

Thus, we only have to take b,c into account to draw the maze.
 (We assume that the edges of the rectangle are already drawn)

For R01 bc = 02 , that gives the first cell on the left: |
 For R02 bc = 10 , ----- 2nd --- of the 1st raw: — ... and so on ...

So, we get a maze that looks (approximately) as shown below:



Program Remarks:

Note that the driver program "MAZE+" and the main routine "MAZE" are consolidated into the same listing. The driver program will prepare all the input values for you with easy prompts.

Perhaps will you find better characters to display walls n°1 & n°2 in a cell ?

Program listing:

01*LBL "MAZE+"	09 *	17 STO [25 GTO 00
02 "SEED=?"	10 SIZE?	18 X<>Y	26 SIGN
03 PROMPT	11 X<>Y	19 STO \	27 STO]
04 STO 00	12 X>Y?	20 *	28 ST+ 01
05 "N^M=?"	13 PSIZE	21 ,12	29*LBL 10
06 PROMPT	14 RDN	22*LBL 00	30 FS? 10
07 RCL X	15 RDN	23 STO IND Y	31 VIEW]
08 RCL Z	16*LBL "MAZE"	24 DSE Y	32 CF 01

33 CF 02	78*LBL 02	123 ST+ Y	168*LBL 08
34 CF 03	79 CLX	124 ,1	169 "C"
35 CF 04	80 SIGN	125 GTO 06	170 ARCL X
36 CLX	81 RCL]	126*LBL 02	171 "`="
37 STO _	82 RCL [127 SIGN	172 ARCL IND X
38 RCL \	83 MOD	128 RCL]	173 FRC
39 RCL]	84 X=Y?	129 ST+ Y	174 E1
40 RCL [85 GTO 02	130 ,02	175 *
41 ST* Z	86 RCL]	131 GTO 06	176 INT
42 +	87 DSE X	132*LBL 03	177 ST- L
43 X>Y?	88 INT	133 RCL]	178 SF 05
44 GTO 02	89 RCL IND X	134 RCL [179 X=0?
45 RCL IND X	90 INT	135 -	180 GTO 08
46 INT	91 X#0?	136 STO Y	181 95
47 X#0?	92 GTO 02	137 ,1	182 XTOA
48 GTO 02	93 SF 04	138 GTO 06	183 RDN
49 SF 01	94 ISG _	139*LBL 04	184 CF 05
50 ISG _	95*LBL 02	140 RCL]	185*LBL 08
51*LBL 02	96 X<> _	141 E	186 X<> L
52 RCL]	97 X=0?	142 -	187 E1
53 RCL [98 GTO 07	143 STO Y	188 *
54 MOD	99 RCL 00	144 ,02	189 INT
55 X=0?	100 R-D	145*LBL 06	190 X=0?
56 GTO 02	101 FRC	146 ST- IND Y	191 GTO 08
57 SIGN	102 STO 00	147 X<> Z	192 FS?C 05
58 RCL]	103 *	148 X<>]	193 "` "
59 +	104 INT	149 E5	194 33
60 RCL IND X	105 SIGN	150 /	195 XTOA
61 INT	106 ST+ L	151 E	196 RDN
62 X#0?	107 FS? 04	152 +	197*LBL 08
63 RCL 02	108 4	153 ST+ IND]	198 RDN
64 SF 02	109 FS? 03	154 GTO 10	199 AVIEW
65 ISG _	110 3	155*LBL 07	200 ISG X
66*LBL 02	111 FS? 02	156 RCL IND]	201 GTO 08
67 RCL]	112 2	157 E2	202 X<>Y
68 RCL [113 FS? 01	158 *	203 FIX 4
69 -	114 E	159 FRC	204 SF 29
70 X<=0?	115*LBL 05	160 E3	205 RCL [
71 GTO 02	116 RDN	161 *	206 RCL \
72 RCL IND X	117 DSE L	162 STO]	207 *
73 INT	118 GTO 05	163 X#0?	208 E3
74 X#0?	119 GTO IND T	164 GTO 10	209 /
75 GTO 02	120*LBL 01	165 ENTER^	210 ISG X
76 SF 03	121 RCL [166 FIX 0	211 CLA
77 ISG _	122 RCL]	167 CF 29	212 END

XF/M Mazes.

Erik Christensen – PPCCJ V10N5p30; (June 1983)

You and your compass are in the midst of a labyrinth of passages. Some of the passages close off when you walk through them, sealing you off. You must find your way out of the maze without getting trapped. In each room there are possible 4 passages that might be open. The 4 possible directions in a room are North, East, South, and West. The maze is mapped in a text file in extended memory, with one character being one room.

You are positioned to the file in a X, Y position. For example, if you were 2 lines down, 10 characters over, in the display you would see "<2,10>DIR-NS" if the possible directions of travel were North and South. Then, to move south you would press the "S" alpha key (while the program runs), and the display would be updated. The program will generate random mazes, but it takes a while. If you have any kind of text file in extended memory that has relatively uniform record lengths, then it will work fine as a maze. Imagine, weaving through your favorite words!

Instructions.

#	SEE	Do	Comments
1a		XEQ "MAZE"	
1b	NAME?	File Name, R/S	If you enter a name not yet used, the program will generate a new maze (go to step 2), else as long as the names file is a text file, the program will use it as a maze. (go to step 4)
2a	SIZE?	Maze size (R/S)	When a dimension of n is entered, the maze will be n by n large. If there is not enough room in extended memory for the maze, you'll see a re-prompt.
2b	SEED?	Seed, R/S	Random number generator seed, 0<x<1. See display count from 1 to dimension of maze, then go to step 4
3a	START?X?Y	X, Enter, Y, R/S	Maze starting coordinates (where in the maze you are initially placed). Go to step 4
4a	<x,y>DIR-	[KEY]	Shows x,y coordinates, and possible directions (NESW). Press key that corresponds to direction you wish to travel. Go to step 4 if you move out of the text, go to step 5 and you have won!. If you press R/S as the direction then you go to step 6.
5a	YOU'RE OUT		You got out of the maze. R/S to start another one.
6a	OK		Flags are restored, to restart at current position, leave regs. R00 to R02 alone and R/S. In you get trapped, go to step 1

Example.

Use a predefined maze pattern to fill a file, then escape it.

```

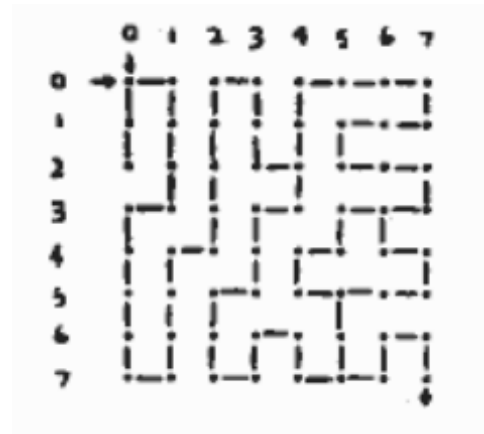
11, "MAZE 1", CRFLAS
"AGAGAEEG", APPREC
"00000AEI", APPREC
"K00MHMEG", APPREC
"R00RIRI", APPREC
"0R0RIMG", APPREC
"00RMIEI", APPREC
"000AG0AG", APPREC
"MIIMFIO", APPREC

XEQ "XMAZE"      NAME?

"MAZE 1", R/S      START? X?Y
0, ENTER?, R/S      20,01DIR-ES

E                  20,11DIR-SW
S                  21,11DIR-NS
..... etc....

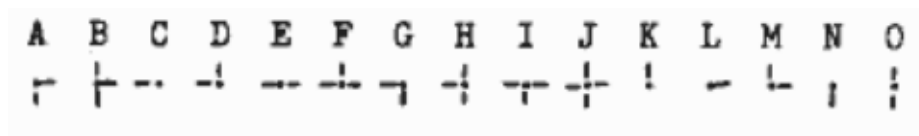
```

**Technical Notes**

The program finds out what passages are open by:

- Getting the character out of the file and into the alpha register
- Getting the ASCII code out of the alpha register and into the X-register
- Putting the ASCII code into flags 0-4
- Then checking flags 0-3 for on/off status that represents an open/closed passage in the room
- Corresponding flag status to directions (0-North, 1-East, 2-South, 3-West)
- Displaying the possible directions of travel

The following letters (A-O) correspond to different combinations of open passages:



With the above information you could easily write your own mazes. And who says that the only thing you can do with words is look at them? Text files never cease to aMAZE. (sorry).

Program listing:

See below the complete code, including the loading program for the example given above.

01*LBL "XMAZE"

02 SIZE?	52 RCL 01	103 SEEKPT	154 DSE 02
03 3	53 RNG	104 FC?C 25	155 RTN
04 X>Y?	54 STO 01	105 GTO 08	156 RTN
05 PSIZE	55 16	106 GETREC	157*LBL 04
06 RCLFLAG	56 *	107 SEEKPT	158 RCLFLAG
07 STO 00	57 CLA	108 ATOX	159 X<> 00
08 FIX 0	58 XTOA	109 15	160 STOFLAG
09 CF 29	59 APPCHR	110 MOD	161 "OK"
10*LBL 05	60 RDN	111 E	162 CLST
11 "DIM?"	61 ISG X	112 +	163 PROMPT
12 ASTO 01	62 GTO 17	113 X<>F	164 RCLFLAG
13 "F.NAME?"	63 FRC	114 SF 04	165 X<> 00
14 AVIEW	64 E	115 "<"	166 STOFLAG
15 CLA	65 +	116 ARCL 01	167 GTO 18
16 AON	66 VIEW 02	117 "`,`,"	168*LBL 08
17 STOP	67 ISG 02	118 ARCL 02	169 "YOU'RE OUT"
18 AOFF	68 GTO 16	119 ">DIR-"	170 RCL 00
19 SF 25	69 RCL 02	120 FS? 00	171 STOFLAG
20 RCLPTA	70 2	121 "N"	172 CLST
21 FS?C 25	71 /	122 FS? 01	173 BEEP
22 GTO 06	72 INT	123 "E"	174 AVIEW
23*LBL 07	73 STO 01	124 FS? 02	175 END
24 CLST	74 STO 02	125 "S"	
25 VIEW 01	75 GTO 18	126 FS? 03	<u>01*LBL "XMZE1"</u>
26 STOP	76*LBL 06	127 "W"	02 "LOADING..."
27 STO 02	77 CLA	128 AVIEW	03 AVIEW
28 X^2	78 SF 25	129*LBL 19	04 11
29 LASTX	79 POSFL	130 GETKEY	05 "MAZE1"
30 +	80 FC?C 25	131 X=0?	06 SF 25
31 7	81 GTO 05	132 GTO 19	07 PURFL
32 /	82 CLST	133 ")5?T"	08 CF 25
33 E	83 "START? X^Y"	134 POSA	09 CRFLAS
34 +	84 PROMPT	135 X<0?	10 "AGAGAEEG"
35 SF 25	85 STO 02	136 GTO 19	11 APPREC
36 CRFLAS	86 X<>Y	137 FC? IND X	12 "OOOOOAED"
37 FC? 25	87 STO 01	138 GTO 19	13 APPREC
38 GTO 07	88 X<>Y	139 XEQ IND X	14 "KOOMEMEG"
39 TIME	89 E3	140 GTO 18	15 APPREC
40 RNG	90 /	141*LBL 00	16 "ADOADAID"
41 STO 01	91 +	142 DSE 01	17 APPREC
42 E3	92 SF 25	143 RTN	18 "OADOADMG"
43 ST/ 02	93 SEEKPT	144 RTN	19 APPREC
44 E	94 FC? 25	145*LBL 01	20 "OADMIED"
45 ST+ 02	95 GTO 06	146 ISG 02	21 APPREC
46 RCL 02	96*LBL 18	147 RTN	22 "OOOAGOAG"
47*LBL 16	97 RCL 02	148 RTN	23 APPREC
48 "E"	98 E3	149*LBL 02	24 "MDMDMFDO"
49 APPREC	99 /	150 ISG 01	25 APPREC
50 DELREC	100 RCL 01	151 RTN	26 "DONE"
51*LBL 17	101 +	152 RTN	27 AVIEW
	102 SF 25	153*LBL 03	28 END

Step Game

George G. Sandoval; UPL #00363C

You and the HP start at extreme opposite ends of the display, you at the right end (at position 10), indicated by you symbol (#, the "not equals" sign), and the HP on the left, at position 1, indicated by his symbol (>, the "greater than" sign). Haves are 1 to 2 steps forward, or 1 step backward. A forward move is one away from your end of the display; a backward move is one going towards your end of the display.

The object of the game is to force an opponent back to his end of the display in such a way that he cannot move anymore. A player may not move forward if there is only one space or position separating him and his opponent; may not move backward if there is no space behind him to move to. You make the first move, and you and the HP alternate. Remember, a player who cannot move loses the game.

Warning: If you wish to move to position 10, you must key in 10 and not 0, although the display displays' position 10 as a 0.

Note: >23456789# is the initial display configuration

1234567>9# means you lose the game.

>2#4567890 means HP loses the game.

Sample Problems:

1. Sample problem in which you lose.

Input	Function	Display	Comments
	XEQ SIZE 028		
	XEQ "STEP", [A]	LOADING -	Loading Data
		123456789#	Initial display configuration
8	[B]	1234567#90	
	[C]	12315#7890	you have to retreat
7	[B]	123156#890	
	[C]	123416#890	HP gives chase
8	[B]	1234167#90	
	[C]	1234517#90	
9	[B]	12345178#0	
	[C]	12345618#0	
10	[B]	123456189#0	you're trapped
	[C]	123456719#	you lose

2. Sample problem in which you win. (Like in some things in life, it is possible for you to win at this game.)

Input	Function	Display	Comments
	[A]	1 2 3 4 5 6 7 8 9 #	Initial display configuration
9	[B]	1 2 3 4 5 6 7 8 # 0	
	[C]	1 2 1 4 5 6 7 8 # 0	
8	[B]	1 2 1 4 5 6 7 # 9 0	hp is on the run!
	[C]	1 1 3 4 5 6 7 # 9 0	
7	[B]	1 1 3 4 5 6 # 8 9 0	
	[C]	1 2 3 4 5 6 # 8 9 0	HP appears cornered!
6	[B]	1 2 3 4 5 # 7 8 9 0	
	[C]	1 1 3 4 5 # 7 8 9 0	
4	[B]	1 1 3 # 5 6 7 8 9 0	Looks like a fight Last nail in the coffin
	[C]	1 2 3 # 5 6 7 8 9 0	
3	[B]	1 2 # 4 5 6 7 8 9 0	Man beats machine again

Program listing:

01*LBL "STEPS"	30 GTO 01	59 RCL 02
02*LBL a	31*LBL 03	60 6
03 XROM "ST\$"	32 RCL 02	61 X<>Y
04*LBL A	33 RCL 01	62 X=Y?
05 FIX 0	34 -	63 GTO 07
06 ,	35 4	64 XEQ D
07 STO 01	36 X<=Y?	65 FS? 01
08 10	37 GTO 06	66 GTO 11
09 STO 02	38 X<>Y	67 -1
10 E	39 XEQ D	68 GTO 01
11 STO 03	40 E	69*LBL D
12*LBL 07	41 FC? 01	70 2
13 E	42 CHS	71 /
14*LBL 01	43 GTO 01	72 INT
15 RCL 01	44*LBL 06	73 LASTX
16 +	45 RCL 03	74 X#Y?
17 STO 01	46 XEQ D	75 SF 01
18 CF 01	47 FS? 01	76 RTN
19 GTO 10	48 GTO 04	77*LBL 11
20*LBL B	49*LBL 05	78 2
21 STO 02	50 CF 01	79 GTO 01
22 GTO 10	51 RCL 02	80*LBL 10
23*LBL C	52 XEQ D	81 CLA
24 E	53 FC? 01	82 5
25 ST- 03	54 GTO 11	83 RCL 02
26 RCL 03	55 -1	84 X>Y?
27 X#0?	56 GTO 01	85 GTO 64
28 GTO 03	57*LBL 04	86 3
29 2	58 CF 01	87 RCL 01

88 X=Y?	106 GTO 66	124 2
89 GTO 35	107*LBL 60	125 *
90 RCL 01	108 RCL 01	126 RCL 02
91 E	109 5	127 +
92 -	110 +	128 E
93 2	111 ARCL IND X	129 +
94 *	112 RCL 02	130 ARCL IND X
95 RCL 02	113 E1	131 PROMPT
96 +	114 +	132*LBL 35
97 8	115 ARCL IND X	133 ARCL 05
98 +	116 PROMPT	134 ARCL 27
99 ARCL IND X	117*LBL 66	135 PROMPT
100 ARCL 27	118 RCL 01	136*LBL 80
101 PROMPT	119 8	137 ARCL 04
102*LBL 64	120 X=Y?	138 ARCL 26
103 5	121 GTO 80	139 PROMPT
104 RCL 01	122 ARCL 04	140 END
105 X>Y?	123 RCL 01	

01*LBL "ST\$"
 02 "LOADING..."
 03 AVIEW
 04 SIZE?
 05 27
 06 X>Y?
 07 PSIZE
 08 " 12345"
 09 ASTO 04
 10 " 12>4#"
 11 ASTO 05
 12 " >2345"
 13 ASTO 06
 14 " 1>345"
 15 ASTO 07
 16 " 12>45"
 17 ASTO 08
 18 " 123>5"
 19 ASTO 09

20 " 1234>"
 21 ASTO 10
 22 " >2#45"
 23 ASTO 11
 24 " >23#5"
 25 ASTO 12
 26 " >234#"
 27 ASTO 13
 28 " 1>3#5"
 29 ASTO 14
 30 " 1>34#"
 31 ASTO 15
 32 "#7890 "
 33 ASTO 16
 34 "6#890 "
 35 ASTO 17
 36 "67#90 "
 37 ASTO 18
 38 "678#0 "

39 ASTO 19
 40 "6789# "
 41 ASTO 20
 42 ">7#90 "
 43 ASTO 21
 44 ">78#0 "
 45 ASTO 22
 46 ">789# "
 47 ASTO 23
 48 "6>8#0 "
 49 ASTO 24
 50 "6>89# "
 51 ASTO 25
 52 "67>9# "
 53 ASTO 26
 54 "67890 "
 55 ASTO 27
 56 END

Skunk for the HP-41C/CV/CX/42

This program created and uploaded by [Russ Gilbert](#). It is free.

This program is supplied without representation or warranty of any kind. Russ Gilbert and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

Doesn't require CX functions. Requires 77 registers for program, 15 for data (41C with extra memory).

Filename SKNK81.RAW. XEQ 'SKNK'.

The game of Skunk is a two dice game, better known on the HP48 by Doug Cannon. HPGene Wright has a 41 version on his site www.rskey.org/gene/hpgene, it requires the CX functions. I haven't asked Doug if I can use his name, Gene says OK. I am grateful to both.

You roll the dice with the A key (in USER), you stay with the B key. This Skunk has options to start, hit R/S for the default.

1. Target score, default 100. 2. Who is first, default player. 3. Auto Roll (your first roll is automatic.) default Yes. 4. Max die, 7 is the default. You can pick any number for the max die, but 6 to 8 are recommended. You get less 'Skunked' and 'Double' with 7 or 8 than with max die 6. If only one is a '1' then you are 'Skunked' and lose your turn. If two '1's are rolled you get the dreaded 'Double' and your score is set to zero. The winner is the one who gets the highest score over the target score after who goes second. After the game is over, R/S will start over, selecting only max die.

The display is like this:

HP:6,4,10:20 30

The numbers are die 1, die 2, total score this turn:HP
total PL total (whose turn score is first).



Program listing:

```

01 LBL "SKNK"
02 CLA
03 "SEED? 0-.99"
04 AVIEW
05 PROMPT
06 STO 00
07 FIX 00
08 SF 27
09 CF 00
10 CF 02
11 0
12 STO 15
13 ":"
14 ASTO 11
15 ", "
16 ASTO 09
17 " "
18 ASTO 12
19 "GOAL? 100"
20 100
21 PROMPT
22 STO 10
23 1
24 STO 14
25 "N"
26 ASTO Y
27 "PLYR 1ST? Y"
28 AON
29 STOP
30 AOFF
31 ASTO X
32 X=Y?
33 XEQ 17
34 "AUTO ROLL? Y"
35 AON
36 STOP
37 AOFF
38 ASTO X
39 X=Y?
40 XEQ 19
41 LBL 16
42 CLA
43 "MAXDIE? 6-8"
44 7
45 PROMPT
46 STO 16
47 CF 29
48 0
49 STO 01
50 STO 02
51 STO 03
52 STO 04
53 STO 05
54 RCL 15
55 1
56 X=Y?
57 SF 00
58 3
59 STO 13
60 GTO IND 14
61 LBL 17
62 2
63 STO 14
64 SF 02
65 RTN
66 LBL 19
67 SF 00
68 1
69 STO 15
70 RTN
71 LBL 01
72 2
73 STO 06
74 1
75 STO 07
76 3
77 STO 13
78 FS?C 00
79 GTO 18
80 XEQ 09
81 STO 04
82 XEQ 09
83 STO 05
84 +
85 ST+ 03
86 LBL 18
87 CLA
88 "PL:"
89 ARCL 04
90 ARCL 09
91 ARCL 05
92 ARCL 09
93 ARCL 03
94 ARCL 11
95 ARCL 01
96 ARCL 12
97 ARCL 02
98 AVIEW
99 TONE 07
100 XEQ 08
101 XEQ 07
102 STOP
103 LBL 08
104 RCL 04
105 1
106 X#Y?
107 RTN
108 RCL 05
109 X#Y?
110 RTN
111 TONE 03
112 TONE 01
113 RCL 15
114 1
115 X=Y?
116 SF 00
117 0
118 STO IND 07
119 STO 04
120 STO 05
121 STO 03
122 XEQ 06
123 CLA
124 "*** DBL ***"
125 AVIEW
126 GTO IND 06
127 LBL 09
128 RCL 16
129 RCL 00
130 9821
131 *
132 .211327
133 +
134 FRC
135 STO 00
136 *
137 INT
138 1
139 +
140 RTN
141 LBL 07
142 RCL 04
143 1
144 X=Y?
145 XEQ 10
146 RCL 05
147 X=Y?
148 XEQ 10
149 RTN
150 LBL 10
151 CLA
152 "*SKUNKED*"
153 AVIEW
154 XEQ 06
155 RCL 15
156 1
157 X=Y?
158 SF 00
159 0
160 STO 04
161 STO 05
162 STO 03
163 GTO IND 06
164 LBL 02
165 2
166 STO 07
167 1
168 STO 06
169 XEQ 09
170 STO 04
171 XEQ 09
172 STO 05
173 +
174 ST+ 03

```

175 CLA	213 STO 04	251 TONE 05
176 "HP:"	214 STO 05	252 TONE 03
177 ARCL 04	215 STO 03	253 TONE 05
178 ARCL 09	216 GTO 01	254 TONE 04
179 ARCL 05	217 LBL 06	255 TONE 03
180 ARCL 09	218 RCL 01	256 TONE 09
181 ARCL 03	219 RCL 10	257 GTO 13
182 ARCL 11	220 X<=Y?	258 LBL A
183 ARCL 02	221 GTO 14	259 GTO 01
184 ARCL 12	222 RCL 02	260 LBL B
185 ARCL 01	223 RCL 10	261 RCL 03
186 AVIEW	224 X<=Y?	262 ST+ 01
187 TONE 03	225 GTO 14	263 0
188 XEQ 08	226 RTN	264 STO 03
189 XEQ 07	227 LBL 14	265 STO 08
190 RCL 02	228 RCL 01	266 FS? 02
191 RCL 03	229 RCL 02	267 XEQ 06
192 +	230 X=Y?	268 GTO 02
193 STO 08	231 GTO 15	269 LBL 12
194 RCL 10	232 X<Y?	270 CLA
195 X<=Y?	233 GTO 05	271 "HPWIN "
196 GTO 03	234 GTO 12	272 ARCL 02
197 RCL 01	235 LBL 15	273 ARCL 11
198 RCL 10	236 CLA	274 ARCL 01
199 X<=Y?	237 "*TIE*"	275 AVIEW
200 GTO 02	238 ARCL 01	276 TONE 06
201 DSE 13	239 ARCL 11	277 TONE 06
202 GTO 02	240 ARCL 02	278 TONE 03
203 LBL 03	241 BEEP	279 TONE 06
204 RCL 03	242 GTO 13	280 TONE 05
205 ST+ 02	243 LBL 05	281 TONE 03
206 FC? 02	244 CLA	282 LBL 13
207 XEQ 06	245 "PLWIN "	283 SF 29
208 RCL 15	246 ARCL 01	284 PROMPT
209 1	247 ARCL 11	285 GTO 16
210 X=Y?	248 ARCL 02	286 END
211 SF 00	249 AVIEW	
212 0	250 TONE 06	

Game of Skunk –

Gene Wright - <https://www.rskey.org/gene/hpgene/skunk.htm>

Having recently sold my HP-48, I miss a few things. One of them is a game called "SKUNK". So, I did the next best thing: I wrote a version for the HP-41! I based it on what I remembered from the SKUNK game for the HP-48 originally written by Doug Cannon. Thanks to Doug. If you have an HP-48 and haven't played Doug's version, do so. The game and graphics are quite fun.

I've been working on this game for over 15 months, so I've found most (hopefully) of the bugs. Bug reports would be welcome, as I intend to support this game for those interested. As of 5/14/97, it's 422 bytes long (fits on two mag cards) and 222 lines. The game is fairly long so if you don't want to key it in, email me to get my mailing address and I'll send copies of the program either on bar code or two magnetic cards you provide for \$5. (That's not designed to make me rich!) If you type it in and like it, drop me an email! HP-42 owners, of course, will have to key the game in anyway. :-(

Description: SKUNK is a 2 dice game of part strategy, part luck. I am unaware of the author of the original idea for the game, I think the game is ancient.

The game begins by asking you to enter a decimal seed. Then you are asked to enter the goal in points. 100 is the default if no entry is made, and trying to get to 200 is relatively hard. Then you are asked if HP should go first or yourself. Pressing R/S without an entry makes you go first. To make HP go first, type the letter Y. When it is your turn, flag 1 is set. When HP is playing, flag 2 is set. NOTE: Flags 1 and 2 do NOT determine who is player #1 or player #2. They are just a convenient way to indicate whether it's you that's playing or HP. Sorry if that turns out to be a little confusing, but....

At this point you begin play. Player #1 rolls first, and his options are to Roll or Pass. By pressing the A key, the dice are rolled, and by pressing the E key you may pass. The object is to roll as many times as you can without getting "Skunked". One is "Skunked" when a one is rolled on either die. If no ones are rolled, then the sum of the two dice is added to the pot. If you are "Skunked" then you lose your turn and the pot goes to zero. If you choose to pass at some time, then the current pot is added to your score. The pot then returns to zero, and the play passes to the next player.

It is extremely disastrous to be "Double Skunked". This is, of course, when a one is rolled on each die. At this point, the pot goes to zero, your score goes to zero, and you lose your turn. This is most undesirable.

The status of the game is shown in the display as: **0,0: PP YY HP** or: **0,0: PP HP YY**

depending on if it's your turn or HP's, and where PP is the point total in the "Pot", YY is your point total (assuming it's your turn), and HP is the HP's point total.

Now, it is only fair that everyone get the same amount of turns, so if player #1 should reach or pass the goal score, then player #2 has one more turn to pass the score of player #1. Thus, if both players reach the destination score, the winner is the player with the highest score. You can see the advantages to being player #2. If player #2 reaches the destination score, and player #1 has not yet reached it, player #2 wins immediately.

If both players reach the destination score and are tied, then HP wins. This game is actually fairly tough to beat! If you can beat this game more than 6 times out of 10, congratulations! Game play hinges on your willingness to risk the points in the pot to roll again. Enjoy! Here's a sample game:

See: _____ Press: (Comments in brackets)

SEETP	XEQ "SKK" (Remember to clear flag 26 for a quiet game)
GORTP	0.987654321 R/S
HP 15TP	R/S (We'll play to 100, the default)
	N R/S (We'll go first and let HP have the final turn)
00:0 0 0	[A] (Press A or the Sigma+ key (HP-41) or XEQ A (HP-42)
5:27 0 0	[A] (A total of 7 points is in the pot)
16:14 0 0	
SKUNK	(A 1 showed up, so I got skunked. It's HP's turn now)
3:6:9 0 0	
16:16 0 0	
SKUNK	(HP got skunked on it's second roll. My turn again)
00:0 0 0 [A]	[A]
3:4:7 0 0	[E] (I'll hold onto the 19 points, so I'll pass to the HP)
6:6:19 0 0	
3:3:6 0 19	
16:13 0 19	(HP got skunked again. Notice my 19 points got moved)
SKUNK	
00:0 19 0	[A]
4:2:6 19 0	[A]
16:13 19 0	(I got skunked again)
SKUNK	
2:6:8 0 19	
2:11:1 0 19	(HP got skunked again)
SKUNK	
00:0 19 0	[A]
6:4:10 19 0	[A]
3:3:16 19 0	[A]
2:2:20 19 0	[A]
6:4:30 19 0	[E] (I'll keep the 30 add'l points)
2:2:4 0 49	
6:11:1 0 49	(HP continues to get skunked. Maybe I'll win 100 to 0?)
SKUNK	
00:0 49 0	[A]
3:6:9 49 0	[A]
3:4:16 49 0	[E] (I'll keep the 16 add'l points)
4:1:5 0 65	
SKUNK	(HP sure is getting skunked alot)

```

00:0 65 0 [A]
4:26 65 0 [A]
4:3:13 65 0 [E] (I'll keep the 13 points)

3:47 0 78
5:113 0 78
SKUNK (Skunked again! Will HP ever score?)

00:0 78 0 [A]
1:56 78 0 (Now I got skunked)
SKUNK

6:17 0 78
SKUNK (HP is skunked again)

00:0 78 0 [A]
3:25 78 0 [E] (I want HP to score sometime!)

5:16 0 83
SKUNK (Oh well)

00:0 83 0 [A]
1:34 83 0 (Depending on starting seed, you can have lots of skunks!)
SKUNK

5:49 0 83
4:5:18 0 83
5:3:26 0 83 (How about that! HP scored!)

00:0 83 26 [A] (I'll go for the kill now!)
5:6:11 83 26 [A]
6:5:22 83 26 [E] (I'll be over 100 points now, so HP has to beat me!)

3:58 26 83
1:23 26 83 (Oh well)
SKUNK
YOU WIN
Y:105 HP:26

```

Program listing:

A few lines of the program below might need some explanation. The symbols "->" means APPEND alpha characters. When you see X NE Y? that is X is not equal to Y? RDN is roll down. Line 80 is append 1 space. So is line 86.

X-functions are used in only a few places. The X<>F instruction is used to set/clear flags in lines 5, 32, and 153. ATOX is used in line 22. If someone wants a version that doesn't use these, email me. It doesn't take much extra programming to mimic what these would do, if you have a vintage HP-41C, for example. Otherwise, use the X-functions. ;-)

Technical specs: This game requires Size of 008 and uses flags 0, 1, 2, 6, 7, and 10. Memory 0 holds the random number seed. Memories 1 and 2 hold the two dice. Memory 3 holds the

sum of the two dice. Memory 4 holds the human's score while memory 5 holds the HP's score. Memory 6 holds the goal point total being played to. Memory 7 is a counter used to determine HP's move (that I could probably get rid of if I did a proper analysis of the stack! Well, there's only so many hours in the day!)

01	LBL "SKK"	48	GTO 10	95	TONE 0
02	<u>LBL 07</u>	49	<u>LBL 09</u>	96	FS?C 00
03	SF 27	50	XEQ 14	97	GTO 90
04	CLX	51	FS? 10	98	CLX
05	X<>F	52	GTO 20	99	FS? 02
06	CF 10	53	RCL 06	100	STO 05
07	FIX 0	54	RCL 05	101	FC?C 01
08	CF 29	55	X < Y?	102	GTO 99
09	"SEED?"	56	GTO 20	103	STO 04
10	PROMPT	57	RCL 04	104	GTO 20
11	CLRG	58	X > Y?	105	LBL E
12	STO 00	59	GTO 90	106	RCL 03
13	"GOAL?"	60	<u>LBL 98</u>	107	ST+ 04
14	2	61	"TOO BAD"	108	RCL 04
15	10^X	62	AVIEW	109	RCL 06
16	PROMPT	63	TONE 3	110	X < = Y?
17	STO 06	64	TONE 1	111	SF 00
18	"HP 1ST?"	65	<u>LBL 08</u>	112	FS? 10
19	AON	66	PSE	113	GTO 20
20	STOP	67	"Y:"	114	FC? 00
21	AOFF	68	ARCL 04	115	GTO 20
22	ATOX	69	>" HP:"	<u>116</u>	<u>LBL 90</u>
23	89	70	ARCL 05	117	"YOU WIN"
24	X=Y?	71	PROMPT	118	AVIEW
25	GTO 20	72	GTO 07	119	BEEP
26	SF 10	73	LBL C	120	GTO 08
27	<u>LBL 99</u>	74	CLA	<u>121</u>	<u>LBL 00</u>
28	CLX	75	ARCL 01	122	XEQ 00
29	STO 01	76	>" ,"	123	STO 01
30	STO 02	77	ARCL 02	124	XEQ 00
31	STO 03	78	>" :"	125	STO 02
32	X<>F	79	ARCL 03	126	1
33	SF 01	80	>" "	127	X=Y?
34	XEQ C	81	RCL 04	128	SF 07
35	TONE 9	82	RCL 05	129	RCL 01
36	STOP	83	FS? 01	130	X=Y?
37	LBL A	84	X<>Y	131	SF 06
38	XEQ 00	85	ARCL X	132	RCL 02
39	TONE 9	86	>" "	133	+
40	FS? 06	87	ARCL Y	134	ST+ 03
41	GTO 13	88	AVIEW	135	GTO C
42	FS? 07	89	RTN	<u>136</u>	<u>LBL 00</u>
43	GTO 09	<u>90</u>	<u>LBL 10</u>	137	RCL 00
44	RTN	91	PSE	138	997
45	GTO A	92	"DOUBLE SKUNK"	139	*
<u>46</u>	<u>LBL 13</u>	93	AVIEW	140	FRC
47	FS? 07	94	TONE 0	141	STO 00

142	6	169	X<=Y?	196	RDN
143	*	170	GTO 05	197	STO 05
144	1	171	FS? 00	198	FC? 10
145	+	172	GTO 06	199	GTO 99
146	INT	173	RCL 07	200	RCL 04
147	RTN	174	3	201	X < = Y?
148	<u>LBL 20</u>	175	X NE Y?	202	GTO 98
149	CLX	176	GTO 02	203	RCL 03
150	STO 03	177	RCL 06	204	ST- 05
151	FS? 00	178	RCL 04	205	GTO 06
152	1	179	-	206	<u>LBL 11</u>
153	X <> F	180	10	207	FS? 07
154	SF 02	181	X>Y?	208	GTO 10
155	3	182	GTO 02	209	<u>LBL 04</u>
156	STO 07	183	RCL 03	210	XEQ 14
157	<u>LBL 06</u>	184	X > Y?	211	FC? 10
158	XEQ 00	185	GTO 03	212	GTO 99
159	FS? 06	186	<u>LBL 02</u>	213	FS? 00
160	GTO 11	187	DSE 07	214	GTO 90
161	FS? 07	188	GTO 06	215	GTO 99
162	GTO 04	189	<u>LBL 03</u>	216	<u>LBL 14</u>
163	TONE 7	190	FS? 00	217	PSE
164	PSE	191	GTO 06	218	"SKUNK"
165	RCL 03	192	RCL 03	219	AVIEW
166	RCL 05	193	ST+ 05	220	TONE 7
167	+	194	GTO 99	221	TONE 0
168	RCL 06	195	<u>LBL 05</u>	222	END

Yahtzee.

[Gene Wright, PPCCJ V12N5 p39 ; \(May 1985\)](#)

The program YZ will play and score the game of Yahtzee. For the complete rules of Yahtzee, your best bet is to read the rules on the box at a store. ;-) It will run on the HP-41CV or HP-42S and needs no plug-in modules and no X-functions. (It will run on an HP-41C with extra memory modules or a Quad Memory Module). I've done my best to debug this, but bug reports are welcome. The program listing is presented below, but if you'd rather not type it, I can provide it on magnetic cards or on bar code for \$5. Email to get my mailing address.

Requirements: Size 024 and 656 bytes of program memory. Note: It will beep unless you clear flag 26 first. Wouldn't want anyone to get caught at work playing it.

Here's a short description. Yahtzee is somewhat like a poker game with dice. You get 13 "hands" of 5 dice to score into rows of a scorepad that are specifically for a certain combination of dice. The row numbers presented below are needed during the game to tell the HP where to score the current roll of dice. Keep these handy!

Rows 1-6 are for scoring as many 1's and 6's as you can.

Rows 7 and 8 are for three and four of a kind, respectively. Row 9 is for a full house (Like 1,1,4,4,4 or 3,3,6,6,6, etc.)

Row 10 is for a straight of 4 in a row (Like 1,2,3,4 or 3,4,5,6, etc.) Row 11 is for a straight of five in a row (Like 1,2,3,4,5 or 2,3,4,5,6) Row 12 is for 5 of a kind or a Yahtzee.

Row 13 is called "Chance", in case you need a spare chance.

To help you get the dice you need, you are given three "rolls" to make the best you can out of the 5 dice. When they are first rolled, pick the ones you want to roll again. When they are re-rolled, you get to pick some to roll the second time. However, once that is done, you must choose a row to score the 5 dice in.

You get a bonus of 35 points if you score 63 or better in the top 6 rows. To get 63, you must average three 1's, 2's, etc.

Rows 7 and 8 will score the total of the dice that are showing. Row 9 scores 25, Row 10 scores 30, and Row 11, scores 40. The Yahtzee (Row 12) scores 50. Chance scores like rows 7 and 8, providing the total of the displayed dice.

It is possible to get two (or more) Yahtzees in a game. The first should be scored in row 12, but to get the 100 point bonus from numbers 2, etc., you must be able to score it in a row as usual. For example, if you get a second Yahtzee of 2's, you can score it for a bonus in Row 2, Row 7, Row 8, Row 9, or Row 13. To indicate to the program to check for this bonus, enter the Row to be used as a NEGATIVE Number! Note: You can't score an extra Yahtzee in Rows 10 or 11.

To play, make sure you have a Size of 024, XEQ YZ, and enter a decimal seed. The display will show the 5 dice you rolled in sorted order with the prompt "ROLL?" at the end of the display. Pick the dice you wish to roll again and enter their position numbers, and press R/S. You can do this twice before having to score the roll. If you don't want to roll any of the dice over, simply press R/S to go straight to scoring.

For example, if the display is showing 22256 and you want to try for 2's, you would press 45 R/S. But, if on the first roll you got 12345, you might want to just score it as a large straight (if not already used) by pressing R/S without entering anything.

After your second re-roll, you are shown the dice and prompted with SCORE? Enter the row you wish to score the roll into and press R/S. If you enter a row that has already been used, you get the SCORE prompt again. If you are not sure which rows remain to be used, press R/S without entering anything and you will be shown a list of the unused rows. The scoring routines will detect when no points should be given and will assign a zero to the row if needed.

Programming notes: The program uses flags 0-20. Printed output may thus be affected. The program, as written will fit on three magnetic cards (if anyone still uses them). The program uses label numbers 1-13 for the corresponding rows on the Yahtzee scorecard. I've spent a lot of time trying to optimize how to determine if the dice in fact satisfy the requirements of each row. To me, when told to score the dice in Row 10, detecting a small straight or giving a zero if one isn't present was the hardest one to do efficiently. I will, of course, be glad to hear of suggestions for programming improvements.

You can save some bytes if you always have a PPC ROM plugged in by changing Label 40 to be LBL 40, 14.018, XROM S2, RTN. (If you have a CCD module, replace the XROM S2 with SORT). This will save about 55 bytes. The current sort routine at label 40 uses a mode change from Degrees to Radians to indicate that a number swap has occurred. (It's kind a fun to watch the RAD switch on and off.) The game is somewhat slow the first time through, but speeds up later.

What's a good score? Anything over 250 is good, but with extra Yahtzee's, you can get up over 450 or higher. Trivia: What's the lowest possible score?
Here is a complete sample game.

The column for "See:" indicates what is shown in the calculator display. The column for "Press:" indicates what you must type in. Remember: unless you clear flag 26, the program may BEEP at you during play!

See:	Press:
5EE07	XEQ YZ
11246 ROLL?	0.123456789, R/S
12445 ROLL?	15, R/S
12234 SCORE?	45, R/S
ROW 10=30	10, R/S
25666 ROLL?	12, R/S
46666 ROLL?	1, R/S
66666 SCORE?	12, R/S
ROW 12=50	(Hear a beep)
14556 ROLL?	125, R/S
13355 ROLL?	123, R/S
12455 SCORE?	1, R/S
ROW 1=1	
22366 ROLL?	345, R/S
22333 ROLL?	0, R/S
22333 SCORE?	9, R/S
ROW 9=25	
24446 ROLL?	15, R/S

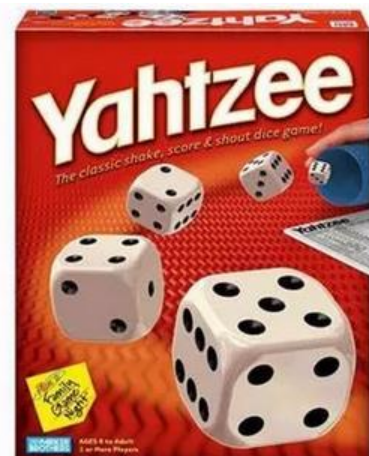
```

24445 ROLL?      15, R/S
11444 SCORE?     4, R/S
ROW 4=12
13346 ROLL?      145 R/S
13336 ROLL?      15 R/S
13333 SCORE?     3, R/S
ROW 3=12
22266 ROLL?      45, R/S
22222 ROLL?      0, R/S
22222 SCORE?     -2, R/S (To get the yahtzee bonus, it's a negative two) ROW
2=10 +100 (This is because it's our second yahtzee of the
game)
12236 ROLL?      145, R/S
22236 ROLL?      45, R/S
11222 SCORE?     7, R/S
ROW 7=8
11556 ROLL?      1234, R/S
44666 ROLL?      12, R/S
46666 SCORE?     6, R/S
ROW 6=24
24446 ROLL?      15, R/S
24446 ROLL?      15, R/S
23444 SCORE?     13, R/S
ROW 13=17
12466 ROLL?      15, R/S
23466 ROLL?      4, R/S
23456 SCORE?     11, R/S
ROW 11=40
12345 ROLL?      1234, R/S
13566 ROLL?      1245, R/S
14555 SCORE?     0, R/S
5,8,
14555 SCORE?     5, R/S
ROW 5=15
13466 ROLL?      123, R/S
25666 ROLL?      12, R/S
34666 SCORE?     8, R/S
ROW 8=0
DONE
TOP=74+35
LOWER=270
END=379

```

Not too bad. Press R/S for another game. Note: If you choose not to play another game and leave YZ to do something else, the flags are left set in end of game condition.

Program Listing Notes: Line 134 is a plus sign in alpha - not an append plus. The symbol for append used in the listing is ">".



Program listing:

01	LBL "YZ"	50	X>Y?	98	CF 20	147	GTO 75
02	CLRG	51	GTO 72	99	PROMPT	148	"DONE"
03	"SEED?"	52	X<>Y	100	INT	149	AVIEW
04	PROMPT	53	RDN	101	13	150	RCL 01
05	STO 00	54	13	102	X<>Y	151	RCL 02
06	LBL 16	55	+	103	X>Y?	152	RCL 03
07	,02	56	RCL 00	104	GTO 66	153	RCL 04
08	LBL 14	57	997	105	X=0?	154	+
09	CF IND X	58	*	106	GTO 25	155	+
10	ISG X	59	FRC	107	X<0?	156	+
11	GTO 14	60	STO 00	108	GTO 65	157	RCL 05
12	14.018	61	6	109	FS? IND X	158	+
13	STO 23	62	*	110	GTO 66	159	RCL 06
14	FIX 0	63	1	111	LBL 64	160	+
15	CF 29	64	+	112	STO 21	161	"TOP ="
16	LBL 15	65	INT	113	RCL 23	162	ARCL X
17	12345	66	STO IND Y	114	STO 22	163	>"+"
18	GTO 71	67	LBL 72	115	CLX	164	63
19	LBL 17	68	RCL Z	116	STO 19	165	X<=Y?
20	1	69	GTO 71	117	STO 20	166	35
21	ST+ 19	70	LBL 20	118	XEQ IND 21	167	X>Y?
22	RCL 19	71	XEQ 40	119	CF 00	168	0
23	3	72	LBL 61	120	"ROW "	169	ARCL X
24	X<=Y?	73	CLA	121	ARCL 21	170	X<>Y
25	GTO 22	74	ARCL 14	122	>"="	171	RDN
26	>" ROLL?"	75	ARCL 15	123	ARCL 20	172	+
27	LBL 27	76	ARCL 16	124	AVIEW	173	AVIEW
28	CLX	77	ARCL 17	125	RCL 20	174	PSE
29	PROMPT	78	ARCL 18	126	STO IND 21	175	RCL 07
30	INT	79	FS?C 00	127	SF IND 21	176	RCL 08
31	X<0?	80	GTO 22	128	X#0?	177	RCL 09
32	GTO 27	81	GTO 17	129	FC?C 20	178	+
33	X=0?	82	LBL 65	130	GTO 63	179	+
34	SF 00	83	ABS	131	2	180	RCL 10
35	X=0?	84	RCL 12	132	10^X	181	+
36	GTO 61	85	X=0?	133	ST+ 12	182	RCL 11
37	LBL 71	86	GTO 61	134	"+"	183	+
38	10	87	RDN	135	ARCL X	184	RCL 12
39	/	88	RCL 14	136	PSE	185	+
40	ENTER	89	RCL 18	137	AVIEW	186	RCL 13
41	INT	90	X=Y?	138	PSE	187	+
42	X<>Y	91	SF 20	139	LBL 63	188	"LOWER="
43	FRC	92	RCL Z	140	CLX	189	ARCL X
44	10	93	GTO 64	141	STO 20	190	AVIEW
45	*	94	LBL 22	142	1.013	191	PSE
46	X=0?	95	>"	143	LBL 75	192	+
47	GTO 20	96	SCORE?"	144	FC? IND X	193	"END="
48	5	97	LBL 66	145	GTO 15	194	ARCL X
49	X<>Y		CLX	146	ISG X	195	PROMPT

196	GTO 16	<u>246</u>	<u>LBL 07</u>	296	RTN	346	RCL 18
<u>197</u>	<u>LBL 25</u>	247	RCL 14	297	RDN	347	RCL 17
198	CLA	248	RCL 16	298	18	348	-
199	1.013	249	X=Y?	299	X<=Y?	349	1
<u>200</u>	<u>LBL 00</u>	250	GTO 13	300	GTO 36	350	X=Y?
201	FC? IND X	251	RCL 18	301	RDN	351	GTO 41
202	ARCL X	252	X=Y?	302	GTO 35	352	30
203	FC? IND X	253	GTO 13	<u>303</u>	<u>LBL 36</u>	353	STO 20
204	>" ,"	254	RCL 17	304	40	354	RTN
205	ISG X	255	RCL 15	305	FS?C 00	<u>355</u>	<u>LBL 41</u>
206	GTO 00	256	X=Y?	306	30	356	1
207	AVIEW	257	GTO 13	307	STO 20	357	SF 00
208	PSE	258	RTN	308	RTN	358	GTO 11
209	GTO 61	<u>259</u>	<u>LBL 08</u>	<u>309</u>	<u>LBL 10</u>	<u>359</u>	<u>LBL 40</u>
<u>210</u>	<u>LBL 01</u>	260	RCL 14	310	RCL 15	360	DEG
211	LBL 02	261	RCL 17	311	RCL 14	361	RCL 17
212	LBL 03	262	X=Y?	312	-	362	RCL 16
213	LBL 04	263	GTO 13	313	1	363	RCL 15
214	LBL 05	264	RCL 15	314	X NE Y?	364	RCL 14
215	LBL 06	265	RCL 18	315	CLX	365	X>Y?
216	RCL IND 22	266	X=Y?	316	X=0?	366	RAD
217	RCL 21	267	GTO 13	317	STO 14	367	X>Y?
218	X=Y?	268	RTN	318	RCL 17	368	X<>Y
219	ST+ 20	269	LBL 09	319	RCL 16	369	STO 14
220	ISG 22	270	SF 14	320	RCL 15	370	RDN
221	GTO 06	271	XEQ 07	321	RCL 14	371	X>Y?
222	RTN	272	FS?C 14	322	X=Y?	372	RAD
<u>223</u>	<u>LBL 13</u>	273	RTN	323	CLX	373	X>Y?
224	FS?C 14	274	RCL 14	324	STO 14	374	X<>Y
225	RTN	275	RCL 15	325	RDN	375	STO 15
226	RCL 14	276	X#Y?	326	X=Y?	376	RDN
227	RCL 15	277	RTN	327	CLX	377	X>Y?
228	RCL 16	278	RCL 17	328	STO 15	378	RAD
229	RCL 17	279	RCL 18	329	RDN	379	X>Y?
230	+	280	X#Y?	330	X=Y?	380	X<>Y
231	+	281	RTN	331	CLX	381	STO 16
232	+	282	25	332	STO 16	382	RDN
233	RCL 18	283	STO 20	333	RDN	383	RCL 18
234	+	284	RTN	334	RCL 18	384	X<Y?
235	STO 20	<u>285</u>	<u>LBL 11</u>	335	X<>Y	385	RAD
236	RTN	286	RCL 23	336	X=Y?	386	X<Y?
<u>237</u>	<u>LBL 12</u>	287	+	337	CLX	387	X<>Y
238	RCL 18	<u>288</u>	<u>LBL 35</u>	338	STO 17	388	STO 18
239	RCL 14	289	RCL IND X	339	XEQ 40	389	X<>Y
240	X#Y?	290	ISG Y	340	RCL 15	390	STO 17
241	RTN	291	RCL IND Y	341	X=0?	391	FS? 43
242	BEEP	292	1	342	RTN	392	GTO 40
243	50	293	-	343	RCL 14	393	END
244	STO 20	294	-	344	X=0?		
245	RTN	295	X NE 0?	345	GTO 41		

Yams for the HP-41CX –

JM Baillard-<http://hp41programs.yolasite.com/yams.php>

Overview

This program allows you to play "Yams" with - but not against - your HP-41CX.
"Yams" is a kind of Yahtzee game: "Yams" = 5 of a kind.

You roll 5 dice, then, if need be, you can roll again some of these dice - at most 2 times at each turn - and you try to get one of the 13 combinations below. Then, you choose one of these combinations, which cannot be chosen again in the next rounds.

If - for instance - you get several times 2 2 2 2 2, it may be registered as yams, full, 4 of a kind, 3 of a kind ... After 13 turns, the HP-41 displays your score, and the object is to get the highest score.

COMBINATION	SCORING
3 of a kind	Sum of all dice + 10 points
4 of a kind	Sum of all dice + 20 points
Full-house	Sum of all dice + 30 points
Straight	Sum of all dice + 40 points
5 of a kind = yams	Sum of all dice + 50 points
"+" combination	Sum of all dice S
any combination	if you press "+" with 4 3 6 5 1 (S = 19)
"-" combination	Sum of all dice S' provided S > S' and
then "-" with 4 4 6 5 1 (S' = 20),	you will receive 19 pts but not 20 pts.
The ones	Sum of the ones
The twos	Sum of the twos
The threes	Sum of the threes
The fours	Sum of the fours
The fives	Sum of the fives
The sixes	Sum of the sixes

If the sum of these last 6 sums reaches 60 points or more, you get a bonus of 30 points.

There are many variants and the following program may be modified according to your preferences.

Instructions of Use:

0.- Initializewith a random seed in register R00

1.- XEQ "YAMS" the HP-41 displays the 5 dice followed by + - BCFYS All the flag indicators: user 0 1 2 3 4 are set

If you want to throw again the dice - say n° 1 2 4 (as seen from the left) press 1, 2, 4, ENTER^ . Then if you want to roll again the dice 1 4 , press 1, 4 ENTER^

The screen is displayed twice, then choose a combination (if you have a "yams" , press [Y]. The corresponding indicator (flag indicator or a character in the alpha register) will have disappeared on the next round.

2.- Continue to throw again the dice if need be until the 13 rounds are over. The Hp-41 will display your score, preceded by "BONUS", if you've got at least 60 points with the ones, the twos, , the sixes.

Warning:

If you get a satisfying combination before the 3rd attempt - i-e after rolling the dice only 1 or 2 times -press R/S (the display will blink) and only then, press the key corresponding to your combination.

You have 41 seconds to decide which dice you want to throw, but only 10 seconds to decide which combination to choose.

Program Listing

We need indicators to show what combinations are still available:

For the ones I've chosen flag F01, for the twos F02, for the threes F03, for the fours F04. Unfortunately, the state of flags F05 & F06 doesn't appear in the display so, I've used F00 for the fives and F27 (which is the "user mode" flag) for the sixes.

The seven other combinations appear as + - B C F Y S for "+" "-" "3 of a kind" "4 of a kind" "Full-house" "Yams" "Straight" respectively.

-I've chosen B because "3 of a kind" = Brelan in French and C because "4 of a kind" = Carré in French. Other choices are clearly possible...

Data Registers: R00 = random seed

R01 = dice n°1	R06 = Full	R11 = 3 of a kind	R16 = the fours
R02 = dice n°2	R07 = Straight	R12 = 4 of a kind	R17 = the fives
R03 = dice n°3	R08 = +	R13 = the ones	R18 = the sixes
R04 = dice n°4	R09 = -	R14 = the twos	R19 = 13 , 12 , , 1 , 0
R05 = dice n°5	R10 = Yams	R15 = the threes	R20 = 3 , 2 , 1 , 0

Flags:	F01 = the 1's	F04 = the 4's
	F02 = the 2's	F00 = the 5's
	F03 = the 3's	F27 = the 6's

Program listing:

01 LBL "YAMS"	56 X=0?	111 LBL 63	166 RTN
02 FIX 0	57 "~B"	112 5	167 RCL 03
03 CF 29	58 RCL 12	113 XEQ 06	168 X=Y?
04 SF 00	59 X=0?	114 STO 17	169 RTN
05 SF 01	60 "~C"	115 CF 00	170 RCL 04
06 SF 02	61 RCL 06	116 RTN	171 X=Y?
07 SF 03	62 X=0?	117 LBL 64	172 RTN
08 SF 04	63 "~F"	118 6	173 RCL 05
09 SF 27	64 RCL 10	119 XEQ 06	174 X=Y?
10 6.018	65 X=0?	120 STO 18	175 RTN
11 CLRGX	66 "~Y"	121 CF 27	176 RCL 01
12 13	67 RCL 07	122 RTN	177 -
13 STO 19	68 X=0?	123 LBL 06	178 4
14 LBL 01	69 "~S"	124 5	179 X#Y?
15 CLX	70 AVIEW	125 SIGN	180 RTN
16 STO 01	71 DSE 20	126 CLX	181 XEQ 08
17 STO 02	72 FS? 30	127 X<>Y	182 40
18 STO 03	73 GTO 10	128 LBL 07	183 +
19 STO 04	74 LBL 04	129 RCL IND L	184 STO 07
20 STO 05	75 41	130 X=Y?	185 RTN
21 3	76 GETKEYX	131 ST+ Z	186 LBL 51
22 STO 20	77 X<> L	132 RDN	187 XEQ 08
23 LBL 02	78 X=Y?	133 DSE L	188 STO 09
24 5	79 GTO 02	134 GTO 07	189 RCL 08
25 LBL 03	80 X<> L	135 X<>Y	190 X#0?
26 RCL IND X	81 X=0?	136 RTN	191 X>Y?
27 X#0?	82 GTO 10	137 LBL 21	192 RTN
28 GTO 03	83 48	138 CHS	193 CHS
29 CLX	84 -	139 STO 06	194 STO 09
30 RCL 00	85 CLRGX	140 RCL 01	195 RTN
31 R-D	86 GTO 04	141 RCL 02	196 LBL 61
32 6	87 LBL 72	142 X#Y?	197 XEQ 08
33 MOD	88 1	143 RTN	198 STO 08
34 STO 00	89 XEQ 06	144 RCL 04	199 RCL 09
35 INT	90 STO 13	145 RCL 05	200 X<Y?
36 1	91 CF 01	146 X#Y?	201 RTN
37 +	92 RTN	147 RTN	202 CHS
38 STO IND Y	93 LBL 73	148 RCL 03	203 STO 08
39 LBL 03	94 2	149 X=Y?	204 RTN
40 X<>Y	95 XEQ 06	150 GTO 05	205 LBL 71
41 DSE X	96 STO 14	151 RCL 02	206 XEQ 08
42 GTO 03	97 CF 02	152 X#Y?	207 50
43 CLA	98 RTN	153 RTN	208 +
44 ARCL 01	99 LBL 74	154 LBL 05	209 STO 10
45 ARCL 02	100 3	155 XEQ 08	210 RCL 05
46 ARCL 03	101 XEQ 06	156 30	211 RCL 01
47 ARCL 04	102 STO 15	157 +	212 X=Y?
48 ARCL 05	103 CF 03	158 STO 06	213 RTN
49 RCL 08	104 RTN	159 RTN	214 CHS
50 X=0?	105 LBL 62	160 LBL 53	215 STO 10
51 "~+"	106 4	161 CHS	216 RTN
52 RCL 09	107 XEQ 06	162 STO 07	217 LBL 12
53 X=0?	108 STO 16	163 RCL 01	218 CHS
54 "~-"	109 CF 04	164 RCL 02	219 STO 11
55 RCL 11	110 RTN	165 X=Y?	220 RCL 01

221 RCL 03	252 STO 12	283 X>Y?	314 +
222 X=Y?	253 RTN	284 X<>Y	315 RCL 17
223 GTO 12	254 LBL 08	285 RDN	316 +
224 RCL 05	255 RCL 01	286 X>Y?	317 RCL 18
225 X=Y?	256 RCL 02	287 X<>Y	318 +
226 GTO 12	257 +	288 RDN	319 60
227 RCL 02	258 RCL 03	289 X<Y?	320 -
228 RCL 04	259 +	290 X<>Y	321 X<0?
229 X#Y?	260 RCL 04	291 STO 05	322 GTO 11
230 RTN	261 +	292 RDN	323 " BONUS"
231 LBL 12	262 RCL 05	293 X>Y?	324 AVIEW
232 XEQ 08	263 +	294 X<>Y	325 30
233 10	264 RTN	295 STO 02	326 +
234 +	265 LBL 10	296 RDN	327 LBL 11
235 STO 11	266 RCL 04	297 X>Y?	328 RCL IND Y
236 RTN	267 RCL 05	298 X<>Y	329 X<0?
237 LBL 13	268 X>Y?	299 STO 03	330 CLX
238 CHS	269 X<>Y	300 X<>Y	331 +
239 STO 12	270 RCL 03	301 STO 04	332 ISG Y
240 RCL 01	271 X>Y?	302 AVIEW	333 GTO 11
241 RCL 04	272 X<>Y	303 GETKEY	334 60
242 X=Y?	273 RCL 02	304 XEQ IND X	335 +
243 GTO 13	274 X<Y?	305 DSE 19	336 " "
244 RCL 02	275 X<>Y	306 GTO 01	337 ARCL X
245 RCL 05	276 X<> 01	307 6.012	338 "~ PTS"
246 X#Y?	277 X>Y?	308 RCL 13	339 AVIEW
247 RTN	278 X<>Y	309 RCL 14	340 FIX 4
248 LBL 13	279 X<> 01	310 +	341 SF 29
249 XEQ 08	280 X>Y?	311 RCL 15	342 END
250 20	281 X<>Y	312 +	
251 +	282 R^	313 RCL 16	

Othello (Reversi)

Valentín Albillo – PPCCJ V8N3 p14

Ed's note: As it turns out, this program was also available in the HP Users' Library – and that's the versión included here. The print copy was very clean and tidy, and it's much easier to copy-paste than to retype all the documentation from the scratch - or using OCR that frequently fails.

00903C PROGRAM DESCRIPTION I

Page 1 of 12

Program Title REVERSI

Contributor's Name Valentin Albillo

Address Padre Rubio, 61 - 2⁰C

City Madrid, 29

State/Country Spain

Zip Code

Program Description, Equations, Variables This program allows the user to play a game of Reversi against an HP-41C.

YOU PLAY 57

FLIP 2 PCES

```

      1 2 3 4 5 6 7 8
1  - - - - - - - -
2  - - - - - - - -
3  O O O - - - - -
4  - - O O O - - -
5  - - O X X X X -
6  - - O O O X X O
7  - - - - O X X -
8  - - - - - O X O

```

The present program includes all features required: plays quite well and will easily defeat a beginner, so it provides a challenging level for everyone. The program itself runs the same with or without a printer, but if one is present, it will print the board.

The program is also autonomous: no data cards required, no card reader required.

It is also quite fast for such a complex game: the HP-41C performs some 30 moves (whole game) in 25 minutes. Besides, the running speed increases as the game goes on.

You can select who makes the first move, and the type of opening: either diagonal or parallel. Also, you may select to print the board after every new position, or only after HP moves (so saving paper and time). The machine recognizes and rejects illegal moves. Can play a single move for you against itself. Even a whole game against itself if you want (imagine, the HP-41C playing both black and white at the same time!)

Though you are supposed to know the rules of the game, a brief explanation will be given, for the sake of completeness. Here is a brief outline of the rules:

Necessary Accessories 3 single-density memory modules (or a quad module).

Operating Limits and Warnings Your move must be of the form xy, with both x and y ranging from 1 to 8, limits included, and the two exceptions to this rule being 0 (no move) and -1 (HP plays for you). Any negative number may be used instead of -1, if desired. The game generally ends when the board is full of pieces, but it may also end if no player can make a legal move. In that unlikely case, the counting of the pieces is not automatically performed. You must do it by yourself.

References New Mathematical Diversions, by Martin Gardner. Includes the rules of Reversi, and some other curiosities. You can also have a look at the Games Pac for the HP-85 computer, which includes a program to play Reversi (not related to this program in any way, to be sure!!!)

Reversi is played on an 8×8 board. There are two standard openings (see illustrations):

- diagonal opening (left)
- parallel opening (right)

One of the players plays the white pieces (represented by the 0), the other the black ones (represented by the checkerboard character).															
1	2	3	4	5	6	7	8								
1	-	-	-	-	-	-	-								
2	-	-	-	-	-	-	-								
3	-	-	-	-	-	-	-								
4	-	-	-	0	-	-	-								
5	-	-	-	0	0	-	-								
6	-	-	-	-	-	-	-								
7	-	-	-	-	-	-	-								
8	-	-	-	-	-	-	-								

To make a move, the player places one of his pieces in an empty location (represented by a dash) taking into account that:

- it must be adjacent to a piece of the other player.
- at least one enemy piece must be enclosed between the just placed piece and another piece of the same color.

This is, any number of pieces enclosed between the played piece and any other of the same color are flipped: they become of the capturer's color. No empty locations can be enclosed, only full rows of enemy pieces can be flipped. The row can be placed in any direction: horizontal, vertical or diagonal. If more than one row is enclosed at the same time, all are flipped. You can capture only when putting a piece on the board: enemy pieces which are left enclosed by yours because of other factors are not captured, of course.

Some example should make it clear. Look at the diagonal opening. If black plays to 64 (6 vertical, 4 horizontal), then the white piece at 54 is between the 2 black pieces at 44 and 64 (just played), so it's flipped: the white piece at 54 becomes black. (By the way, you play black, HP plays white).															
1	2	3	4	5	6	7	8								
1	0	0	0	-	0	0	0								
2	0	0	0	0	0	0	0								
3	0	0	0	0	0	0	0								
4	0	0	0	0	0	0	0								
5	0	0	0	0	0	0	0								
6	0	0	-	0	0	0	0								
7	0	0	0	0	0	0	0								
8	0	0	0	0	0	0	0								

Now, look at the illustration at the left of these lines: if white plays at 14, the black pieces at 12 and 13 are enclosed between the just played piece at 14 and the white piece at 11, so they would be flipped. Simultaneously, the black pieces at 15, 16 and 17 are between the just played piece at 14 and the white piece at 18, so they would be flipped, too.

On the other hand, in the same board position, if black plays at 63, it would flip the white pieces at 62, 53, 43, 33, 23, 64, 65, 66, and 67, because there is another black piece at the end of each row of white pieces, and none of the rows contain empty locations between pieces.

PROGRAM CHARACTERISTICS

The program is exactly 672 bytes (96 registers) long, so it exactly fits onto 3 magnetic cards. The program is optimized for running speed: each location on the board is stored into a single data register, so a minimum SIZE 117 is required. This makes it necessary to have at least 3 single-density memory modules attached, in order to run the program, leaving a port free to plug in the card reader or the printer.

Registers are used as follows: R00 through R07 are scratch. R08 through R15 contain the directions array, necessary to scan each row. R16 through R27 store an array of constants used by the strategic part of the program to compute each move. R17 through R116 store the 8×8 board, including edges (thus being actually a 10×10 board). As you may see, the constants array and the board overlap, so saving 11 registers. This is possible because the edges may be any number except +1 or -1, and none of the constants have those values. White (HP's) pieces are stored as +1, black (yours) ones as -1, and empty locations are 0. The edges are typically 0, but can be any number except +1 or -1.

The program uses flags 1, 2, 3, 4, and 5. If flag 3 is set, your move is being tested for legality, or HP is playing your pieces against its own. If flag 4 is set, a given number is not yet considered legal. If flag 1 is set, HP plays

Printer is set to
Normal Mode

SF 02
XEQ "REVERSI"
DIAG ?

RUN

	1	2	3	4	5	6	7	8
1	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
4	—	—	—	⊗	○	—	—	—
5	—	—	—	○	⊗	—	—	—
6	—	—	—	—	—	—	—	—
7	—	—	—	—	—	—	—	—
8	—	—	—	—	—	—	—	—

HP 1ST ?
N
MOVE ?
64
YOU PLAY 64
FLIP 1 PCES
I PLAY 63
FLIP 1 PCES

RUN

	1	2	3	4	5	6	7	8
1	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
4	—	—	—	⊗	○	—	—	—
5	—	—	—	○	⊗	—	—	—
6	—	—	—	○	⊗	—	—	—
7	—	—	—	—	—	—	—	—
8	—	—	—	—	—	—	—	—

acknowledges your move, and, since flag 02 is cleared, prints the board reflecting your move.

The board is printed. Your move at 76 just flipped the white piece at 65, which became black. You must be aware that this printout is not a direct continuation of the previous one, since we took the game up two moves later.

The machine plays to 66, so flipping once more the piece at 65. As you may see, unlike other games, such as chess or checkers, pieces never move from where they are left, but merely change sides any number of times. Of course, the object of the game is to have the maximum number of pieces on the board when the game ends.

The board is printed now, showing the effects of the machine move on the position.

SAMPLE GAME: Load the program, SIZE 117, and press the following:
SF 02 (selects one board only)

XEQ "REVERSI" : see printout at the left.

- the display asks you whether you want to play DIAGonal opening: you agree by pressing R/S
- the board is printed now reflecting the diagonal opening that you have selected. This is the initial position. You are playing black (checkerboard characters) and HP plays white (the 0's).

(if you are not using a printer, you need an actual 8×8 board, and a set of 64 reversible pieces, one side white, the other black. Dispose them as in the printout, and always actualize the board after your moves and after HP moves).

- the machine prompts you whether it makes the first move
- enter an N and press R/S (N stands for NO): you move first
- the machine then prompts for your move
- enter 64, then R/S (you put a piece at 6 vertical, 4 horizontal)
- the machine tests your move, finds it legal, and acknowledges the move, displaying also the number of flipped pieces
- then computes its move, displays it, the number of pieces it flips, and prints the board

(the board was not printed after your move because we set flag 02)

the board reflects the position after the moves. Your move at 64 flipped the white piece at 54, which became black, but then the machine moved to 63 flipping that same piece once more to white. This is so because by playing at 63 the piece at 54 is enclosed between both white pieces at 63 and 45

... the game continues ... (You:53, HP:65) then, we decide to have a printing of both boards, so we clear flag 02, and enter 76, R/S as our move: (the flag is cleared using the keyboard sequence CF 02). The machine

MOVE ?
CF 02
76 RUN

YOU PLAY 76
FLIP 1 PCES

	1	2	3	4	5	6	7	8
1	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
4	—	—	—	⊗	○	—	—	—
5	—	—	⊗	⊗	○	—	—	—
6	—	—	○	○	⊗	—	—	—
7	—	—	—	—	⊗	—	—	—
8	—	—	—	—	—	—	—	—

I PLAY 66
FLIP 1 PCES

	1	2	3	4	5	6	7	8
1	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
4	—	—	—	⊗	○	—	—	—
5	—	—	⊗	⊗	○	—	—	—
6	—	—	○	○	○	—	—	—
7	—	—	—	—	⊗	—	—	—
8	—	—	—	—	—	—	—	—

your pieces for you. If flag 2 is set and the printer is present, the board will not be printed after your moves (except, of course, if you make the last move). If flag 2 is clear, the board is printed after every move. All flags are controlled by the program, except flag 2, which is user-dependent: you may set or clear it from the keyboard as often as you like. Flag 5 is set before a sequence of board positions is tested. If the flag is set at the end of the sequence, none of the positions tested are valid.

Remember that the program is printer-compatible: if you do not use a printer, it runs the same, except that the board is not printed, of course.

TIPS AND REMARKS

Here are a few typical running times. These times are just the time needed to compute HP moves. They do include time required to print the board, but, of course they do not include the time required for you to think out your own move.

- an average game : 30 HP moves
 - without printer: 25 minutes
 - printer, SF 02 (1 board): 60 minutes
 - printer, CF 02 (2 id.): 75 minutes

As you may see from these figures, the printer slows down significantly the execution speed, but the convenience of the automatic handling of the board, and the fact that an actual board is not needed at all, together with the game being recorded on the paper tape, make it worth the price.

Remember also that execution gets faster as the program progresses, from some 70 seconds for a move near the beginning of the game, to a few seconds for a move near the end of the game. This is possible because HP keeps track of already occupied locations, and once a group of 5 locations is tested to be occupied, they are not tested any more, speeding up the search algorithm quite a lot when the game is close to its termination.

No moves are random, so the same game is played if you make exactly the same moves. This feature is useful: if you made a mistake that allowed HP to win, you can repeat that game once more, this time avoiding the error, to see who wins now. As you'll see, the level of play is quite good for such a tiny program running under the speed limitations of the HP-41C. Any improvements to the playing logic are welcome, however.

There are several ways of making room for improvements, or to fit the program into 2 RAMs (instead of 3). Possible shortcuts are:

- 1) Delete lines 68, 69, change LBL "REVERSI" to LBL "R", line 260 to "OK", and shorten other alpha comments. This saves 27 to 30 bytes at almost no cost.
- 2) If you have no printer, or do not want printing of the board, you can delete lines 6, 62, 195 through 251, 254 through 258 (limits always included) and change line 49 to 60 instead of 61. This modification saves 116 bytes.
- 3) You may use a data card: delete lines 7 through 30 (both included) and insert in their place:
07 16.027
08 RDTAX

This saves another 148 bytes, but a card reader is needed, and you must load a data card when the program asks for one. The data card contains the constants that the program stores (in lines 7 through 30) in their respective registers. See program listings.

Remember that, although the game normally ends when the board is full of pieces, it may end if no player can make a legal move (or if a player loses all his/its pieces). In these cases, the automatic counting of the pieces to decide the winner is not performed: you'll have to do it manually.

```

  1 2 3 4 5 6 7 8
1  O X X X X X X O
2  O X X X X X X -
3  O O O O O O O
4  X X O O O X X O
5  X X O O O O X O
6  X O O O O O O O
7  X O O O O O O O
8  O O O O O O O O

```

```

MOVE ?
      -1  RUN
NO MOVE
I PLAY 28
FLIP 8 PCES

```

```

  1 2 3 4 5 6 7 8
1  O X X X X X X O
2  O O O O O O O O
3  O O O X O O O O
4  X X O O O O X O
5  X X O O O O X O
6  X O O O O O O O
7  X O O O O O O O
8  O O O O O O O O

```

```

GAME IS OVER

```

```

HP: 49, YOU: 15

```

```

I WON

```

SAMPLE GAME CONTINUED: In the printout at the left, a typical game ends.

HP has just moved. Then you are prompted for your move. In the position shown, there is just one empty location left. But you cannot place a piece there, because no white pieces would result enclosed between your piece and another of your pieces. So you have no legal move. However, if you are a beginner, you may be unsure about it, so you decide to have the machine select your move (if any) for you:

Enter -1, R/S. HP begins to search for a suitable move for you. But as expected, finds none, displays (and beeps; you may have noticed by now that most messages are beeped as well as displayed and printed) NO MOVE, then proceeds to search for its move. Finally, after a few seconds, it moves to 28 (where else?!), and, while doing so, flips no less than 8 of your pieces: those located at 22, 23, 24, 25, 26, 27, 37, and 46.

The board is printed for the last time. Then the machine realizes that the game has ended, displays GAME IS OVER, and counts both black and white pieces on the board, to decide the winner. This time, it displays HP: 49, YOU: 15, meaning there are 49 white pieces on the board, while you have only 15 of your pieces remaining. Obviously, HP has won, so it displays a final I WON message. Once this message is on the display, there is only one possibility left for you: TRY AGAIN.

TEST GAME: If desired, test that your program is correctly loaded by executing the following game.

Diagonal opening, HP first. Only the moves are shown (no flip pieces)

YOU	HP	YOU	HP	YOU	HP	YOU	HP	YOU	HP
--	65	42	68	57	85	25	16	38	48
46	33	75	36	83	58	26	52	78	82
64	63	35	84	76	41	32	47	71	87
43	66	86	51	61	34	23	14	12	11
72	53	31	56	62	74	15	73	0	21
67	81	27	18	24	13	17	37	77	88
								<u>22</u>	<u>28</u>

FINAL SCORE: 17 47, so HP WON.

NOTE: If you play with a printer (and set it to NORM, as recommended), you'll have each machine move printed, as well as displayed. However, if you play without a printer, and you happen to miss the I PLAY xy display, do not worry. Simply use backarrow to clear the MOVE ? display, and the last HP's move will be in the display, in the form xy. (Use backarrow just once. Using it twice or more consecutively would also clear the xy move! You can also simply turn alpha on and off to clear the MOVE ? prompt from the display.)

(11-110)				
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the program. You play black. HP white.			
2	If you want to use the printer, plug it in now and set NORM position.			
3	If a printer is used and you want to suppress board printing after your moves, press: the board will now be printed just after HP moves. This can be done at any time		SF 02	Flag 2 annunc. on
or 3	To print the board each time, press:		CF 02	Flag 2 annunc. off
4	Make sure you have at least SIZE 117.			
5	Begin the game, press:		XEQ "REVERSI"	DIAG?
6	If you want to play diagonal opening:		R/S	HP 1ST?
or 6	If you want to play parallel opening:	N	R/S	HP 1ST?
7	If you want HP to make the first move:		R/S	I MOVE
or 7	If you want to make the first move:	N	R/S	MOVE?
8	<u>IF IT IS YOUR TURN</u> (MOVE? on the display) Enter your move (x=vertical, y=horizontal) (Your move is tested for legality. If it is found to be illegal, you'll be prompted once more for your move with MOVE?. Go to Step 9, then)	xy	R/S	YOU PLAY xy FLIP p PCES or ILLEGAL MOVE?
or 9	You have no legal move: enter: and HP proceeds to compute its move.	0	R/S	
or 9	You want the machine to play your pieces against its own in this turn: enter: and HP computes your move, displays: and then automatically computes its own move. NO MOVE is displayed if the machine finds no legal move for your pieces. If you want a whole machine/machine game, always enter -1 as your move.	-1	R/S	YOU PLAY xy FLIP p PCES or NO MOVE
10	<u>IF HP MOVES</u> it will think about its move for a while, then display: xy is the location where HP puts its piece and p is the number of your pieces flipped, NO MOVE is displayed if no legal move is possible for HP. You then have the turn once more: Go to Step 8			I PLAY xy FLIP p PCES or NO MOVE
11	Once the last player makes the last move, you should see: where nn = number of white (HP) pieces on the board mm = number of black (you) pieces on the board <u>Of course, the player with the most pieces at the end of the game wins the game.</u> So, if HP has 24 pieces on the board and you have 40, you won. If HP has 40 and you 24, HP wins. But if both have 32 pieces, it is a tie and no winning message is displayed.			GAME IS OVER HP: nn, YOU: mm I WON or YOU WON

NOTES: If the printer is plugged in, everything that appears in the display is printed as well, and the resulting board position is printed after every legal move if Flag 02 is clear, and only after HP moves if it is set. After the last move, the board is printed also, regardless of the status of Flag 02.

You may set or clear Flag 02 using SF 02 and CF 02 respectively from the keyboard as often as you like. You may do it at any time during program execution, when~ ever the machine is at a halt.

If no player can make a legal move, or if one player loses all his pieces, the game is ended, but this is not recognized by the program, and the automatic counting of the pieces is not performed. Do it yourself, to determine the winner. The board, if not already printed, may be forced to be printed by the following series of keystrokes:

GTO .202

R/S

and halt the program just after the 8th row is printed, by pressing R/S. Once the board is printed, you can perform the counting.

The machine-plays-for-you feature is very useful. You can use it freely whenever you don't know what to play: let the machine play (honestly) your pieces, hoping its selection is a good one. Or, if you are unsure whether you have any legal move or not, let the machine play your pieces:

- if there is a legal move for you it will be found
- if no legal move at all, NO MOVE is displayed, and the machine now computes its own move.

This capability is especially useful for beginners; also, if you want the machine to play a whole game against itself, always enter -1 as your move, and you'll see HP in action as never before!

DATA REGISTERS				STATUS				
01	Scratch			SIZE 117	TOT. REG. 213	USER MODE		
to				ENG	FIX	SCI	ON OFF XX	
07				DEG	RAD	GRAD		
				FLAGS				
08	Directions array			#	INIT S/C	SET INDICATES	CLEAR INDICATES	
to				01	HP playing your pieces			
15				02	Only 1 board		Both boards	
16	Constants array	17	to Board	03	Move tested legal			
to				04	Move not yet legal			
27		116		05	Set before sequence of board positions is tested. If the flag is set at the end of the sequence, none of the positions tested are valid.			
(Constants array and Board overlap)				12	Double wide print		Single wide print	
				23	Alpha input		No alpha input	
			29	Decimal point		Suppress Decimal		
				Flag 29 is also set to indicate the first move of the game.				
			55	Printer exists		No printer		
				ASSIGNMENTS				
FUNCTION		KEY		FUNCTION		KEY		

Program listing:

01*LBL "REVERSI"

02 117	51 CF 23	101 CLAXON
03 XROM "INIT"	52 AON	102 "NO MOVE"
04 CF 01	53 PROMPT	103 AVIEW
05 CF 12	54 RCL 09	104 PSE
06 .8188111883	55 RCL 08	105*LBL 00
07 STO 16	56 FS?C 23	106 FS?C 01
08 .8661683139	57 X<>Y	107 GTO 14
09 STO 17	58 STO 61	108 "MOVE?"
10 .1316636633	59 X<>Y	109 PROMPT
11 STO 18	60 STO 71	110 X=0?
12 .3684855158	61 XEQ 06	111 GTO 14
13 STO 19	62 "HP 1ST?"	112 SF 03
14 .4148141564	63 PROMPT	113 "YOU"
15 STO 20	64 AOFF	114 X<0?
16 .6553564346	65 FS?C 23	115 SF 01
17 STO 21	66 GTO 00	116 X<0?
18 .3435747552	67 "I MOVE"	117 GTO 08
19 STO 22	68 AVIEW	118 XEQ 12
20 .5742472425	69 SF 29	119 FC?C 04
21 STO 23	70*LBL 14	120 GTO 14
22 .7376626732	71 "I"	121 CLAXON
23 STO 24	72 CF 03	122 "ILLEGAL"
24 .3723268287	73*LBL 08	123 AVIEW
25 STO 25	74 16.027	124 GTO 00
26 .7178212812	75 FS?C 29	125*LBL 12
27 STO 26	76 21	126 SF 04
28 .1772772227	77 STO 05	127 17
29 STO 27	78*LBL 11	128 +
30 SIGN	79 RCL IND 05	129 STO 00
31 STO 62	80 X=0?	130 RCL IND 00
32 STO 09	81 GTO 05	131 X#0?
33 CHS	82 SF 05	132 RTN
34 STO 08	83*LBL 13	133 CF 05
35 STO 72	84 RCL 10	134 STO 01
36 9	85 X^2	135 8.015
37 STO 15	86 *	136 STO 02
38 CHS	87 STO 06	137 RCL 09
39 STO 14	88 INT	138 FC? 03
40 +	89 XEQ 12	139 CHS
41 STO 11	90 FC?C 04	140 STO 04
42 CHS	91 GTO 00	141*LBL 01
43 STO 10	92 RCL 06	142 RCL 00
44 11	93 FRC	143 RCL IND 02
45 STO 13	94 X#0?	144 +
46 CHS	95 GTO 13	145 STO 03
47 STO 12	96 FS? 05	146 RCL IND X
48 61	97 STO IND 05	147 RCL 04
49 STO 07	98*LBL 05	148 X#Y?
50 "DIAG?"	99 ISG 05	149 GTO 12
	100 GTO 11	150*LBL 03

151 LASTX	198*LBL 06	245 ADV
152 ST+ 03	199 FC? 55	246 FS? 03
153 RCL IND 03	200 GTO 12	247 GTO 12
154 RCL 04	201 ADV	248 ADV
155 X=Y?	202 31	249 ADV
156 GTO 03	203 STO 00	250*LBL 12
157 CHS	204 45	251 DSE 07
158 X#Y?	205 STO 01	252 RTN
159 GTO 12	206 79	253 FC? 02
160 STO IND 00	207 STO 02	254 GTO 12
161*LBL 04	208 2.01	255 FS?C 03
162 LASTX	209 STO 03	256 XEQ 06
163 ST- 03	210 8	257*LBL 12
164 RCL 00	211 SKPCOL	258 32
165 RCL 03	212 49.056	259 "GAME OVER"
166 X=Y?	213 STO 04	260 28.105
167 GTO 12	214*LBL 02	261 AVIEW
168 RCL 08	215 RCL 13	262 0
169 ST* IND Y	216 SKPCOL	263*LBL 07
170 ST- 01	217 X<>Y	264 RCL IND Y
171 GTO 04	218 ACCHR	265 +
172*LBL 12	219 ISG X	266 ISG Y
173 ISG 02	220 GTO 02	267 GTO 07
174 GTO 01	221 PRBUF	268 2
175 RCL 01	222 28.035	269 /
176 X=0?	223 STO 05	270 X<>Y
177 RTN	224*LBL 09	271 RDN
178 CF 04	225 RCL 04	272 ST- Z
179 >" PLAY "	226 ACCHR	273 +
180 RCL 00	227 RCL 15	274 ADV
181 17	228 SKPCOL	275 "HP: "
182 -	229 SF 12	276 ARCL X
183 ARCL X	230*LBL 10	277 >," YOU:"
184 AVIEW	231 RCL IND 05	278 ARCL Y
185 FC?C 01	232 RCL 09	279 AVIEW
186 FC? 03	233 +	280 BEEP
187 BEEP	234 RCL IND X	281 ADV
188 PSE	235 ACCHR	282 PSE
189 "FLIP "	236 RCL 03	283 X=Y?
190 ARCL 01	237 SKPCOL	284 STOP
191 >" STNS"	238 ISG 05	285 "I"
192 AVIEW	239 GTO 10	286 X<Y?
193 PSE	240 PRBUF	287 "YOU"
194 FC? 02	241 ST+ 05	288 >" WON"
195 GTO 06	242 CF 12	289 PROMPT
196 FS? 03	243 ISG 04	290 END
197 GTO 12	244 GTO 09	

5x5 Mini-Chess

Valentín Albillo – PPCCJ V8N6 p66

This program challenges the user to play chess against the 41C. The game is played in a 5x5 board instead of the standard 8x8 (see reasons below) but this hardly matters, as - all standard chess rules are implemented, including pawn promotion.

The program is absolutely printer-compatible, but if a printer is present, it will print the board, making extensive use of the graphic capabilities of the printer. Also, you may have the board printed after every move, or just after HP moves, to save paper and time.

I originally wrote an 8x8 game but:

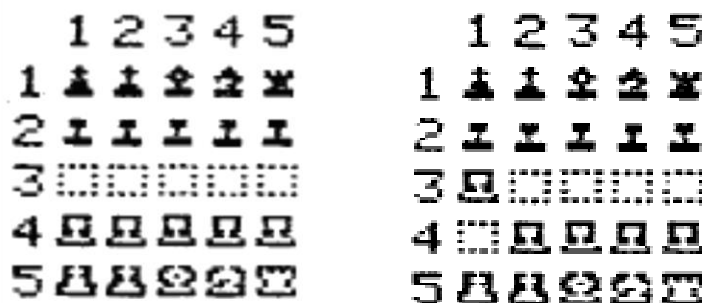
a) an 8x8 board cannot be printed using special characters, because of printer limitations – The buffer cannot hold more than 44 columns at a time, and each special character already takes 7 columns. The board could be printed using numbers to identify the pieces, or some combinations of characters, but even the best attempt was much worse and unrecognizable than the present version.

b) 8x8 game took the full memory of a 41C (with 4 RAM modules), so unless you had a 41CV or a Quad module neither the printer nor the card reader could be plugged, making very difficult to load and run the program.

c) 8x8 game, using the same playing logic as this 5x5 version, took several hours per move, playing very weak, and thus making the game uninteresting. This version requires 3 memory modules, card reader and optionally the printer.

On the other hand, this 5x5 version provides the following advantages:

- a) The board is printed using BLDSPEC special characters, so you can clearly see the position without using an actual board. All handling of the board is automatic.
- b) Though the board is 5x5, you still have all pieces of conventional chess arranged in the same order (see illustration below): king, queen, bishop, rook and a row of pawns. All pieces have the same powers and restrictions as in standard chess.



- c) This 5x5 version fits in 3 memory modules, leaving a port free to plug the card reader and the printer if desired. Also, due to the reduced size game progresses faster than in 8x8 chess, taking an average of 20 moves per game (8x8 averages 40), making the game more active. Both armies get into battle very soon.

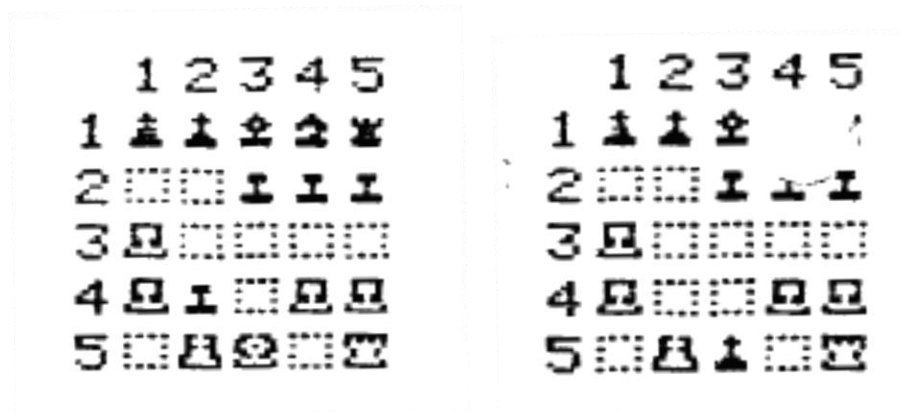
And also, as the number of alternatives for a given position is less than in 8x8, the machine level of play is much better, so that HP plays a quite good, non-trivial game. It can checkmate you if you don't play fine enough!

All standard rules of chess are implemented, with the following three exceptions:

1. Since the king is already in a corner, no castling is necessary.
2. Since there's only one empty row between the pawns, a pawn may only advance one position on its first move
3. No capture "en passant" is allowed.

As you see these exceptions are mostly due to the board size. All other rules are unchanged, for instance pawn promotion is allowed: like in standard chess if a pawn reaches the opposite side it becomes any piece desired by the player (except king and another pawn, obviously).

Such an example is given in the illustration below: HP moves its pawn in 42 (standard row/column matrix notation: 4 is vertical, 2 is horizontal) to 53, this taking the white bishop at that location (you always play white, BTW), it becomes a queen (see printout) and gives check (not shown). In case of pawn promotion HP always selects a queen, but you may choose any piece you wish.



If some HP move results in a check being given to your king, the machine shows the word CHECK after its move. There are two exceptions to this rule:

1. If a pawn promoted to a queen by HP results in a check that is not indicated
2. If HP moves a piece that, while not giving check by itself, leaves your king under attack from some other HP piece then that check is not indicated either.

HP will never make illegal moves, but your moves are not tested for legality (you are assumed to play honestly). If your king is under check and you forget the fact and move some other piece HP will actually take your king on its next move!

HOW IT WORKS

Here is a brief and concise explanation of the program internal mechanics. First of all, the board, though it is 5x5, is stored including edges, thus it becomes a 9x9 board (edges are two squares wide). The edges' are necessary to simplify the "move-a-piece" algorithms, thus saving program memory and, more important, time required for a move. However, a 9x9 board would take 81 registers. That's too much. First a saving can be made, because the upper left corner and the lower right one can be suppressed, saving 2 registers. But then,

one realizes that, as the edges must contain alpha constants, any alpha, the BLDSPEC characters may be stored on an edge. That saves 13 additional registers. Further, the bottom edge may be suppressed if we simply make use of flag 25 (the error flag) to detect those NONEXISTENT registers: if a register is nonexistent, it is an edge ! (this makes necessary to have a size of exactly 097. Otherwise, the register would exist !). This saves another 19 registers. Thus the 9x9 board takes just 47 registers, instead of 81. Very good saving, indeed !

Now, the pieces are stored as a code in the location where it stands. The code is composed of two parts: the integer part is the code itself, positive for white pieces, negative for black ones. The decimal part is the "value" the machine gives to the piece. Those codes are:

king = 6.50
queen = 5.09
bishop = 4.03
knight = 3.03
rook = 2.05
pawn = 1.01

so the king is considered to have a value of 50, the queen is 9, rook is five, bishop & knight are of the same value, and pawn is worth 1. This is accordingly to the standard chess valuation for pieces. Empty locations have a 0 value.

In chess, almost every piece moves in a different way. So a "move generator" is programmed, which generates all legal moves for any given piece. The algorithm to decide the move is as follows.

let G = maximum loss for a move (particularized for a given move) -
and T = minimum gain for a move (general)

Initially, set T to -99. Then, scan the board to find an HP piece. Once any HP piece is found, generate a move for that piece. Test to see if the generated move is illegal. If it is, generate another move for the piece. On the other hand, if it is legal, call the valuation routines

The evaluation routine assigns a value for a move, taking into account the following factors:

- material gained (i.e. captures and promotion)
- material lost
- pawn position
- attacks to the enemy king
- attacks to HP's king (or the player's, whichever is being evaluated)

all those factors are given some weight, and merged into a single value V.

If the value V1 is less than or equal to T, discard that move, and generate another. If it is not, save the position, make the move in the board, set G to 99, and scan the board for a white piece. Once found, generate a move for that piece. Test its legality. Call the evaluation routine, etc, etc.

The final outcome is a value for the minimum gain (once all possible moves for black and white pieces, and respective responses have been confronted, evaluated, etc), together with the move (recorded) which produces this minimum gain. If the gain is -99, HP has been checkmated (or stalemated, see User instructions). Otherwise it performs and displays the

move which results in this minimum gain. The algorithm uses thus, resembles the alpha-beta algorithm used in computer chess programs.

Remember: your moves are not tested for legality.

There are two exceptions to the check status indication

Castling, capture "en passant" and pawns moving two locations forward at the beginning are not allowed.

Size must exactly be 097 (no more, no less)

Do not make any changes to the program, unless you want it to have bugs. Specifically don't add any subroutines: all 6 levels are used up/

Reference: Martin Gardner described the 5x5 version of chess in one of his remarkable books on Recreative Mathematics.

Sample Game.

Let's try it. Make sure you have a size of exactly 097 (must be exact, no more no less, because the error flag is used to detect non-existing registers above 096 – which saves 20 registers. If the size were greater than 097 some of these registers would exist, causing unintended errors. Load the printer "P" routine and set the printer in NORM position (if you have no printer, skip this procedure). Press [SHIFT] GTO . . see PACKING momentarily. Now load the main file MCHES. Press:

XEQ "MCHES", note that this version will load the needed data automatically from an X-Mem file, so no use for the card reader as in the original version.

If the printer is connected the initial board will be printed out. You're white and HP is black. Next you're prompted with "HP 1st?" to know who makes the first move. You want to make the move, so press "N", followed by R/S.

Next the program asks for from the FROM / TO locations to specify your piece and move. Say you move from 41 to 31 which you'd duly input, following each number by R/S of course.

Now the program shows "I MOVE" while it thinks – and if flag 0 is set then the board will be printed as well – if not, the printing will only occur after HP moves, not everytime.

<pre> 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 □ □ □ □ □ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ </pre>	41 → 31	<pre> 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ □ □ □ □ 4 □ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ </pre>	22 → 31	<pre> 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ □ ♠ ♠ ♠ 3 ♠ □ □ □ □ 4 □ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ </pre>
--	---------	--	---------	--

"I MOVE" is scrolled in the display while HP thinks, then several minutes later it displays: "FROM 22 to 31, and prints the board. As you may see from the figures, you moved your pawn one step forward and HP captured it with its pawn at 22 (remember the row/column matrix notation). The game continues as shown in the printouts below: you advance your pawn at 43, then HP captures your pawn at 42, giving check. You move your king and HP

moves once more its pawn to 53, capturing your bishop and getting promoted to a queen (see the second black queen at 53!)

1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠
2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠
3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠
4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠
5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠

You now decide to capture the pawn at 24 with your pawn at 33, and HP captures your queen with its own queen at 53, giving check. You then realize you have been checkmated, because no move will save your king from the attack of the queen at 52, protected by the second queen at 12 and the bishop at 13 impedes your retreat. You input "-1" and HP acknowledges the victory with a happy "CHECKMATE", "I WON" message. Bette rluck next time!

1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠
2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠
3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠
4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠
5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠

End-Game Examples

In the position shown above (on the right) , you move your king from 54 to 33, giving check to the black king at 21 (and also menacing the queen at 12). HP prints the new position and proceeds to consider its response. It soon finds none, and decides you have given checkmate – so it displays the "I MOVE, CHECKMATE, YOU WON message sequence.

In the position shown below (on the left) HP moves and wins. Black moves its queen from 35 to 55 taking your rook at 55, and gives check to the white king at 51. Then your move is requested after the position is printed. You suddenly find, to your dismay, that no move will save your king from the attack so you've been checkmated: enter -1 as your move and HP acknowledges the victory displaying "CHECKMATE", "I WON".

1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠	1 ♠ ♠ ♠ ♠ ♠
2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠	2 ♠ ♠ ♠ ♠ ♠
3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠	3 ♠ ♠ ♠ ♠ ♠
4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠	4 ♠ ♠ ♠ ♠ ♠
5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠	5 ♠ ♠ ♠ ♠ ♠

As a third case, in this other position (shown above, on the left), you're in trouble. Your only pieces left are your king at 52 and a blocked pawn at 34. Now HP moves its bishop from 53 to 31 and requests your move. But you cannot move at all, because the pawn is blocked and your king, though not under check, is surrounded by enemy pieces and has no legal move either. You then enter zero (0) as your move and the machine acknowledges the stalemate displaying "STALEMATE"; a tie.

Sample game and times.

If desired test that your program is correctly loaded by running this game, where you play first. Check is indicated with a plus sign (+), and numbers in brackets represent the code of pieces obtained by pawn promotion (answers to the "PIECE?" prompt). Times are given too.

YOU	HP	no printer	printer
44-34	25-34	3'08	4'45
43-34	23-34	3'08	4'45
53-44	34-45	2'16	3'26
54-33	24-33	3'25	5'11
42-33	22-32	4'04	6'11
33-23	12-22	7'18	11'05
23-14 (3.03)	22-44	6'25	9'45
55-53	45-55 (queen)	6'38	10'05
53-55	15-14	14'40	22'17
41-31	13-31	10'48	16'25
52-54	44-54+	4'20	6'35
55-54	14-54+	4'51	7'22
-1	CHECKMATE-I WON		
	total time =	71'01	107'52
	average per move =	5'55	8'59

Notes: the 3.03 is the code for a knight. Your pawn promotes, and you chose a knight. 2 turns later, HIP promotes its pawn and selects a queen. As you can see, using the printer Slows down the execution time by a factor of 1.52 (52% slower). Anyway, this is not an average example; it has been chosen to show maximum times. For instance, the 14'40-seconds required to find the move 15-14, is a maximum: the 41c had to explore some 750 - moves to find the answer, so the time had to be large. That's so because HP had 26 possible options, each one having at least 19 responses from you, etc. If you want to shorten times when playing, simplify the position, change pieces, avoid open positions, etc. The execution time depends quadratically of the number of HP options and linearly of the number of your responses to each option.

Program listing:

01*LBL "MNCHSS"	15 STO 94	30 -	45 sREG 74
02 97	16 ST- 58	31 STO 95	46 CLs
03 XROM "INIT"	17 E1	32 ST- 59	47 ASTO 79
04 XROM	18 E3/E+	33 2.05	48 FS? 55
"CHDTA"	19 STO 83	34 STO 96	49 XROM "CPRT"
05 9	20 STO 84	35 ST- 60	50 "HP 1ST? Y/N"
06 STO 16	21 STO 85	36 "A"	51 AVIEW
07 ST- 17	22 STO 86	37 50.091	52*LBL 55
08 6.5	23 STO 87	38 SIGN	53 GETKEY
09 STO 92	24 ST- 65	39*LBL 01	54 71
10 ST- 56	25 ST- 66	40 RCL IND L	55 X=Y?
11 5.09	26 ST- 67	41 X=0?	56 GTO 00
12 STO 93	27 ST- 68	42 ASTO IND L	57 30
13 ST- 57	28 ST- 69	43 ISG L	58 -
14 4.03	29 INT	44 GTO 01	59 X#Y?

60 GTO 55	113 X<Y?	166 "CHECKMATE"	219 RCL 02
61*LBL 99	114 GTO 04	167 AVIEW	220 STO 05
62 CF 00	115 CLX	168 CLAXON	221*LBL 15
63 FS? 55	116 X<> IND 12	169 CLA	222 RCL IND 04
64 XEQ 16	117 STO IND 13	170 ARCL X	223 ST+ 05
65 "FROM?"	118 XEQ 08	171 "" WON"	224 RCL 05
66 PROMPT	119 "FROM "	172 PROMPT	225 XEQ 08
67 "I"	120 RCL 12	173*LBL 05	226 FS? 18
68 X<0?	121 XEQ 09	174 "STALEMATE"	227 GTO 00
69 GTO 04	122 "" TO "	175 RASP	228 X<0?
70 X=0?	123 RCL 13	176 PROMPT	229 GTO 00
71 GTO 05	124 XEQ 09	177*LBL 09	230 CF 09
72 XEQ 06	125 RCL 00	178 INT	231 X=0?
73 STO 00	126 FRC	179 ENTER^	232 SF 09
74 "TO?"	127 RCL 22	180 ENTER^	233 XEQ 12
75 PROMPT	128 *	181 9	234 FS? 05
76 XEQ 06	129 FRC	182 /	235 GTO 00
77 STO 01	130 X#0?	183 INT	236 FS? 09
78 CLX	131 "", CHECK"	184 +	237 GTO 15
79 X<> IND 00	132 RASP	185 51	238*LBL 00
80 STO IND 01	133 AVIEW	186 -	239 ISG 04
81 XEQ 07	134 FS? 55	187 ARCL X	240 GTO 14
82 FC? 55	135 XROM "CPRT"	188 RTN	241 RTN
83 GTO 00	136 FC? 55	189*LBL 06	242*LBL 36
84 FS? 00	137 STOP	190 ENTER^	243 SF 07
85 XROM "CPRT"	138 GTO 99	191 ENTER^	244*LBL 33
86*LBL 00	139*LBL 07	192 E	245 SF 05
87 "I MOVE"	140 60	193 -	246*LBL 32
88 AVIEW	141 RCL 01	194 5	247*LBL 34
89 PI	142 X>Y?	195 /	248*LBL 35
90 STO 09	143 RTN	196 INT	249 RTN
91 96.055	144 2	197 2	250*LBL 13
92 STO 02	145 RCL IND 01	198 /	251 SF 06
93 CHS	146 X>Y?	199 -	252 RCL 02
94 STO 00	147 RTN	200 46	253 9
95*LBL 11	148 "PIECE?"	201 +	254 XEQ 09
96 RCL IND 02	149 PROMPT	202 RTN	255 FS? 18
97 SIGN	150 STO IND 01	203*LBL 12	256 E
98 X=0?	151 RTN	204 STO 03	257 X=0?
99 GTO 00	152*LBL 08	205 ABS	258 XEQ 12
100 LASTX	153 ABS	206 CF 05	259 RCL 02
101 X<0?	154 2	207 CF 06	260 RCL 22
102 XEQ 12	155 X<Y?	208 CF 07	261 XEQ 09
103*LBL 00	156 RTN	209 2	262 FS? 18
104 DSE 02	157 92	210 X>Y?	263 CLX
105 GTO 11	158 RCL 13	211 GTO 13	264 X>0?
106 RCL 09	159 X<Y?	212 X<>Y	265 XEQ 12
107 PI	160 RTN	213 30	266 RCL 02
108 X=Y?	161 -5.09	214 +	267 8
109 GTO 05	162 STO IND 13	215 XEQ IND X	268 XEQ 09
110 "YOU"	163 RTN	216 RCL IND X	269 FS? 18
111 -25	164*LBL 04	217 STO 04	270 RTN
112 RCL 00	165 ASTO X	218*LBL 14	271 X<=0?

272 RTN	325 STO IND 02	378 FS? 18	431 SF 01
273*LBL 12	326 RCL 07	379 RTN	432*LBL 32
274 CF 08	327 STO IND 08	380*LBL 12	433*LBL 34
275 STO 07	328 RTN	381 INT	434*LBL 35
276 FRC	329*LBL 12	382 6	435 RTN
277 E2	330 .5	383 X#Y?	436*LBL 12
278 *	331 ST+ 06	384 RTN	437 SF 02
279 STO 06	332 92	385 .41	438 RCL 11
280 RCL Z	333 RCL 08	386 ST+ 06	439 RCL 17
281 STO 08	334 X<Y?	387 RDN	440 XEQ 09
282 .4	335 RTN	388 RTN	441 FS? 18
283 FS? 07	336 SF 08	389*LBL 07	442 E
284 ST- 06	337 9	390 CF 01	443 X=0?
285 FS? 06	338 ST+ 06	391 CF 02	444 XEQ 13
286 XEQ 12	339 RTN	392 CF 03	445 FS? 19
287 FC? 07	340*LBL 13	393 2	446 RTN
288 XEQ 13	341 FS? 06	394 X>Y?	447 RCL 11
289 RCL 00	342 GTO 13	395 GTO 12	448 RCL 20
290 RCL 06	343 RCL 03	396 X<>Y	449 XEQ 00
291 X<=Y?	344 30	397 30	450 FS? 19
292 RTN	345 -	398 +	451 RTN
293 RCL 03	346 RCL IND X	399 XEQ IND X	452 RCL 11
294 FS? 08	347 STO 01	400 RCL IND X	453 RCL 21
295 -5.09	348*LBL 03	401 STO 14	454*LBL 00
296 STO IND 08	349 RCL 08	402*LBL 28	455 XEQ 09
297 CLX	350 STO 10	403 RCL 11	456 FS? 18
298 STO IND 02	351*LBL 10	404 STO 15	457 RTN
299 56.096	352 RCL IND 01	405*LBL 29	458 X#0?
300 STO 11	353 ST+ 10	406 RCL IND 14	459 X>0?
301 STO 09	354 RCL 10	407 ST+ 15	460 RTN
302 CF 19	355 XEQ 08	408 RCL 15	461*LBL 13
303*LBL 21	356 FS? 18	409 XEQ 08	462 FRC
304 RCL IND 11	357 GTO 00	410 FS? 18	463 ABS
305 SIGN	358 XEQ 12	411 GTO 00	464 E2
306 X=0?	359 X=Y?	412 X>0?	465 *
307 GTO 00	360 RTN	413 GTO 00	466 FS? 03
308 LASTX	361 FS? 05	414 CF 10	467 .4
309 X<=0?	362 GTO 00	415 X=0?	468 FS? 03
310 GTO 00	363 LASTX	416 SF 10	469 -
311 XEQ 07	364 X=0?	417 XEQ 13	470 FS? 02
312 FS? 19	365 GTO 10	418 FS? 19	471 XEQ 13
313 GTO 04	366*LBL 00	419 RTN	472 RCL 06
314*LBL 00	367 ISG 01	420 FS? 01	473 X<>Y
315 ISG 11	368 GTO 03	421 GTO 00	474 -
316 GTO 21	369 RTN	422 FS? 10	475 RCL 00
317 RCL 09	370*LBL 13	423 GTO 29	476 X<>Y
318 STO 00	371 RCL 08	424*LBL 00	477 X<=Y?
319 RCL 02	372 RCL 22	425 ISG 14	478 SF 19
320 STO 12	373 XEQ 00	426 GTO 28	479 X<=Y?
321 RCL 08	374 RCL 08	427 RTN	480 RTN
322 STO 13	375 8	428*LBL 36	481 RCL 09
323*LBL 04	376*LBL 00	429 SF 03	482 X<>Y
324 RCL 03	377 XEQ 09	430*LBL 33	483 X<Y?

484 STO 09	494 SF 04	503 "PRINT	512*LBL 08
485 RTN	495 RCL Z	BOTH?"	513 CF 18
486*LBL 13	496 9	504 AON	514 SF 25
487 .5	497 FC? 04	505 PROMPT	515 RCL IND X
488 +	498 CLX	506 AOFF	516 SIGN
489 RCL Z	499 +	507 FS?C 23	517 FS?C 25
490 60	500 RTN	508 SF 00	518 X=0?
491 X<>Y	501*LBL 16	509 RTN	519 SF 18
492 CF 04	502 CF 23	510*LBL 09	520 LASTX
493 X<=Y?		511 +	521 END

Creating the X-Mem file with the data:

01*LBL "CHDTA"	25 CHS	49 "-----7 3"
02 GOOSE	26 STO 26	50 XEQ 00
03 E	27 11	51 "-----o *"
04 STO 18	28 STO 31	52 XEQ 00
05 CHS	29 CHS	53 "T-----@U"
06 STO 19	30 STO 27	54 XEQ 00
07 E1	31 16.019	55 "6o "
08 STO 22	32 STO 32	56 XEQ 00
09 CHS	33 24.031	57 "80 "
10 STO 20	34 STO 33	58 XEQ 00
11 8	35 20.023	59 " i "
12 STO 23	36 STO 34	60 XEQ 00
13 CHS	37 16.023	61 " m "
14 STO 21	38 STO 35	62 XEQ 00
15 19	39 STO 36	63 " o "
16 STO 28	40 37.049	64 XEQ 00
17 CHS	41 "-----*"	65 "Xo "
18 STO 24	42 XEQ 00	<u>66*LBL 00</u>
19 17	43 "-----/ *"	67 LEFT
20 STO 29	44 XEQ 00	68 RCL [
21 CHS	45 "-----& { *"	69 STO IND Y
22 STO 25	46 XEQ 00	70 RDN
23 7	47 "-----n *"	71 ISG X
24 STO 30	48 XEQ 00	72 END

```

RR037 = 0; 96; 122; 127; 122; 96; 0;
RR038 = 0; 96; 114; 127; 114; 96; 0;
RR039 = 0; 100; 110; 123; 110; 100; 0;
RR040 = 0; 108; 102; 119; 126; 108; 0;
RR041 = 0; 102; 124; 126; 124; 102; 0;
RR042 = 0; 96; 102; 126; 102; 96; 0;
RR043 = 85; 0; 65; 0; 65; 0; 85;
RR044 = 96; 95; 89; 65; 89; 95; 96;
RR045 = 103; 89; 67; 65; 67; 89; 103;
RR046 = 110; 83; 89; 72; 65; 83; 126;
RR047 = 110; 91; 81; 68; 81; 91; 110;
RR048 = 112; 95; 77; 64; 77; 95; 112;
RR049 = 112; 95; 69; 64; 69; 95; 112;

```


Checkers 3.0

Valentín Albillo –PPCCJ V8N1 p31

This program allows the user to play a game of checkers against the 41C. Checkers is a most popular game, known all over the world, though the rules vary slightly from country to country.

This program, similarly to Othello, is printer-compatible (runs with or without printer), but if a printer is present, it will print the board automatically, freeing the user of the handling of the board. The printer routine is independent from Checkers 3.0, so members without a printer do not have to make any changes to the program.

It allows the user to select who moves first. It plays as fast as possible for such a complex game, taking from 1 to 2 minutes to perform its move if no printer. (The printer slows down the execution some 35%). The strategy used is quite good, offering a real challenge for everyone.

Checkers is played on an 8 by 8 board. One player plays white (represented by the O's), the other player plays black (represented by the checkerboard character). In this version, you play always black, though you can select who makes the first move.

Each player has 12 pieces at the beginning of the game, set as in the standard position (see below). To make a move, you should place one of your pieces into one of the adjacent squares to the one in which your piece was, and diagonally in the direction of the enemy (this is, always forwards from you, always backwards for the machine pieces). Thus you can move a piece by placing it into a free location situated diagonally adjacent (and in front) of the location where your piece is currently. Your pieces always move forwards, the machine's always backwards (that is, each player moves in the enemy side's direction). Or you can take a piece of the enemy, which also has to be located diagonally adjacent to one of yours, by jumping with your piece over it, and placing it in a free location (the next free location always moving diagonally adjacent to the enemy's piece).

Multiple jumps are allowed when you capture a piece, jump over it to the next free location, and if from that location you can make another capture with the same piece, then you must do it, taking as many pieces in the move as you legally can.

Capture is compulsory: if you can take a piece you must do it. However, if you have several different captures possible you may select any of them, not just the one resulting in the maximum number of pieces taken. Pieces capture in the direction they move. If a piece reaches the opposite side of the board (row 0 or 7), it becomes a king. MKings move and take exactly as a normal piece, but can do it backwards and forwards (always diagonally, of course). The king may be distinguished from the normal pieces as follows: white ones are represented as zeroes, black ones as crosshatch. See the figure at the left.

Each location is represented by a 2-digit number following the coordinate system of the figures: for instance, 15 means 1 horizontal, 5 vertical. In the figure there is one black king at 37 and another at 55, one white king at 22, another at 20. If white moves it can play from 20 to 42 (taking the black piece) to 64 (taking a second piece), or from 46 to 64 (taking the black king) to 42 (taking another black piece). Or even from 22 to 40 (taking the black piece). Or from 66 to 44 (taking a black king) to 62 (taking another piece). No other moves are legal, since capturing is mandatory. If black plays, he may move from 37 to 15 (taking a piece) to 33 (taking another piece) to 11 (taking a white king). A good example of multiple

jump!, or move from 31 to 13 (taking the king) to 35 (takes another piece) to 17 (taking still another and becoming a king). That one is even better! This should make clear the mechanics of multiple jumps, and how a piece becomes a king too.

How to use the program.

If you have no printer load just the main program.

If you have a printer load the printing routine first, then GTO.. (PACKING) and load the main program. Set the printer to NORM position.

The main program has 502 lines (760 bytes long)

The printing routine has 56 lines (102 bytes long)

And requires SIZE 084 to run. If you use the printer routine, you'll need 3 memory modules. If you don't print, don't load the printer routine and that will save you one RAM module (so you'll need only two). Also if you have a printer but don't intend to print the board, unplug the printer now.

Now XEQ "CHKERS" -> the board will be printed, reflecting the standard initial position – and you'll be prompted: "HP 1.ST?"

If you want HP to make the first move press R/S. "I MOVE" appears while the machine thinks the move.

If you want to make the first move, press "N", then R/S. This will prompt for your move : "MOVE?"

- a) If the machine moves (I MOVE in the display) it will think its move for a while (from 1 to 2 minutes), then print: I MOVE → FROM xx TO yy (where xx and yy are the coordinates of the initial location of the moved piece, and the location where it moves to.

If it is a multiple jump. "TO zz" appears, showing the successive new locations where the piece is placed.

If a printer is present the resulting board position is printed now. Otherwise you need to update the board yourself, removing the jumped pieces from it if necessary. Once the board is printed the machine prompts for your move: → "MOVE?"

- b) If you move (you have been prompted by "FROM?"):
Input the xx for the location where the piece you want to move is now: xx, R/S → "TO?"
Then input the yy for the location where the piece moves to: yy, R/S

Very important: your move is not tested for legality, so you must be very careful not to make illegal moves. Remember, always move diagonally and forwards (except if a king is moving, which can move backwards too); one square at a time – except when taking a piece, which will jump over it. Capture is compulsory).

If you took a piece the machine asks you for another jump, making thus a multiple jump possible. If you can jump once more (over another enemy piece, naturally) input the zz coordinates of the location where the piece moves to.

Multiple jumps example.

Suppose the board is set as shown in the figure to the left, and that it's your turn to move. Say you move FROM 40 to 31, leaving the conditions for a multiple jump to be done by the machine. Not surprisingly the program responds with "I MOVE FROM 04 to 22... TO 40, setting the piece on the bottom row and thus becoming a king and leaving thing as shown in the figure in the center:

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

If you then move from 62 to 53 the king at 40 jumps to 62 (taking your piece at 51), then to 44 (taking your piece at 53), and further to 22 (taking your piece at 33). Good jump! See the figure to the right for this stage.

An example of amazing jumps. Suppose the board is set as shown in the figure to the left, and that it's your turn to move. Say you move from 40 to 51, and then the program does I MOVE FROM 35 to 24

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

	0	1	2	3	4	5	6	7
7	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—

If you move from 22 to 13 the machine makes a jump from 04 to 22 (taking your piece at 13), then to 40 (taking your piece at 31 and becoming a king, further to 62 (the piece continues the multiple jump, now being a king and taking your piece at 51), then to 44 (taking the piece at 53), then to 22 (being a king it can jump backwards and forwards, so it takes your piece at 33). Incredible jump that has cleaned out half the board!

Remarks: No moves are random. The same game is played if you make always the same moves. The level of play is quite good for such a difficult game, as you'll see. Do not make any changes to the program. There are sequences of instructions that seem to be easily improved (for instance GTO 27... LBL 27, XEQ 28, RTN, LBL 28... RTN). Do not "improve" them or you'll find yourself with a program making illegal moves from time to time (a cheating player!)

Test Game. If desired, test that your program is correctly loaded by executing the following game sequence, HP first.

15-04/02-13, 06-15/11-02, 15-24/22-33, 04-22/31-13, 26-15/
13-04, 17-26/62-53, 55-64/33-44, 46-55/42-33, 24-42/51-33, 64-42/
40-31, 35-53/02-13, 55-64/13-24, 64-73/24-06, 75-64/06-17, 66-75/
17-35, 73-62/04-15, 64-73/15-06, 75-64/06-17, 62-51/33-24, 51-40/
24-15, 40-22/20-11, 22-33/11-22, 33-11/00-22, 42-31/60-51, 31-20/
35-44, 20-11/44-62, 11-33/17-26, 33-24/15-06, 37-15/06-17, 24-35
17-06, 35-26/06-24, 26-17/, etc, etc, etc.

Program listing:

01*LBL "CHKERS"	38 PROMPT	75 STO 02	112 RTN
02 84	39 AOFF	76 RCL 03	113 X<0?
03 XROM "INIT"	40 FS?C 23	77 XEQ 00	114 RTN
04 CF 12	41 GTO 46	78 RTN	115 RCL 05
05 E	42*LBL 20	79*LBL 02	116 XEQ 13
06 STO 20	43 "I MOVE"	80 - E	117 X=0?
07 STO 36	44 AVIEW	81 E	118 GTO 27
08 STO 52	45 7 E-3	82 STO 02	119 X<0?
09 STO 68	46 STO 00	83 XEQ 00	120 RTN
10 STO 29	47*LBL 09	84 E	121 RCL 02
11 STO 45	48 7 E-3	85 XEQ 00	122 ST+ 04
12 STO 61	49 STO 01	86 E	123 RCL 03
13 STO 77	50*LBL 10	87 STO 02	124 ST+ 05
14 STO 22	51 - E	88 CHS	125 RCL 05
15 STO 38	52 E	89 XEQ 00	126 X<0?
16 STO 54	53 RCL 00	90 E	127 RTN
17 STO 70	54 RCL 01	91 XEQ 00	128 7
18 CHS	55 XEQ 13	92 RTN	129 X<Y?
19 STO 33	56 STO 15	93*LBL 00	130 RTN
20 STO 49	57 X>Y?	94 STO 03	131 RCL 04
21 STO 65	58 GTO 00	95 RCL 00	132 X>Y?
22 STO 81	59 X=Y?	96 INT	133 RTN
23 STO 26	60 XEQ 01	97 RCL 02	134 X<0?
24 STO 42	61 RCL 15	98 +	135 RTN
25 STO 58	62 -2	99 STO 04	136 RCL 05
26 STO 74	63 X=Y?	100 RCL 01	137 XEQ 13
27 STO 35	64 XEQ 02	101 INT	138 X=0?
28 STO 51	65*LBL 00	102 RCL 03	139 XEQ 28
29 STO 67	66 ISG 01	103 +	140 RTN
30 STO 83	67 GTO 10	104 STO 05	141*LBL 27
31 -99	68 ISG 00	105 X<0?	142 XEQ 28
32 STO 09	69 GTO 09	106 RTN	143 RTN
33 FS? 55	70 GTO 38	107 7	144*LBL 28
34 XROM "KPRT"	71*LBL 01	108 X<Y?	145 RCL 05
35 CF 23	72 STO 02	109 RTN	146 X#0?
36 "HP 1ST?"	73 XEQ 00	110 RCL 04	147 GTO 00
37 AON	74 E	111 X>Y?	148 RCL 00

149 RCL 01	202 RCL 00	255 INT	308 X#Y?
150 XEQ 13	203 INT	256 RCL 18	309 GTO 45
151 - E	204 STO 10	257 X#Y?	310 .
152 E	205 RCL 01	258 RTN	311 RCL 10
153 X#Y?	206 INT	259 RCL 01	312 RCL 12
154 GTO 00	207 STO 11	260 INT	313 STO 00
155 2	208 RCL 04	261 RCL 07	314 +
156 ST+ 06	209 STO 12	262 X#Y?	315 2
157*LBL 00	210 RCL 05	263 RTN	316 /
158 RCL 01	211 STO 13	264*LBL 00	317 RCL 11
159 INT	212 RTN	265 2	318 RCL 13
160 RCL 05	213*LBL 04	266 ST- 06	319 STO 01
161 -	214 STO 08	267 RTN	320 +
162 ABS	215 7	268*LBL 05	321 2
163 2	216 RCL 04	269 E	322 /
164 X#Y?	217 RCL 08	270 ST+ 06	323 XEQ 12
165 GTO 00	218 +	271 RTN	324 RCL 16
166 5	219 X<0?	272*LBL 38	325 - E
167 ST+ 06	220 RTN	273 RCL 09	326 E
168*LBL 00	221 X>Y?	274 -99	327 X#Y?
169 7	222 RTN	275 "YOU WIN"	328 GTO 00
170 RCL 01	223 RCL 05	276 X=Y?	329 ST+ X
171 INT	224 E	277 PROMPT	330 STO 02
172 X#Y?	225 -	278 "FROM "	331 XEQ 06
173 GTO 00	226 X<0?	279 ARCL 10	332 2
174 2	227 RTN	280 ARCL 11	333 STO 02
175 ST- 06	228 XEQ 13	281*LBL 39	334 RCL 03
176*LBL 00	229 X<0?	282 STO 09	335 XEQ 06
177 RCL 01	230 GTO 05	283 `` TO "	336*LBL 00
178 INT	231 RCL 04	284 ARCL 12	337 RCL 16
179 X=0?	232 RCL 08	285 ARCL 13	338 -2
180 ISG 06	233 -	286 CLAXON	339 X#Y?
181 X=0?	234 STO 18	287 PROMPT	340 GTO 00
182 GTO 03	235 X<0?	288 RCL 13	341 STO 02
183 7	236 RTN	289 X=0?	342 XEQ 06
184 RCL 04	237 7	290 GTO 07	343 2
185 X=Y?	238 X<Y?	291 RCL 10	344 XEQ 06
186 ISG 06	239 RTN	292 RCL 11	345 2
187*LBL 03	240 RCL 05	293 XEQ 13	346 STO 02
188 - E	241 E	294*LBL 40	347 CHS
189 E	242 -	295 STO 16	348 XEQ 06
190 XEQ 04	243 STO 07	296 RCL 12	349 2
191 E	244 X>Y?	297 RCL 13	350 XEQ 06
192 XEQ 04	245 RTN	298 XEQ 12	351*LBL 00
193 CLX	246 RCL IND 19	299 .	352 RCL 09
194 X<> 06	247 X<=0?	300 RCL 10	353 -99
195 RCL 09	248 RTN	301 RCL 11	354 X=Y?
196 X=Y?	249 RCL 18	302 XEQ 12	355 GTO 45
197 RTN	250 RCL 07	303 RCL 10	356 CLA
198 X>Y?	251 XEQ 13	304 RCL 12	357 GTO 39
199 RTN	252 X=0?	305 -	358*LBL 07
200 X<>Y	253 GTO 00	306 ABS	359 -2
201 STO 09	254 RCL 00	307 2	360 GTO 40

361*LBL 06	412 FRC	463 INT	08 STO 02
362 STO 03	413 E1	464 LASTX	09 79
363 RCL 00	414 *	465 FRC	10 STO 01
364 INT	415 STO 15	466 E1	11 16
365 RCL 02	416 "TO?"	467 *	12 STO 00
366 +	417 PROMPT	468 X<> 03	13 48.055
367 STO 04	418 E1	469 STO 15	14 CF 12
368 RCL 01	419 /	470 X<>Y	15 25
369 INT	420 INT	471 X<> 02	16 GTO 00
370 RCL 03	421 STO 00	472 STO 14	17*LBL 01
371 +	422 STO 02	473 X<>Y	18 11
372 STO 05	423 LASTX	474 GTO 08	19*LBL 00
373 X<0?	424 FRC	475*LBL 00	20 SKPCOL
374 RTN	425 E1	476 RCL 03	21 X<>Y
375 7	426 *	477 7	22 ACCHR
376 X<Y?	427 STO 03	478 X#Y?	23 ISG X
377 RTN	428 STO 01	479 GTO 20	24 GTO 01
378 RCL 04	429 RCL 14	480 2	25 PRBUF
379 X>Y?	430 RCL 15	481 RCL 02	26 27.08308
380 RTN	431*LBL 08	482 RCL 03	27 STO 05
381 X<0?	432 XEQ 13	483 XEQ 12	28 8
382 RTN	433 RCL 02	484 GTO 20	29*LBL 02
383 RCL 05	434 RCL 03	485*LBL 13	30 E
384 XEQ 13	435 XEQ 12	486 X<>Y	31 -
385 X#0?	436 .	487 8	32 STO 14
386 RTN	437 STO IND 19	488 *	33 ACX
387 RCL 02	438 RCL 14	489 +	34 SF 12
388 2	439 RCL 02	490 20	35 2
389 /	440 -	491 +	36 SKPCOL
390 RCL 00	441 ABS	492 STO 19	37*LBL 10
391 +	442 2	493 RDN	38 2
392 RCL 03	443 X#Y?	494 RCL IND T	39 SKPCOL
393 2	444 GTO 00	495 RTN	40 RCL IND 05
394 /	445 .	496*LBL 12	41 +
395 RCL 01	446 RCL 14	497 X<>Y	42 RCL IND X
396 +	447 RCL 02	498 8	43 ACCHR
397 XEQ 13	448 +	499 *	44 ISG 05
398 X>0?	449 2	500 +	45 GTO 10
399 GTO 28	450 /	501 20	46 PRBUF
400 RTN	451 RCL 15	502 +	47 65
401*LBL 45	452 RCL 03	503 RDN	48 ST- 05
402 FS? 55	453 +	504 STO IND T	49 CF 12
403 XROM "KPRT"	454 2	505 END	50 RCL 14
404*LBL 46	455 /		51 X#0?
405 "FROM?"	456 XEQ 12	01*LBL "KPRT"	52 GTO 02
406 PROMPT	457 "+TO?"	02 ADV	53 ADV
407 E1	458 PROMPT	03 31	54 ADV
408 /	459 X<0?	04 STO 03	55 ADV
409 INT	460 GTO 00	05 35	56 END
410 STO 14	461 E1	06 STO 04	
411 LASTX	462 /	07 45	

Connect Four for HP-41CX

Kai Schröder, http://achim-und-kai.de/kai/hp41cx/vier_gewinnt_e.html

Most people know "Four Wins" by MB - the game with a blue board consisting of seven rows and six columns, in which yellow and red chips are thrown in on the top. The player, who manages to place four of his chips in a row - horizontal, vertical, or diagonal - wins.

For a while I played this game with my friends intensively, and some time I thought by myself, why not write a program for my HP-41CX ? For some months I was devoted most of the time to these program (more than to my studies ;-)) - every free minute I spent writing it -, and it culminated in the repetition of a first diploma exam . . . :-) So you can understand, that this program is something special for me ! ;-)

The program uses the HP-41CX up to the last byte and without synthetic programming techniques it would have never been possible to realize it. The running program consists of two parts, EXMEM and GAME6. First, EXMEM must be loaded.

Attention : Before EXMEM is copied into extended memory, all jumps back have to be performed by hand! The corresponding GTO's are marked with an arrow "<--". By means of GTO.*** you have to jump to the corresponding line, then in run mode execute SST - the program statement is shown in the display until the label is found.

EXMEM must be the first program in extended memory (CAT 4 must show EXMEM as first entry). EXMEM is called by means of synthetic programming methods in extended memory. If you forget a jump by hand the correct label isn't found, and the processor runs into the main program GAME6, what inevitably leads to wrong results or even in an endless loop. After once being executed EXMEM cannot be copied back into main memory.

After EXMEM was copied into extended memory it can be removed from main memory and GAME6 can be loaded. Before starting the game SIZE 106 must be executed.

In normal speed mode the time to compute the next column in which the HP-41CX throws its chip can last up to 20min, normally about 15 min. With TURBO speed the time about halves.

To prevent battery voltage to decrease too much three times during main loop execution flag 49 (battery voltage flag) is checked. In case flag 11 (automatic program start) is set and the HP-41CX powers off itself. On power on "BATTERY" is displayed to indicate low battery voltage. After changing batteries, the game can be continued - all information remains in memory (see further down).

Course of Game:

On starting the game first a seed for the random number generator must be entered. Now the player chooses, whether he or the HP-41CX begins. If the player wants to start, the input is 1, otherwise 0. If the HP-41CX begins the game, the column is displayed, in which it throws its first chip. If the player is to begin, after a sound "INPUT : COLUMN ?" is shown in the display. Now the player enters the column in which he throws his chip and presses R/S.

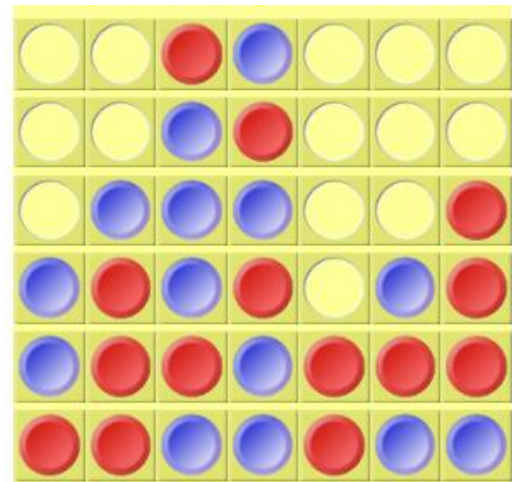
Important : later correction of the entered column is not possible - therefore be careful to enter the correct column ! Now the HP-41CX computes the column in which it throws its chip and after a BEEP displays this. Pressing R/S the prompt for the player appears again.

In case a nonsensical or full column was entered this is recognized and the player has to enter the column once again.

In case the player wins (indeed, this can happen! ;-)) the BEEP sounds and "CONGRATULATE" is displayed. Pressing R/S the rest of the text is shown. If the HP-41CX wins "I AM SORRY," appears after the BEEP. Pressing R/S shows the rest of the text. Then the column is displayed in which the HP-41CX threw its last chip. If the game ends in a draw, after the BEEP "DRAW" is displayed.

In case battery voltage decreases too much during program execution the HP-41CX powers off itself. After changing batteries, the game can be continued. You must pay attention to the following:

- The last input of a column must be cancelled. The corresponding register R01 through R07 (for the seven columns of the board) must be decremented by 1 (the chip has to be "get out").
- Now execute XEQ 16 and the prompt appears. Now you have to enter **the same column** - otherwise it will result in irreparable damages !



Program listing:

<u>01*LBL</u>	24 74.08	47 X<>Y	71 FS?C 09
<u>"4WINS"</u>	25 CLRGX	48 RCL IND X	72 GTO 12
02 106	26 E	49 X<>Y	73 SF 04
03 PSIZE	27 TONE 8	50 STO [74 GTO 37
04 " 4 WINS"	28 CF 22	51 FS?C 02	<u>75*LBL "M"</u>
05 AVIEW	29	52 STO 00	76 FC?C 07
06 CLRG	"^COLUMN=?"	53 2	77 GTO 00
07 SF 00	30 PROMPT	54 +	78 RCL [
08 GTO 37	31 FC? 22	55 X<>Y	79 7
<u>09*LBL "L"</u>	32 GTO 16	56 7	80 X=Y?
10 FIX 0	33 INT	57 *	81 GTO 01
11 CF 29	34 RCL IND X	58 +	82*LBL 00
12 XEQ 15	35 6	59 STO 53	83 FC?C 08
13 12	36 X=Y?	60 E	84 GTO 00
14 X<>F	37 GTO 39	61 8	85 RCL [
15 R^	38 R^	62 XEQ 18	86 E
16 R^	39 R^	63 10.051	87 X#Y?
17 X#0?	40 X<Y?	64 STO 53	88 GTO 00
18 GTO 00	41 GTO 39	65 FS? 05	89*LBL 01
19 SF 06	42 7	66 GTO 35	90 E
20 GTO 17	43 X<Y?	67 FC? 03	91 XEQ 15
21*LBL 00	44 GTO 39	68 GTO 11	92 E2
22 SF 05	45 ISG IND Y	69 FS?C 06	93 *
23*LBL 16	46 ""	70 GTO 01	94 INT

95 3	152 ISG 03	209 X<Y?	265 GTO 01
96 MOD	153 ""	210 X<>Y	266 5
97 X=0?	154 3	211 RCL 03	267 +
98 GTO 40	155 GTO 00	212 X<Y?	268 STO IND
99 X=Y?	156*LBL 01	213 X<>Y	55
100 GTO 41	157 4	214 RCL 04	269 DSE IND
101 GTO 12	158 RCL 58	215 X<Y?	56
102*LBL 00	159 X#Y?	216 X<>Y	270 ""
103 5	160 GTO 12	217 RCL 05	271 GTO 00
104 RCL 00	161 RCL [218 X<Y?	272*LBL 01
105 X=Y?	162 X=Y?	219 X<>Y	273 2
106 GTO 02	163 GTO 03	220 RCL 06	274 -
107 3	164 ISG Y	221 X<Y?	275 STO IND
108 X=Y?	165 ""	222 X<>Y	55
109 GTO 03	166 X=Y?	223 RCL 07	276*LBL 00
110 X<>Y	167 GTO 02	224 X<Y?	277 RCL IND
111 4	168 X>Y?	225 X<>Y	55
112 X>Y?	169 GTO 01	226 2	278 RCL IND X
113 GTO 04	170 2	227 +	279 RCL IND
114 RCL [171 ST- Z	228 7	56
115 X<Y?	172 RDN	229 *	280 X>Y?
116 GTO 12	173 X=Y?	230 2	281 GTO 00
117 5	174 GTO 01	231 +	282 8
118 X=Y?	175 GTO 02	232 STO 62	283 XEQ 19
119 GTO 05	176*LBL 03	233*LBL 22	284 X=0?
120 GTO 06	177 4	234 51	285 GTO 01
121*LBL 02	178 RCL [235 RCL 53	286 2
122 RCL [179 X<=Y?	236 INT	287 GTO 02
123 X#Y?	180 GTO 02	237 X>Y?	288*LBL 01
124 GTO 05	181*LBL 01	238 GTO 37	289 E
125 GTO 12	182 RCL 58	239 RCL 62	290 GTO 02
126*LBL 03	183 DSE X	240 X<Y?	291*LBL 00
127 RCL [184 ""	241 GTO 37	292 .
128 X#Y?	185 ISG IND X	242 FS? 49	293*LBL 02
129 GTO 06	186 ""	243 GTO 38	294 STO IND
130 GTO 12	187 GTO 00	244*LBL 20	61
131*LBL 04	188*LBL 02	245 RCL 52	295 GTO IND
132 RCL [189 RCL 58	246 INT	52
133 X#Y?	190 ISG X	247 XEQ 19	296*LBL 01
134 GTO 05	191 ""	248 X=0?	297 DSE 53
135 3	192 ISG IND X	249 GTO 21	298 ""
136 X=Y?	193 ""	250*LBL 23	299 ISG 61
137 GTO 06	194 GTO 00	251 2	300 GTO 12
138 X>Y?	195*LBL 12	252 RCL 53	301 4
139 GTO 05	196 ISG 04	253 INT	302 ST+ 53
140 RCL [197 ""	254 7	303 GTO 11
141 5	198 4	255 /	304*LBL 02
142 X#Y?	199*LBL 00	256 INT	305 6
143 GTO 12	200 R^	257 STO IND	306 ST+ 53
144*LBL 05	201 SF 05	56	307 ISG 61
145*LBL 40	202 CF 03	258 7	308 GTO 12
146 ISG 05	203 GTO 17	259 *	309 24
147 ""	204*LBL 11	260 RCL 53	310 ST- 53
148 5	205 .	261 INT	311 GTO 11
149 GTO 00	206 STO 00	262 X<>Y	312*LBL 03
150*LBL 06	207 RCL 01	263 -	313 7
151*LBL 41	208 RCL 02	264 X>Y?	314 ST+ 53

315 ISG 61	371 X=Y?	428*LBL 01	485 RCL 58
316 GTO 12	372 GTO 01	429 RCL 63	486 X=0?
317 28	373 ST+ 64	430 X=0?	487 GTO 01
318 ST- 53	374 GTO 02	431 GTO 09	488 RCL 59
319 GTO 11	375*LBL 00	432 GTO 25	489 X=0?
320*LBL 04	376 ISG 65	433*LBL 02	490 GTO 02
321 8	377 ""	434 RCL 65	491 RCL 69
322 ST+ 53	378 GTO 02	435 2	492 STO 82
323 ISG 61	379*LBL 01	436 X#Y?	493 RCL 73
324 GTO 12	380 ISG 63	437 GTO 10	494 RCL IND
325 32	381 ""	438 SF 10	69
326 ST- 53	382*LBL 02	439 GTO 27	495 GTO 03
327 GTO 11	383 ISG 61	440*LBL 03	496*LBL 00
328*LBL 05	384 GTO 08	441 X=0?	497 RCL 66
329 ISG 53	385 4	442 GTO 10	498 STO 82
330 ""	386 ST- 61	443 RCL 65	499 RCL 70
331 ISG 61	387 RCL 65	444 3	500 RCL IND
332 GTO 12	388 X=Y?	445 X#Y?	66
333 4	389 GTO 21	446 GTO 10	501 GTO 03
334 ST- 53	390 4	447 2	502*LBL 01
335 GTO 11	391 RCL 64	448 STO 00	503 RCL 67
336*LBL 06	392 X=Y?	449 GTO 27	504 STO 82
337 6	393 GTO 26	450*LBL 31	505 RCL 71
338 ST- 53	394 RCL 63	451 -	506 RCL IND
339 ISG 61	395 +	452 E	67
340 GTO 12	396 4	453 X#Y?	507 GTO 03
341 24	397 X=Y?	454 GTO 21	508*LBL 02
342 ST+ 53	398 GTO 25	455 XEQ 36	509 RCL 68
343 GTO 11	399 RCL 63	456 3	510 STO 82
344*LBL 07	400 3	457 ST+ IND Y	511 RCL 72
345 8	401 X=Y?	458 GTO 21	512 RCL IND
346 ST- 53	402 GTO 27	459*LBL 30	68
347 ISG 61	403 RCL 64	460 RDN	513*LBL 03
348 GTO 12	404 X=Y?	461 -	514 RCL 00
349 32	405 GTO 27	462 E	515 X#0?
350 ST+ 53	406 E	463 X#Y?	516 GTO 30
351 GTO 11	407 RCL 63	464 GTO 00	517 RDN
352*LBL 12	408 X>Y?	465 RCL 00	518 FS?C 10
353 ISG 55	409 GTO 02	466 E	519 GTO 31
354 ISG 56	410 GTO 03	467 X#Y?	520 -
355 GTO 23	411*LBL 10	468 GTO 01	521 E
356*LBL 11	412 RCL 65	469 XEQ 36	522 X=Y?
357 FS? 49	413 X>0?	470 5	523 GTO 00
358 GTO 38	414 GTO 01	471 ST+ IND Y	524 X<>Y
359 4	415*LBL 09	472 GTO 00	525 2
360 ST- 61	416 2	473*LBL 01	526 /
361 3	417 RCL 64	474 XEQ 36	527 FRC
362 ST- 55	418 X#Y?	475 ISG IND X	528 X#0?
363 ST- 56	419 GTO 21	476 ""	529 GTO 21
364 63.065	420 RCL 65	477*LBL 00	530 RCL 82
365 CLRGX	421 +	478 .	531 73
366*LBL 08	422 4	479 STO 00	532 +
367 RCL IND	423 X#Y?	480 GTO 21	533 RCL 63
61	424 GTO 21	481*LBL 27	534 3
368 X=0?	425 E	482 RCL 57	535 X=Y?
369 GTO 00	426 STO 00	483 X=0?	536 GTO 01
370 E	427 GTO 27	484 GTO 00	537 2

538 GTO 02	594 FS?C 04	650 STOP	704 GTO 20
539*LBL 01	595 GTO 01	651 RCL 01	705 1.007
540 E	596 RCL 08	652 RCL 02	706 STO 52
541*LBL 02	597 E6	653 X>Y?	707 ISG 53
542 STO IND T	598 *	654 X<>Y	708 GTO 22
543 GTO 21	599 INT	655 RCL 03	709 GTO 37
544*LBL 00	600 E1	656 X>Y?	710*LBL 15
545 RCL 63	601 MOD	657 X<>Y	711 RCL 08
546 3	602 X=0?	658 RCL 04	712 E^X
547 X=Y?	603 GTO 00	659 X>Y?	713 FRC
548 GTO 00	604 7	660 X<>Y	714 R-D
549 XEQ 36	605 X<Y?	661 RCL 05	715 FRC
550 30	606 GTO 00	662 X>Y?	716 STO 08
551 ST+ IND Y	607 X<>Y	663 X<>Y	717 RTN
552 GTO 21	608 73	664 RCL 06	718*LBL 35
553*LBL 36	609 +	665 X>Y?	719 RCL 04
554 RCL 82	610 RCL IND X	666 X<>Y	720 X>0?
555 84	611 X#0?	667 RCL 07	721 GTO 00
556 +	612 GTO "K"	668 X>Y?	722 4
557 RTN	613 X<>Y	669 X<>Y	723 1
558*LBL 00	614 73	670 6	724 STO 04
559 SF 01	615 -	671 X=Y?	725 GTO 17
560*LBL 37	616 RCL IND X	672 GTO 33	726*LBL 00
561 FS? 49	617 6	673 GTO 16	727 SF 09
562 GTO 38	618 X=Y?	674*LBL 19	728 XEQ 15
563 FS? 04	619 GTO "K"	675 DSE X	729 E5
564 GTO 00	620 ISG IND Z	676 ""	730 *
565 E	621 ""	677 RCL 54	731 INT
566 CHS	622 RDN	678 X<>Y	732 2
567 STO 00	623 FC? 06	679 Y^X	733 MOD
568*LBL 00	624 GTO 01	680 RCL IND	734 X=0?
569 GTO "XM"	625 RDN	53	735 GTO 00
570*LBL 38	626 STO 58	681 X<>Y	736 5
571 "LOW BAT"	627 R^	682 ST/ Y	737 1
572 SF 11	628 GTO 01	683 X<>Y	738 STO 05
573 OFF	629*LBL 00	684 INT	739 GTO 17
574 AVIEW	630 RCL 04	685 RCL 54	740*LBL 00
575 STOP	631 6	686 MOD	741 3
576*LBL 39	632 X=Y?	687 RTN	742 1
577 TONE 3	633 GTO "K"	688*LBL 18	743 STO 03
578 "BAD	634 RCL 77	689 XEQ 19	744 GTO 17
INPUT"	635 X#0?	690 X<>Y	745*LBL 32
579 AVIEW	636 GTO "K"	691 ST* Z	746 97.1
580 PSE	637 ISG 04	692 *	747 STO 81
581 GTO 16	638 ""	693 ST- IND	748 CLRGX
582*LBL "N"	639 4	53	749 CF 07
583 GTO 01	640 FS? 06	694 X<>Y	750 CF 08
584*LBL "K"	641 STO 58	695 ST+ IND	751 85.091
585 ISG 83	642 R^	53	752 STO 92
586 GTO 00	643*LBL 01	696 RTN	753 CLRGX
587 GTO 32	644 BEEP	697*LBL 25	754 93.096
588*LBL 00	645 "COLUMN:	698 .	755 STO 84
589 XEQ 15	"	699 RCL 52	756 74.08
590*LBL 17	646 ARCL Y	700 INT	757 STO [
591 FS?C 05	647 AVIEW	701 XEQ 18	758 1.007
592 GTO 01	648 FS?C 10	702*LBL 21	759 STO \
593*LBL "O"	649 GTO 34	703 ISG 52	760 STO 52

761*LBL 13	02 FS?C 00	51 ISG 81	102 STO \
762 RCL IND [03 GTO 17	52 GTO 01	103*LBL 30
763 2	04 FS?C 01	53*LBL 28	104 RCL [
764 X=Y?	05 GTO 15	54 RCL IND	105 RCL IND \
765 GTO 00	06 FS?C 04	97	106 X#Y?
766 RCL [07 GTO 19	55 STO 01	107 GTO 02
767 INT	08 RCL 85	56 RCL IND	108 RCL \
768 73	09 RCL 86	98	109 4
769 -	10 RCL 87	57 STO A	110 -
770 RCL IND X	11 RCL 88	58 RCL IND	111 RCL IND X
771 6	12 XROM	99	112*LBL 11
772 X=Y?	"SRT1"	59 STO B	113 E
773 GTO 00	13 RCL 89	60 RCL IND	114 ST+ IND Y
774 ISG IND Z	14 XROM	00	115 97.1
775 ""	"SRT1"	61 STO C	116 STO 81
776 RDN	15 RCL 90	62 RCL 01	117 CLRGX
777 SF 05	16 XROM	63 RCL A	118*LBL 29
778 GTO 17	"SRT1"	64 RCL B	119 CF 07
779*LBL 00	17 RCL 91	65 X#Y?	120 CF 08
780 ISG [18 XROM	66 GTO 12	121 1.007
781 GTO 13	"SRT1"	67 R^	122 STO 52
782*LBL 14	19 STO 93	68 X#Y?	123 R^
783 RCL IND \	20 RDN	69 GTO 12	124 85.091
784 6	21 STO 94	70 R^	125 STO 92
785 X=Y?	22 RDN	71 X#Y?	126 CLRGX
786 GTO 01	23 STO 95	72 GTO 12	127 93.096
787 ISG IND \	24 RDN	73 R^	128 STO 84
788 ""	25 STO 96	74 X#Y?	129 RDN
789 RCL \	26*LBL 16	75 GTO 12	130 GTO "N"
790 X<>Y	27 RCL IND	76 4.005	131*LBL 00
791 SF 05	84	77 STO D	132 2
792 GTO 17	28 X=0?	78*LBL 03	133 X#Y?
793*LBL 01	29 GTO "K"	79 97.1	134 GTO 00
794 ISG \	30 RCL IND	80 STO 81	135 29
795 GTO 14	92	81*LBL 10	136 RCL IND
796*LBL 33	31 X=Y?	82 RCL D	84
797 BEEP	32 GTO 00	83 INT	137 X>Y?
798 " DRAW"	33 ISG 92	84 RCL IND	138 GTO 31
799 AVIEW	34 GTO 16	81	139 GTO 07
800 GTO 34	35 FS? 07	85 X=Y?	140*LBL 00
801*LBL 26	36 GTO 28	86 GTO 11	141 20
802 BEEP	37 FS? 08	87 ISG 81	142 RCL IND
803	38 GTO 07	88 GTO 10	84
"CONGRATS"	39*LBL 00	89 RCL D	143 X>Y?
804 AVIEW	40 11	90 3	144 GTO 31
805 PSE	41 ST- 92	91 X=Y?	145 11
806 "YOU WON"	42 RCL IND	92 GTO 12	146 ST+ 92
807 AVIEW	92	93 ISG D	147 SF 08
808*LBL 34	43 X>0?	94 GTO 03	148 ISG 92
809 RCL 09	44 GTO 00	95 3	149 GTO 16
10 STOF LAG	45*LBL 31	96 STO D	150*LBL 07
811 CLA	46 73	97 GTO 03	151 FS? 07
812 CLST	47 ST- 92	98*LBL 12	152 GTO 28
813 CLRG	48 RCL 92	99 XROM	153 85.091
814 END	49 INT	"SRT1"	154 STO 92
	50 STO IND	100 STO [155 ISG 84
<u>01*LBL "XM"</u>	81	101 101	156 GTO 16

157 GTO "K"	212*LBL 00	267*LBL 27	321 STO 38
158*LBL 01	213 XEQ 20	268 ISG 07	322 STO 39
159 84	214 X<=Y?	269 ""	323 STO 40
160 ST+ 92	215 GTO 23	270 7	324 STO 45
161 SF 07	216 3	271*LBL 00	325 STO 46
162 ISG 92	217 X<>Y	272 X<>Y	326 STO 47
163 GTO 16	218 X<=Y?	273 SF 04	327 117
164 GTO 28	219 GTO 24	274 GTO "O"	328 STO 10
165*LBL 02	220 5	275*LBL 20	329 STO 11
166 ISG \	221 X<>Y	276 RCL 08	330 STO 12
167 ""	222 X<=Y?	277 E^X	331 STO 17
168 GTO 30	223 GTO 26	278 FRC	332 STO 18
169*LBL 19	224 GTO 27	279 STO 08	333 STO 19
170 E	225*LBL 01	280 E6	334 STO 24
171 RCL 00	226 XEQ 20	281 *	335 STO 25
172 X=Y?	227 X<=Y?	282 RTN	336 STO 26
173 GTO 00	228 GTO 24	283*LBL 15	337 1054
174 ISG Y	229 3	284 BEEP	338 STO 34
175 ""	230 X<>Y	285 "SORRY,	339 STO 41
176 X=Y?	231 X<=Y?	YOU LOST"	340 STO 48
177 GTO 01	232 GTO 25	286 AVIEW	341 121
178 6	233 5	287 PSE	342 STO 13
179 X=Y?	234 X<>Y	288 RCL 82	343 STO 20
180 GTO 02	235 X<=Y?	289 SF 10	344 STO 27
181 ISG X	236 GTO 22	290 R^	345 1.007
182 ""	237*LBL 21	291 R^	346 STO 52
183 X=Y?	238 ISG 01	292 PSE	347 3
184 GTO 05	239 ""	293 GTO 29	348 STO 54
185 GTO "M"	240 E	294*LBL 17	349 57.06
186*LBL 00	241 GTO 00	295 RCLFLAG	350 STO 61
187 RCL [242*LBL 22	296 STO 09	351 66.069
188 6	243 ISG 02	297 730	352 STO 55
189 X=Y?	244 ""	298 STO 35	353 70.073
190 GTO 00	245 2	299 STO 36	354 STO 56
191 SF 07	246 GTO 00	300 STO 37	355 .02
192 GTO "M"	247*LBL 23	301 STO 42	356 STO 83
193*LBL 01	248 ISG 03	302 STO 43	357 93.096
194 RCL [249 ""	303 STO 44	358 STO 84
195 7	250 3	304 STO 49	359 85.091
196 X=Y?	251 GTO 00	305 STO 50	360 STO 92
197 GTO 01	252*LBL 24	306 STO 51	361 97.1
198 GTO "M"	253 ISG 04	307 13	362 STO 81
199*LBL 02	254 ""	308 STO 14	363 FRC
200 RCL [255 4	309 STO 15	364 "SEED=?"
201 E	256 GTO 00	310 STO 16	365 PROMPT
202 X=Y?	257*LBL 25	311 STO 21	366 STO 08
203 GTO 00	258 ISG 05	312 STO 22	367 "START?"
204 GTO "M"	259 ""	313 STO 23	HP=0"
205*LBL 05	260 5	314 STO 28	368 AVIEW
206 RCL [261 GTO 00	315 STO 29	369 CLX
207 2	262*LBL 26	316 STO 30	370 STOP
208 X=Y?	263 ISG 06	317 324	371 GTO "L"
209 GTO 01	264 ""	318 STO 31	372 END
210 SF 08	265 6	319 STO 32	
211 GTO "M"	266 GTO 00	320 STO 33	

Wari for the HP-41C/CV/CX

<https://www.hpmuseum.org/software/41td/wari.htm>

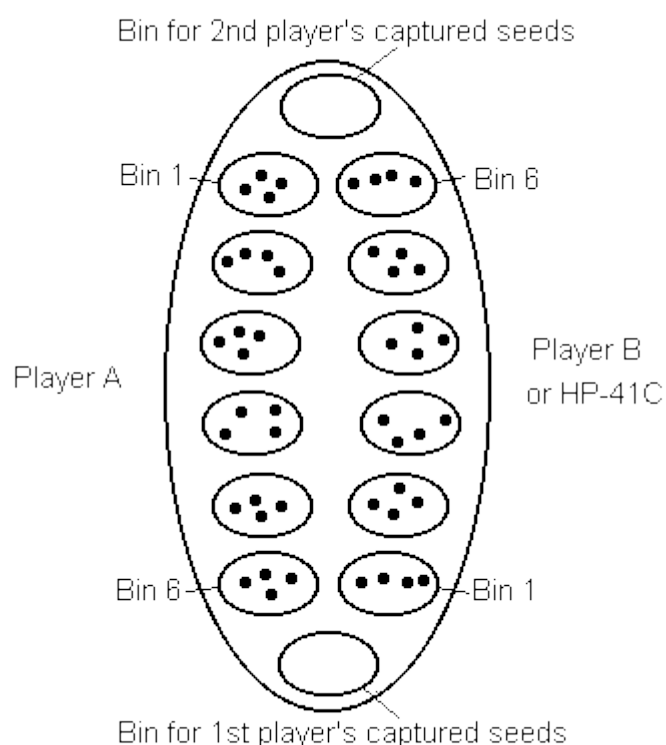
This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

Wari* is a board game which has been played for at least several centuries in various forms throughout Africa. The game is played on a board containing (generally) twelve small pits or bins, and two large pits. Forty-eight beads, seeds, or other counters are moved and captured according to certain rules.

The Wari board shown here is set up to begin a game.



Each player in turn removes all the counters from one bin on his side and distributes them one-at-a-time into successive bins moving counterclockwise, skipping the two bins which are for storing captured counters. If the last counter drops into an opponent's hole containing one or two counters, the contents of that hole are captured and placed in the player's scoring pit. Counters in an unbroken sequence of two- and three-counter bins on the opponent's side

clockwise from the captured bin are also captured. If a bin contains twelve counters or more, that bin is skipped when the counters from that bin are distributed.

The above rules are implemented in the calculator program. Special rules, such as prohibiting moves which remove all of the opponent's counters, were deemed to be variations of the basic game and were not programmed. It is possible to come to a situation where a few counters will circulate forever. In this case each player claims the counters on his side.

To make a play on the calculator Wari board, the player specifies the bin he wants to move by keying in a number from 1 to 6 and then pushing either [A] or [B] (for player A or B). The machine then moves the counters from the specified bin according to the rules. To play against the calculator, signal to the calculator to move by pressing [C]. The calculator will then move player B's counters.

When one of the sites of the board is displayed, as designated by leading A or B, it is as if you moved around to that side of the board. In other words, bin 1 for either players side is always to the left and counterclockwise always to the right. If you are looking at side A, [R/S] will get you side B. If you are looking at side B, [R/S] will get you the score. If you are looking at the score, [R/S] will get you side A.

*Also known as Man-Kalah, Awari, and many other names.

Note: Requires 1 Memory Module on HP-41C

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] WARI	SEED?
3	Key in number between 0 and 1	seed	[R/S]	A 4,4,4,4,4,4
4	To see the other side of the board and/or the score, press [R/S]. This can be done as often as desired.		[R/S]	A (board) or B (board) or A= or B=
5	Players A, B or the 41C moving B's counters can move in any order with anyone starting the game.			
	Player A	bin #	[A]	A (board)
	Player A	bin #	[B]	A (board)
	HP-41C (moving for B)		[C]	A (board)
6	Go to step 4 or 5 as desired.			
7	To start a new game press [E] and go to step 4 or 5 as desired.		[E]	A 4,4,4,4,4,4

Example

Keystrokes:	Display:
[XEQ] [ALPHA]	
SIZE [ALPHA] 018	
[XEQ] [ALPHA]	
WARI [ALPHA]	SEED?


```
.9977663333 [R/S]      A 4,4,4,4,4,4
[R/S]                  B 4,4,4,4,4,4
[R/S]                  A=0, B=0
[C]                    A 5,5,4,4,4,4
[R/S]                  B 4,4,4,0,5,5
6 [A]                  A 5,5,4,4,4,0
[R/S]                  B 5,5,5,1,5,5
[C]                    A 6,6,4,4,4,0
[R/S]                  B 5,5,0,2,6,6
2 [A]                  A 6,0,5,5,5,1
[R/S]                  B 6,6,0,2,6,6
[C]                    A 7,1,6,6,6,0
[R/S]                  B 6,6,0,2,6,0
[R/S]                  A=0, B=2
.
.
.
```

Program listing:

01 LBL "WARI"	40 STO T	79 GTO 00
02 SF 27	41 +	80 RCL 15
03 CF 29	42 +	81 INT
04 FIX 00	43 12	82 GTO A
05 "SEED?"	44 MOD	83 LBL 00
06 PROMPT	45 X=0?	84 6
07 STO 00	46 X<> L	85 STO 16
08 LBL E	47 7	86 0
09 1.012	48 X<=Y?	87 STO 15
10 CLRG	49 GTO 00	88 LBL 04
11 4	50 X<>Y	89 RCL 16
12 LBL 06	51 R^	90 RCL IND 16
13 STO IND Y	52 CHS	91 X=0?
14 ISG Y	53 RCL IND Y	92 GTO 00
15 GTO 06	54 X<=Y?	93 RCL X
16 GTO 50	55 GTO 00	94 12
17 LBL C	56 X<>Y	95 /
18 CF 05	57 3	96 INT
19 0	58 +	97 X>0?
20 STO 17	59 X<=Y?	98 GTO 00
21 STO 15	60 GTO 00	99 RDN
22 2	61 SF 07	100 +
23 XEQ 51	62 RCL Z	101 12
24 CF 06	63 RCL 17	102 MOD
25 X=0?	64 X>Y?	103 X=0?
26 SF 06	65 GTO 00	104 LASTX
27 CF 07	66 X#Y?	105 7
28 7.012	67 GTO 01	106 X>Y?
29 STO 16	68 FS? 06	107 GTO 00
30 LBL 05	69 GTO 00	108 RCL IND Y
31 RCL 16	70 LBL 01	109 X=0?
32 INT	71 RDN	110 GTO 00
33 RCL IND 16	72 STO 17	111 3
34 X=0?	73 RCL 16	112 X<=Y?
35 GTO 00	74 STO 15	113 GTO 00
36 RCL X	75 LBL 00	114 SF 07
37 12	76 ISG 16	115 R^
38 /	77 GTO 05	116 RCL 15
39 INT	78 FC?C 07	117 X<=Y?

118 X<>Y	165 LBL A	212 FC? 05
119 STO 15	166 STO 15	213 ST+ 14
120 LBL 00	167 STO 17	214 ST- IND 17
121 DSE 16	168 RCL IND 15	215 RCL 17
122 GTO 04	169 X=0?	216 1
123 FC?C 07	170 GTO 50	217 -
124 GTO 00	171 STO 16	218 12
125 RCL 15	172 ST- IND 15	219 MOD
126 GTO A	173 LBL 08	220 X=0?
127 LBL 00	174 RCL 17	221 LASTX
128 6	175 1	222 STO 17
129 XEQ 51	176 +	223 GTO 07
130 1	177 12	224 LBL 50
131 +	178 MOD	225 SF 05
132 STO 15	179 X=0?	226 "A "
133 STO 16	180 LASTX	227 1.006
134 LBL 03	181 STO 17	228 XEQ 00
135 6	182 RCL 15	229 PROMPT
136 +	183 X=Y?	230 "B "
137 RCL IND X	184 GTO 08	231 7.012
138 X#0?	185 1	232 XEQ 00
139 GTO 00	186 ST+ IND 17	233 PROMPT
140 RCL 15	187 DSE 16	234 "A="
141 1	188 GTO 08	235 ARCL 13
142 -	189 LBL 07	236 "↑, B="
143 6	190 RCL 17	237 ARCL 14
144 MOD	191 7	238 PROMPT
145 X=0?	192 FS? 05	239 GTO 50
146 LASTX	193 GTO 01	240 LBL 09
147 RCL 16	194 X<=Y?	241 "↑, "
148 X=Y?	195 GTO 50	242 LBL 00
149 GTO 01	196 GTO 00	243 ARCL IND X
150 RDN	197 LBL 01	244 ISG X
151 STO 15	198 X>Y?	245 GTO 09
152 GTO 03	199 GTO 50	246 RTN
153 LBL 01	200 LBL 00	247 LBL 51
154 SF 05	201 RCL IND 17	248 RCL 00
155 "NO MOVE"	202 2	249 9821
156 AVIEW	203 X>Y?	250 *
157 STOP	204 GTO 50	251 .211327
158 GTO 50	205 RDN	252 +
159 LBL 00	206 4	253 FRC
160 RCL 15	207 X<=Y?	254 STO 00
161 LBL B	208 GTO 50	255 *
162 6	209 RDN	256 INT
163 +	210 FS? 05	257 RTN
164 CF 05	211 ST+ 13	258 END

Mancala for the HP-41CX

[brianddk-https://www.hpmuseum.org/forum/thread-6203.html?highlight=mancala](https://www.hpmuseum.org/forum/thread-6203.html?highlight=mancala)

This is a mancala program I wrote for the hp42s (Free42). For those not familiar with mancala, its a very achient game that has become repopularized over the last few years in the US. You can now get a mancala board at most stores that sell board games. This particular 'flavor' of mancala kahla(6,4), but it seemed to be the one that is most available in my area. This is a fun game to play, but most mathmatcal analysis of possible permutations make this game significantly harder for Player 2. Whoever goes first is likely to win.

Game Display

The game display will show a number between 1-million and 2-million, for both Player 1 and Player 2. The fractional part is the player's score (0.10 = 10 points). The number in the million'th place is purely for alignment and should be ignored, the other numbers represent your 6 'pits'. The 'pit' to the far left is 'pit 1' the pit to the far right is 'pit 6'. To move, you specify a pit number to move. **Note:** 41-C users will need to swap (X<>Y) to see the P2 score and pit display.

```
; Game Display
;
; x: Z,DCB,A98.P2
; y: Z,123,456.P1
;
; Where,
; 'Z,' - Ignore the 'millionth' place, its a place holder, nothing
more
; 'P1' - The score for Player 1 (in the X vector)
; 'P2' - The score for Player 2 (in the Y vector)
; '1|D' - # of beans in 'pit #1' for P1 and P2
; '2|C' - # of beans in 'pit #2' for P1 and P2
; '3|B' - # of beans in 'pit #3' for P1 and P2
; '4|A' - # of beans in 'pit #4' for P1 and P2
; '5|9' - # of beans in 'pit #5' for P1 and P2
; '6|8' - # of beans in 'pit #6' for P1 and P2
;
; Indicators
; 'GRAD' - Player1's turn when 'GRAD' is displayed
; 'RAD' - Player2's turn when 'RAD' is displayed
```

Gameplay

To start the game, simply `XEQ` the `MANCA` program. The game will show the initial board and set the indicator for Player 1 to take his turn. Player 1 can then study the board and pick a pit to move. Thier pick is given by placing the pick in the level 1 (x) on the stack then hit `run` (aka `R/S`). The game will then move the beans according to the rules and redisplay the board. It is now time for the next move. The `GRAD` / `RAD` indicator will light to instruct the players as to whos turn it is. Keep in mind, earning extra turns is a key strategy of the game.

Program listing:

<u>01*LBL "MANCA"</u>	58 GTO 27	115 RCL IND 16
02 XEQ 07	59*LBL 17	116 STOP
03*LBL 09	60 "PLAYER2 WON!"	117 RTN
04 XEQ 02	61*LBL 27	118*LBL 18
05 FS? 03	62 SF 03	119 CF 04
06 GTO 05	63 PROMPT	120 INT
07*LBL 20	64*LBL 28	121 E
08 XEQ 04	65 RTN	122 X<>Y
09 XEQ 18	66*LBL 04	123 X<Y?
10 FS? 04	67 1.006	124 SF 04
11 GTO 20	68 STO 16	125 6
12 XEQ 10	69 14	126 X<>Y
13 XEQ 23	70 STO 17	127 X>Y?
14 GTO 09	71 E6	128 SF 04
15*LBL 05	72 STO IND 17	129 FS? 04
16 XEQ 03	73*LBL 12	130 GTO 19
17 RTN	74 E1	131 STO 16
18*LBL 07	75 6	132 FS? 01
19 CF 01	76 RCL 16	133 GTO 01
20 CF 02	77 INT	134 14
21 CF 03	78 -	135 X<>Y
22 CF 04	79 Y^X	136 -
23 13	80 RCL IND 16	137 STO 16
24 STO 16	81 *	138*LBL 01
25 4	82 ST+ IND 17	139 RCL IND 16
26*LBL 08	83 ISG 16	140 X=0?
27 STO IND 16	84 GTO 12	141 SF 04
28 DSE 16	85 15	142*LBL 19
29 GTO 08	86 STO 17	143 RCL 16
30 0	87 E6	144 RTN
31 STO IND 16	88 STO IND 17	145*LBL 10
32 7	89 13.007	146 0
33 STO 16	90 STO 16	147 X<> IND 16
34 X<>Y	91*LBL 15	148 STO 17
35 STO IND 16	92 E1	149*LBL 11
36 SF 01	93 RCL 16	150 E
37 GRAD	94 INT	151 RCL 16
38 RTN	95 8	152 +
39*LBL 02	96 -	153 14
40 CF 03	97 Y^X	154 MOD
41 0	98 RCL IND 16	155 STO 16
42 STO 17	99 *	156 FS? 01
43 7	100 ST+ IND 17	157 XEQ 21
44 STO 16	101 DSE 16	158 FS? 02
45 24	102 GTO 15	159 XEQ 22
46 RCL IND 16	103 14	160 E
47 X#Y?	104 STO 16	161 ST+ IND 16
48 X>Y?	105 FIX 2	162 DSE 17
49 GTO 14	106 RCL 07	163 GTO 11
50 X<>Y	107 E2	164 E
51 RCL IND 17	108 /	165 RCL IND 16
52 X#Y?	109 ST+ IND 16	166 X=Y?
53 X>Y?	110 RCL 00	167 XEQ 26
54 GTO 17	111 E2	168 RTN
55 GTO 28	112 /	169*LBL 21
56*LBL 14	113 ST+ IND 17	170 X=0?
57 "PLAYER1 WON!"	114 RCL IND 17	171 ISG 16

172 CF 00	195 RTN	218 RTN
173 RTN	196 GTO 06	219 FS? 01
174*LBL 22	197*LBL 16	220 GTO 25
175 7	198 0	221 CF 02
176 X<>Y	199 STO 17	222 SF 01
177 X=Y?	200 RDN	223 GRAD
178 ISG 16	201 X<=Y?	224 GTO 24
179 CF 00	202 RTN	225*LBL 25
180 RTN	203*LBL 06	226 CF 01
181*LBL 26	204 14	227 SF 02
182 7	205 X<>Y	228 RAD
183 RCL 16	206 -	229*LBL 24
184 FS? 01	207 STO 16	230 RTN
185 GTO 13	208 0	231*LBL 03
186 FS? 02	209 X<> IND 16	232 CF 01
187 GTO 16	210 ST+ IND 17	233 CF 02
188 RTN	211 RTN	234 CF 03
189*LBL 13	212*LBL 23	235 CF 04
190 7	213 7	236 FIX 4
191 STO 17	214 RCL 16	237 DEG
192 RDN	215 X=Y?	238 END
193 X#Y?	216 RTN	
194 X>Y?	217 X=0?	

Example turn



The player begins sowing from the highlighted house.



The last seed falls in the store, so the player receives an extra move.



The last seed falls in an empty house on the player's side. The player collects the highlighted seeds from both his house and the opposite house of his opponent and will move them to the store.

Gork, Foxhole Grenades

[Philip T. Frohme - PPCCJ V12N8p11 ; \(August 1985\)](#)

Gork is a program that uses no complex equations or exotic logic to send it's user into a state of incoherent babbling. Although this is a very simple game, much can be learned about the behavior of random numbers by playing it.

Like a Hi-Low game, the user must guess a random number. In this game, however, the random number doesn't stay put. It's movements are also random, although somewhat controlled. I try to program this game into every new machine I encounter as a programming exercise.

SCENARIO: Picture yourself sitting in a foxhole that is too deep to see out off. To your right is a box containing an infinite number of hand grenades. Clyde, your trusty sidekick to your left, is NTB (Not To Bright). Although unable to throw grenades due to a self inflicted injury, he is able to stand and happens to be just tall enough to see out of the foxhole. Advancing toward your foxhole is a Gork (typical fierce looking monster with the usual six arms, scales, fangs, claws, etc.). You have ordered Clyde to tell you where your grenades land in relation to the Gork's location.

The Gork will start on a random point between 0 and 100. You are to try to hit him with a direct lob of your grenades. No matter what you may have heard, close does not count and only aggravates the user. With each missed lob, the Gork will advance toward the foxhole a random number set within the bounds of the user's skill level. If the skill level is 5, the Gork can stay put or move a maximum of 5 yards toward the foxhole.

Sound easy? Clyde will tell you how many yards you missed by but since he is NTB He cannot tell you if your lob was long or short. The grenades also give off a lot of smoke that conceal the Gork's Movements. The user might think he has been lobbing short when the Gork has actually advanced ahead of the impact point.

If the Gork reaches the foxhole before being hit, he eats the user. The Gork in my 41C is well fed. Skill levels range from a sure thing at 0 to whatever the user feels lucky with. A level of 10 is unreachable by many players. Even a level a 3 can be very challenging depending on the starting location of the Gork. The closer the user has missed the Gork, the less likely he/she will be able to accurately estimate the Gork's movements.

HINT: Most players will take random shots at the Gork based on where they think he is. To prevent this "double random" situation, always try to lob the grenades the same distance from the Gork. Using this method, the player's grenade lobs will not aid in the Gor'k's escape.

Philip T. Frohme- August 1985

Program listing:

01*LBL "GORK"	36 XROM "VA"	71 XROM "VA"
02 CLRG	37 PSE	72 BEEP
03 CF 29	38 " BY "	73 RCL 03
04 FIX 0	39 ARCL Y	74 E
05 "SEED?"	40 "` YARDS"	75 -
06 XEQ 02	41 XROM "VA"	76 RCL 02
07 STO 00	42 PSE	77 /
08*LBL 00	43 " AT "	78 E2
09 "SKILL?"	44 ARCL Z	79 *
10 XEQ 02	45 XROM "VA"	80 ST+ 04
11 E	46 .	81 "GORK KILLED"
12 +	47 XROM "RN"	82 AVIEW
13 STO 03	48 RCL 03	83 PSE
14 .	49 *	84 " AT "
15 STO 02	50 INT	85 ARCL 01
16 XROM "RN"	51 RCL 01	86 "` YARDS"
17 E2	52 X<>Y	87 XROM "VA"
18 *	53 -	88 PSE
19 E	54 STO 01	89 GTO 06
20 +	55 .	90*LBL 05
21 RND	56 X<>Y	91 "-GOBBLED UP-"
22 STO 01	57 X<=Y?	92 XROM "VA"
23*LBL 01	58 GTO 05	93 TONE 4
24 E	59 GTO 01	94 2
25 ST+ 02	60*LBL 02	95 ST/ 04
26 "LOB GRENADE"	61 CF 22	96*LBL 06
27 XEQ 02	62 XROM "VA"	97 " IN "
28 STO Y	63 TONE 9	98 ARCL 02
29 RCL 01	64*LBL 03	99 "` TRIES"
30 -	65 PSE	100 XROM "VA"
31 ABS	66 FC?C 22	101 PSE
32 .	67 GTO 03	102 GTO 00
33 X=Y?	68 RTN	103 END
34 GTO 04	69*LBL 04	
35 " YOU MISSED"	70 "*DIRECT HIT*"	

Ed's Note: "RN" and "VA" are the PPC ROM routines of the same name for Random Numbers and non-Stop View Alpha.

TARGET, War Games

Mark Gessner, PPCCJ V11N9 p38; (Nov.Dec 1884)

Here's a game which pits your gut reactions against the computer's cold calculations.

You are responsible for a large cannon, set along high dunes on a scorched desert. Some distance away is the enemy, with a weapon similar to yours. Your enemy is not human, though, it is a computerized weapon system. Your mission is to destroy the enemy, by hitting it with, a deadly shoot from your cannon. Your weapon is aimed by specifying the elevation of the barrel. You have a man perched high atop the dune next to you, who Calls out the distance your projectile landed in front of or behind your target. Based on this information, you adjust the elevation of your cannon until you destroy the- target.

After each shot you take, it takes you some time to reload the cannon. While you reload, the computer fires on you. It has no allies, but it has an infrared detector which can sense how far away its exploding shells hit. It cannot determine your distance away from itself, because the sensor is not sensitive enough to detect your body heat. It can detect whether its shells hit in front of you or behind you, by checking for your shadow in the infrared pattern it gets back. Using this distance information, the computer' follows a stepping algorithm to zero in on your position.

Your man on the dune has a radio receiver, which he has rigged to pick up the stray radio-frequency emissions produced by the computer. He is able to decode four different signals but he does not know what they stand for; it is your job to figure them out before it is too late. The man is in good position to determine the distances the enemy shells fall away from you, but he is unable to determine the angle of elevation of the enemy's cannon.

There is a vantage point which will let you see this; if you can direct your man to the proper place, he will be able to tell you exactly what the computer's next move will be. If you find this location, you can relax. You can actually let the computer calculate the proper elevation for you, then you can use that elevation to blow the computer away. It's not fair', but this is war, remember'?

Finally, if you should win, you'll hear a mild fanfare as the people of the free world take you up on their shoulders and give you a tickertape parade through the streets of New York. If the enemy targets you for destruction, you will hear only the sound of death. In either case, when the game is done, you can search the desert for more of these computerized death machines, and when one is located, start all over again.

Requirements: HP-41C, Extended Functions module, Hyperactive Imagination Pac -1B.

Program listing:

01*LBL "TARG"	50 AVIEW	99 STO 46	148 ABS
02 50	51 PSE	100 ABS	149 RCL 49
03 XROM "INIT"	52 PSE	101 11	150 4.5
04 .	53 RCL 41	102 X>Y?	151 /
05 STO 47	54 SIN	103 GTO IND 19	152 X<=Y?
06 STO 48	55 ABS	104 CLA	153 GTO 04
07 "SKILL? 0-5"	56 20	105 "a"	154 2.2
08 PROMPT	57 *	106 ARCL 46	155 ST/ 45
09 E3	58 LASTX	107 "` FT"	156 SF 04
10 /	59 +	108 AVIEW	157*LBL 04
11 STO 15	60 STO 44	109 PSE	158 FS? 03
12*LBL 00	61 LASTX	110 PSE	159 GTO 05
13 RCL 15	62 STO 45	111 ISG 20	160 RCL 46
14 STO 14	63 DEG	112 GTO C	161 ABS
15 X=0?	64*LBL 01	113 FS? 06	162 RCL 49
16 SF 06	65 E-3	114 GTO 03	163 E1
17 FIX 1	66 STO 20	115 ISG 14	164 /
18 RAD	67*LBL A	116 GTO a	165 X<=Y?
19 RCL 41	68 .	117 CF 10	166 GTO 05
20 SIN	69 "ANGLE?"	118 SF 06	167 2.5
21 ABS	70 PROMPT	119 SF 05	168 ST/ 45
22 8 E2	71 CLD	120*LBL a	169 SF 03
23 *	72 STO 43	121 FS? 09	170*LBL 05
24 2 E2	73 43	122 GTO 02	171 FS? 02
25 +	74 STO 16	123 XEQ 09	172 GTO 06
26 STO 42	75 11	124 GTO 07	173 RCL 46
27 X^2	76 STO 19	125*LBL 02	174 ABS
28 16.1	77 GTO D	126 RCL 44	175 RCL 49
29 /	78*LBL C	127 RCL 12	176 45
30 45	79 "COMPUTER..."	128 -	177 /
31 SIN	80 AVIEW	129 RCL 46	178 X<=Y?
32 X^2	81 E	130 RCL 13	179 GTO 06
33 *	82 ST+ 16	131 -	180 4
34 STO 41	83 ST+ 19	132 /	181 ST/ 45
35 STO 49	84*LBL D	133 RCL 13	182 SF 02
36 .9	85 RCL 42	134 *	183*LBL 06
37 *	86 X^2	135 CHS	184 FS? 01
38 RCL 42	87 2	136 RCL 12	185 GTO 07
39 SIN	88 *	137 +	186 RCL 46
40 ABS	89 32.2	138 XEQ 04	187 ABS
41 *	90 /	139 R^	188 RCL 49
42 RCL 41	91 RCL IND 16	140 R^	189 90
43 E-1	92 SIN	141 STO 44	190 /
44 *	93 *	142 FC? 06	191 X<=Y?
45 +	94 RCL IND 16	143 SF 10	192 GTO 07
46 STO 41	95 COS	144*LBL 03	193 3
47 "DIS "	96 *	145 FS? 04	194 ST/ 45
48 ARCL 41	97 RCL 41	146 GTO 04	195 SF 01
49 "` FT"	98 -	147 RCL 46	196*LBL 07

197 FS? 10	218 RCL 44	239 ""DESTROYED"	260 E
198 GTO 01	219 STO 12	240 AVIEW	261 ST+ 48
199 FS?C 05	220 RCL 46	241 PSE	262*LBL 13
200 GTO 01	221 STO 13	242 TONE 8	263 RCL 47
201 RCL 46	222 SF 09	243 TONE 9	264 ENTER^
202 X>0?	223 RTN	244 E	265 RCL 48
203 GTO 08	224*LBL 10	245 ST+ 47	266 +
204 FS? 08	225 2	246 GTO 13	267 /
205 XEQ 10	226 ST/ 45	247*LBL 12	268 E2
206 SF 07	227 CF 07	248 "YOU ARE HIT"	269 *
207 RCL 45	228 CF 08	249 AVIEW	270 STO 49
208 ST+ 44	229 RTN	250 TONE 6	271 CLA
209 GTO 01	230*LBL 11	251 FIX 2	272 FIX 2
210*LBL 08	231 TONE 8	252 "ANGLE="	273 "AVG "
211 FS? 07	232 TONE 9	253 ARCL 44	274 ARCL 49
212 XEQ 10	233 "*** HIT ***"	254 "" DEG"	275 "" %"
213 SF 08	234 AVIEW	255 AVIEW	276 .
214 RCL 45	235 TONE 8	256 PSE	277 X<>F
215 ST- 44	236 TONE 9	257 "I WIN"	278 PROMPT
216 GTO 01	237 PSE	258 AVIEW	279 GTO 00
217*LBL 09	238 "TARGET"	259 PSE	280 END

Robot Trap for the HP-41C/CV/CX

HP Co. – Games Solutions Books

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

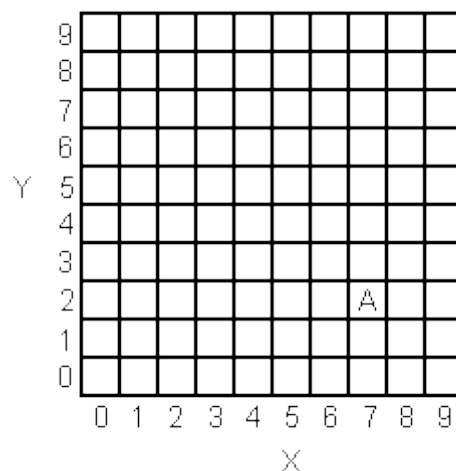
This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

You move your android to any adjacent square on a 10 x 10 playing board studded with destructive force fields and up to 49 enemy robots in such a way as to lure the robots into their own electronic booby traps and save the android. The robots will always close on the android, moving to the square adjacent to their present position which is nearer to the row and column position of the android, and will team up to destroy him. The android, like the robots, is destroyed by moving into a force field, and the android is also destroyed by colliding with a robot. If robots collide all but one involved are destroyed. You choose the initial number of robots. The number of force fields is equal to the initial number of robots plus one. Even a few robots can be challenging and the more robots the more difficult. All initial positions are randomly generated.

If you move the android into a force field you will see "ZAP" then "TOO BAD". If you move the android into a robot you will see "STOMP" then "TOO BAD". If a robot stumbles into a force field you will see "STUMBLE". And, if robots hit, you will see "BUMP". Finally, if all robots are destroyed, you will see "YOU WIN".

The board is set up as below:



Board positions are denoted as x.y . The android shown is at 7,2. To move the android use the digits keys to specify directions as follows:

1 - down and left	6 - right
2 - down	7 - up and left
3 - down and right	8 - up
4 - left	9 - up and right
5 - no movement	

Note: Do not move off the board. Execution times at all points in the program increase with the initial number of robots.

Note: Requires 1 Memory Module on HP-41C

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] RT	SEED?
3	Key in number between 0 and 1	seed	[R/S]	NO.OF ROBOTS
4	Key in the number of robots you wish.	num	[R/S]	SET SIZE(NNN)
5	Set size specified.		[XEQ] SIZE(NNN) [R/S]	F.F. AT ()*
	(If size was already set at or above specified, the program will skip that prompt.)			
6	Continue pressing [R/S] to see the rest of the force field and robot Placements until the android placement appears.		[R/S].
			[R/S]	ANDROID: ()
7	Move the android (only when "ANDROID: ()" is in display.	direction	[R/S]	ROBOT AT ()
8	Go to 6 to see the rest of the robot placements and new android placement.			
9	For a new game		[E]	NO.OF ROBOTS
10	Go to step 4.			
*	Force field placements are only output at the beginning of the game.			

Example

Keystrokes:	Display:
[XEQ] [ALPHA]	
SIZE [ALPHA] 014	
[XEQ] [ALPHA]	
RT [ALPHA]	SEED?
.12569 [R/S]	NO. OF ROBOTS
3 [R/S]	F.F. AT 7,8
[R/S]	F.F. AT 7,1
[R/S]	F.F. AT 3,6
[R/S]	F.F. AT 3,3
[R/S]	ROBOT AT 6,8
[R/S]	ROBOT AT 0,6
[R/S]	ROBOT AT 7,9
[R/S]	ANDROID: 6,1
3 [R/S]	STUMBLE
	ROBOT AT 7,7
[R/S]	ROBOT AT 1,5
[R/S]	ANDROID: 7,0
4 [R/S]	ROBOT AT 6,6
[R/S]	ROBOT AT 2,4
[R/S]	ANDROID: 6,0
.	.
.	.
.	.

Program listing:

01 LBL "RT"	28 /	56 RCL IND Y	84 FS? 05
02 SF 27	29 ST+ 03	57 X=Y?	85 GTO 97
03 "SEED?"	30 6.005	58 GTO 00	86 PSE
04 PROMPT	31 +	59 RDN	87 "ANDROID
05 STO 00	32 STO 02	60 DSE Y	SAFE"
06 LBL E	33 RCL 03	61 GTO 11	88 AVIEW
07 "NO. OF	34 +	62 STO IND 03	89 TONE 09
ROBOTS"	35 .001	63 1	90 TONE 08
08 PROMPT	36 +	64 ST+ 04	91 TONE 09
09 STO 03	37 STO 01	65 ISG 03	92 TONE 07
10 2	38 FRC	66 GTO 00	93 TONE 09
11 *	39 6	67 FIX 01	94 RTN
12 1	40 +	68 CF 28	95 LBL 10
13 +	41 STO 03	69 RCL 01	96 STO 04
14 6	42 5.004	70 "F. F. "	97 LBL 16
15 +	43 STO 04	71 ASTO 03	98 RCL IND 04
16 "SET SIZE "	44 100	72 XEQ 10	99 X<0?
17 FIX 00	45 XEQ 99	73 LBL 98	100 GTO 10
18 CF 29	46 10	74 CF 28	101 SF 05
19 ARCL X	47 /	75 RCL 02	102 CLA
20 SF 25	48 STO 05	76 "ROBOT "	103 ARCL 03
21 1	49 LBL 00	77 ASTO 03	104 " -AT "
22 -	50 RCL 04	78 CF 05	105 ARCL X
23 STO IND X	51 100	79 XEQ 10	106 AVIEW
24 FC?C 25	52 XEQ 99	80 "ANDROID: "	107 STOP
25 PROMPT	53 10	81 ARCL 05	108 LBL 10
26 RCL 03	54 /	82 AVIEW	109 ISG 04
27 1 E3	55 LBL 11	83 SF 28	110 GTO 16

111 RTN	145 ST+ 05	179 -1	213 RCL 01
112 LBL 97	146 RCL 05	180 X<> IND 03	214 X<>Y
113 CF 22	147 XEQ 96	181 X<0?	215 LBL 12
114 STOP	148 FS? 05	182 GTO 10	216 RCL IND Y
115 FC? 22	149 GTO 95	183 XEQ 96	217 X=Y?
116 GTO 98	150 RCL 02	184 FC? 05	218 RTN
117 GTO IND X	151 STO 03	185 GTO 10	219 RDN
118 LBL 01	152 LBL 14	186 -1	220 ISG Y
119 -1.1	153 RCL 05	187 "BUMP"	221 GTO 12
120 GTO 10	154 INT	188 FS? 06	222 CF 06
121 LBL 02	155 RCL IND 03	189 "STUMBLE"	223 RCL 02
122 -.1	156 X<0?	190 AVIEW	224 X<>Y
123 GTO 10	157 GTO 10	191 TONE 09	225 LBL 13
124 LBL 03	158 INT	192 CLD	226 RCL IND Y
125 .9	159 -	193 LBL 10	227 X=Y?
126 GTO 10	160 X#0?	194 STO IND 03	228 RTN
127 LBL 04	161 SIGN	195 ISG 03	229 RDN
128 -1	162 RCL 05	196 GTO 15	230 ISG Y
129 GTO 10	163 FRC	197 RCL 05	231 GTO 13
130 LBL 05	164 RCL IND 03	198 XEQ 96	232 CF 05
131 0	165 FRC	199 FC? 05	233 RTN
132 GTO 10	166 -	200 GTO 98	234 LBL 99
133 LBL 06	167 X#0?	201 LBL 95	235 RCL 00
134 1	168 SIGN	202 "STOMP"	236 9821
135 GTO 10	169 10	203 FS? 06	237 *
136 LBL 07	170 /	204 "ZAP"	238 .211327
137 -.9	171 +	205 AVIEW	239 +
138 GTO 10	172 ST+ IND 03	206 TONE 00	240 FRC
139 LBL 08	173 LBL 10	207 "TOO BAD"	241 STO 00
140 .1	174 ISG 03	208 AVIEW	242 *
141 GTO 10	175 GTO 14	209 RTN	243 INT
142 LBL 09	176 RCL 02	210 LBL 96	244 RTN
143 1.1	177 STO 03	211 SF 05	245 END
144 LBL 10	178 LBL 15	212 SF 06	

Scatter for the HP-41C/CV/CX

HP Co. – Games Solutions Books

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

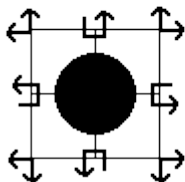
Overview

N atoms are randomly placed in a black box with dimensions 10x10. No atom can be on the edge of the box. By firing particles into the box from the edges, and noting their exit locations, you attempt to find the atom positions. For a single atom, the scatters and reflections are as shown in Figure 1. Multiple atom scatters are simple extensions of this diagram: See, for examples, Figures 2 and 3. Note in particular, the back reflections of Figure 3 which arise from two combined scatters. More complex scattering and reflection are shown in Figure 4 where atom 4 causes scatter A, atom 2 causes B, 1 causes C, 4 (again) causes D, 3 causes E, and atom 1 reflects the particle back along the convoluted path. The numbering of the box grid is given in Figure 5. The 5th position on the base has coordinates 5.0, the 7th on the right is 9.7, and so on.

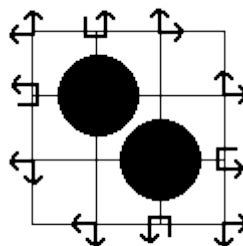
You select the value of N, and the machine places the N atoms randomly. You then fire particles from the edge: The machine tracks them and displays the output edge locations. At any time you can get the machine to confirm or reject any suspected atom location. If the guess is wrong, you are "penalized" by having the number of used particles increased by 5. The object of the game is not only to find the atoms, but to do so with the minimum number of probes.

NOTE: Although 9 atoms may be placed, a "good" game is 4 or 5.

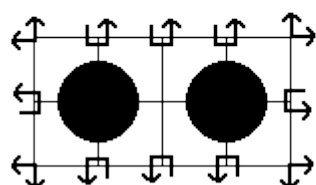
Diagrams



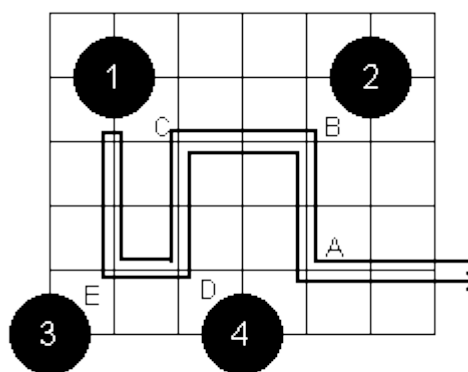
1. Single atom



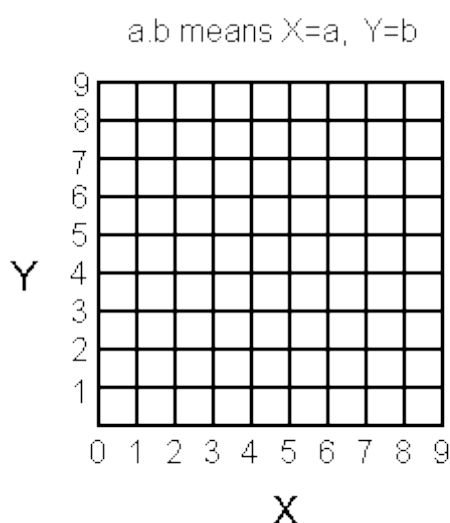
2. Two atoms



3. Two atoms



4. Complex Reflection



5. Box numbering and axes

Note: Requires 1 Memory Module on HP-41C

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] SCATTER	SEED?
3	Key in number between 0 and 1	seed	[R/S]	NO. OF ATOMS?
4	Key in number of atoms to be placed.	N	[R/S]	READY
5	Key in an entrance point.	x.y	[R/S]	(x).(y)
6	To guess a position key one in.	x.y	[A]	YES (or) NO
				() PROBES

	Continue with steps 5 and 6 as desired.			
7	To start a new game go to step 4.		[E]	NO. OF ATOMS?
8	If at any time you give up you can find all positions.		[C]	(x).(y) ...

Example

Set up and find 4 atoms

<u>Keystrokes:</u>	<u>Display:</u>
[XEQ] [ALPHA]	
SIZE [ALPHA] 022	
[XEQ] [ALPHA]	
SCATTER [ALPHA]	SEED?
.191062 [R/S]	NO. OF ATOMS?
4 [R/S]	READY
2.0 [R/S]	0,2
4.0 [R/S]	9,2
3.3 [A]	YES
	2 PROBES
6.9 [R/S]	6,9
6.8 [A]	NO
	8 PROBES
0.8 [R/S]	9,8
.	.
.	.

Program listing:

LINE	<u>KEYS</u>	
01	<u>LBL "SCATTER"</u>	25 +
02	SF 27	26 RCL 12
03	CF 29	27 INT
04	SF 28	28 X<>Y
05	FIX 00	29 STO 09
06	"SEED?"	30 LBL 01
07	PROMPT	31 RCL IND Y
08	STO 00	32 X=Y?
09	LBL E	33 GTO 00
10	0	34 RDN
11	STO 11	35 LBL 09
12	"NO. OF ATOMS?"	36 DSE Y
13	PROMPT	37 GTO 01
14	STO 10	38 STO IND 12
15	1 E3	39 ISG 12
16	/	40 GTO 00
17	1	41 "READY"
18	+	42 AVIEW
19	STO 12	43 GTO 20
20	LBL 00	44 LBL 10
21	XEQ 10	45 RCL 00
22	XEQ 10	46 9821
23	10	47 *
24	/	48 .211327
		49 +
		50 FRC
		51 STO 00
		52 8
		53 *
		54 1
		55 +
		56 INT
		57 RTN
		58 LBL 20
		59 STOP
		60 INT
		61 STO 12
		62 STO 15
		63 CF 00
		64 X=0?
		65 SF 00
		66 9
		67 X=Y?
		68 SF 00
		69 CF 01
		70 X=Y?
		71 SF 01
		72 LASTX

73 FRC	122 STO 18	171 FS? 00
74 10	123 RCL 16	172 SF 02
75 *	124 STO 19	173 SF 00
76 STO 13	125 SF 02	174 FS?C 02
77 STO 14	126 LBL 09	175 CF 00
78 X=Y?	127 DSE 21	176 CF 01
79 SF 01	128 GTO 05	177 RCL 18
80 1	129 FS?C 02	178 X=0?
81 ST+ 11	130 GTO 09	179 GTO 03
82 LBL 02	131 0	180 X>0?
83 RCL 10	132 ENTER	181 SF 01
84 STO 21	133 9	182 GTO 02
85 10	134 FS? 01	183 LBL A
86 STO 17	135 X<>Y	184 RCL 10
87 LBL 05	136 RCL 12	185 X<>Y
88 RCL IND 21	137 RCL 13	186 LBL 04
89 INT	138 FS? 00	187 RCL IND Y
90 LASTX	139 X<>Y	188 "YES"
91 FRC	140 RDN	189 X=Y?
92 10	141 FS? 00	190 GTO 09
93 *	142 X<>Y	191 RDN
94 RCL 13	143 STO 15	192 DSE Y
95 -	144 RDN	193 GTO 04
96 X<>Y	145 STO 14	194 5
97 RCL 12	146 LBL 03	195 ST+ 11
98 -	147 CLA	196 "NO"
99 FS? 00	148 ARCL 15	197 LBL 09
100 X<>Y	149 "┐, "	198 AVIEW
101 STO 20	150 ARCL 14	199 PSE
102 ABS	151 AVIEW	200 CLA
103 1	152 GTO 20	201 ARCL 11
104 -	153 LBL 09	202 "┐ PROBES"
105 X>0?	154 FS?C 03	203 AVIEW
106 GTO 09	155 GTO 03	204 GTO 20
107 RDN	156 RCL 12	205 LBL C
108 STO 16	157 RCL 13	206 RCL 10
109 FS? 01	158 FS? 00	207 FIX 01
110 CHS	159 X<>Y	208 CF 28
111 X<0?	160 RCL 19	209 LBL 06
112 GTO 09	161 1	210 CLA
113 RCL 17	162 FS? 01	211 ARCL IND X
114 X<>Y	163 CHS	212 AVIEW
115 X>Y?	164 -	213 PSE
116 GTO 09	165 +	214 DSE X
117 SF 03	166 FS? 00	215 GTO 06
118 X#Y?	167 X<>Y	216 SF 28
119 CF 03	168 STO 13	217 FIX 00
120 STO 17	169 RDN	218 GTO 20
121 RCL 20	170 STO 12	219 END

Hexapawn for the HP-41C/CV/CX

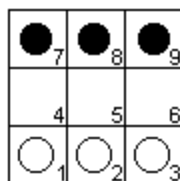
HP Co. – Games Solutions Books

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

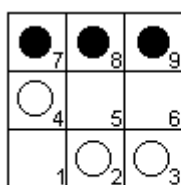
This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

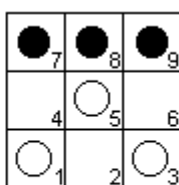
Hexapawn is a game which is programmed to learn from its mistakes. The game is played with chess pawns on a 3x3 board. Pawns may advance one square at a time or capture the opponent's pawns by moving diagonally one square. The game starts with the pawns positioned as follows:



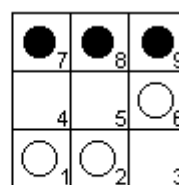
Note the resemblance between the 41C digit keys and the board numbering. The three allowed opening moves for the first player (in this example, white) are 1 to 4 (keyed as 1.4), 2 to 5 (2.5), and 3 to 6 (3.6).



1 to 4

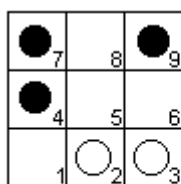


2 to 5

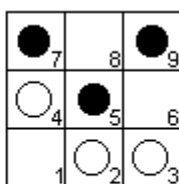


3 to 6

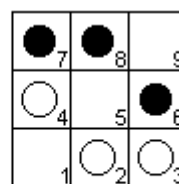
Black's three possible responses to white's 1.4 move are 8 to 4, 8 to 5, and 9 to 6.



8 to 4



8 to 5



9 to 6

Black can move diagonally and capture white (8 to 4), or he can move either pawns 8 or 9 straight ahead one square. The black pawn at 7 is blocked. Note that the only way a pawn can

move to an open square is straight ahead. Also the only way a pawn can capture is by moving diagonally.

The game is won by advancing a pawn to the third row, capturing all of the opponent's pawns, or creating a position in which the opponent cannot move.

Moves are made by keying in the board position of the pawn to be moved, a decimal point, then the board position the pawn is to be moved to. The 41C does not check for illegal moves; therefore, you are on your honor not to cheat. The 41C selects its move at random, but if it is then punished, it remembers not to make that move in that situation. Thus, if the machine makes a poor move and is punished, it will not repeat the mistake.* Also, if the mirror image game is played, it will not make the mirror image of the poor move. If a point is reached in a game where all possible moves for a certain board configuration have received previous punishment, "NO MOVE" and "YOU WIN" is displayed, just as if there really were no move. If you cannot move, you can if you wish be a good sport and tell the 41C by keying 0 for your move. It will respond with "I WIN". If chess pawns are not available for visualization, different colored coins work well.

*Similarly, you can punish good moves to make it play a losing game.

Note: Requires 2 Memory Modules on HP-41C

Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize type of game: human first,		[XEQ] HUMAN	SEED?
	or machine first.		[XEQ] MACHINE	SEED?
3	Key in a seed for the random number generator between 0 and 1. (Just hit [R/S] if a seed has been previously entered.)	seed	[R/S]	READY (or)
				() to ()
4	Key in your move FROM.TO	F.T	[R/S]	() to ()
	Repeat step 4 until game is over			
5	OPTIONAL: After the 41C displays its move, punish it.		[E]	AAAIIII...
6	To signify that you can't move	0	[R/S]	I WIN
7	To start a new game with the same player first and punishments remembered.		[A]	READY (or)
				() to ()

	Go to step 4			
8	To start a new game with a different player first and/or forgetting punishments go to step 2.			

Example

Keystrokes:

Display:

```
[XEQ] [ALPHA] SIZE
[ALPHA] 014
[XEQ] [ALPHA]
MACHINE [ALPHA] SEED?
.111111111 [R/S] 8 to 5
1.5 [R/S] 7 to 4
(A bad move; therefore,
punish)
[E] AAIIIII...
5.8 [R/S] YOU WIN
```

Start a new game with the 41C remembering its punishment.

```
[A] 8 to 5
1.5 [R/S] 7 to 5
3.5 [R/S] 9 to 5
0 [R/S] I WIN
```

Program listing:

```
01 LBL "MACHINE"      24 STO 10      47 FC?C 08
02 XEQ 01             25 .8596      48 GTO 00
03 8388607            26 STO 06      49 CF 07
04 STO 01             27 1        50 3.6
05 3139583            28 FS? 05      51 FS? 05
06 STO 02             29 GTO 20      52 CHS
07 34314              30 "READY"     53 X=Y?
08 STO 03             31 AVIEW      54 SF 07
09 SF 05              32 GTO 30      55 RDN
10 GTO A              33 LBL 01      56 1.4
11 LBL "HUMAN"        34 SF 27      57 FC? 05
12 XEQ 01             35 CF 07      58 CHS
13 16777215           36 CLRG      59 X=Y?
14 STO 01             37 "SEED?"     60 SF 07
15 16756735           38 RCL 00      61 RDN
16 STO 02             39 PROMPT     62 1.5
17 524413             40 STO 00      63 X=Y?
18 STO 03             41 RTN        64 SF 07
19 CF 05              42 LBL 30      65 RDN
20 LBL A              43 STOP      66 LBL 00
21 SF 09              44 "I WIN"     67 CF 09
22 SF 08              45 X=0?     68 STO 13
23 9503               46 PROMPT     69 FRC
```

70 .7	133 XEQ 23	196 FRC
71 X<=Y?	134 137.9563	197 STO 06
72 GTO 26	135 XEQ 23	198 RCL 13
73 RCL 13	136 GTO 25	199 INT
74 XEQ 50	137 LBL 22	200 LBL 20
75 INT	138 9503.8596	201 STO 08
76 STO 11	139 XEQ 23	202 RCL 04
77 LASTX	140 7341.748586	203 STO 09
78 FRC	141 XEQ 23	204 RCL 05
79 10	142 7449.7475	205 STO 07
80 *	143 XEQ 23	206 2
81 STO 12	144 3237.848562	207 RCL 08
82 CF 06	145 XEQ 23	208 Y^X
83 XEQ 21	146 8957.7451	209 STO 05
84 .9	147 XEQ 23	210 3
85 STO 13	148 8849.9596	211 STO 13
86 FS? 05	149 SF 09	212 RCL 00
87 GTO 22	150 XEQ 23	213 9821
88 3185.848596	151 CF 09	214 *
89 XEQ 23	152 6.9	215 .211327
90 7397.7475	153 STO 13	216 +
91 XEQ 23	154 8849.7475	217 FRC
92 1316.417596	155 XEQ 23	218 STO 00
93 XEQ 23	156 6687.7475	219 *
94 1142.845396	157 XEQ 23	220 1
95 XEQ 23	158 855.7475	221 +
96 2531.759596	159 XEQ 23	222 INT
97 XEQ 23	160 1194.845253	223 STO 04
98 1023.848586	161 XEQ 23	224 LBL 02
99 XEQ 23	162 2583.756263	225 3
100 6758.515286	163 XEQ 23	226 RCL 04
101 XEQ 23	164 6702.63	227 1
102 7163.9586	165 XEQ 23	228 +
103 XEQ 23	166 1260.41	229 X>Y?
104 2720.4142	167 XEQ 23	230 1
105 XEQ 23	168 1368.7541	231 STO 04
106 1131.8475	169 XEQ 23	232 X<> 13
107 XEQ 23	170 6887.9596	233 RCL IND 13
108 818.9596	171 XEQ 23	234 X<>Y
109 XEQ 23	172 2783.414295	235 X<> 13
110 6650.9596	173 XEQ 23	236 RDN
111 XEQ 23	174 6995.5152	237 RCL 05
112 992.96	175 XEQ 23	238 /
113 XEQ 23	176 1179.5253	239 FRC
114 677.4152	177 XEQ 23	240 .5
115 XEQ 23	178 2286.7475	241 X<=Y?
116 369.75	179 XEQ 23	242 GTO 04
117 XEQ 23	180 2270.9596	243 DSE 13
118 600.4186	181 XEQ 23	244 GTO 02
119 XEQ 23	182 2594.96	245 RCL 07
120 384.8463	183 XEQ 23	246 STO 05
121 XEQ 23	184 621.4163	247 STO 04
122 693.4152	185 XEQ 23	248 FS? 08
123 XEQ 23	186 432.52	249 CF 09
124 461.5263	187 XEQ 23	250 FS? 09
125 XEQ 23	188 GTO 25	251 RTN
126 569.419596	189 LBL 23	252 LBL 25
127 XEQ 23	190 ISG 13	253 "NO MOVE"
128 411.8452	191 INT	254 AVIEW
129 XEQ 23	192 RCL 10	255 PSE
130 195.5286	193 X#Y?	256 LBL 26
131 XEQ 23	194 RTN	257 "YOU WIN"
132 585.4175	195 LASTX	258 AVIEW

259 STOP	302 "┐ TO "	345 FRC
260 LBL 04	303 LASTX	346 3
261 RCL 04	304 FRC	347 *
262 1	305 10	348 INT
263 X#Y?	306 *	349 CHS
264 CF 08	307 ARCL X	350 1
265 -	308 AVIEW	351 FS? 06
266 2	309 GTO 30	352 ST+ X
267 *	310 LBL 50	353 +
268 10^X	311 FC? 07	354 3
269 RCL 06	312 RTN	355 ENTER
270 *	313 STO 06	356 9
271 FRC	314 INT	357 ENTER
272 10	315 1	358 RCL 12
273 *	316 -	359 -
274 INT	317 3	360 Y^X
275 STO 11	318 /	361 *
276 LASTX	319 INT	362 3
277 FRC	320 5	363 ENTER
278 10	321 +	364 9
279 *	322 XEQ IND X	365 RCL 11
280 INT	323 RCL 06	366 -
281 STO 12	324 -	367 Y^X
282 SF 06	325 RTN	368 1
283 XEQ 21	326 LBL 05	369 FS? 06
284 3	327 5	370 ST+ X
285 RCL 12	328 RTN	371 *
286 X>Y?	329 LBL 06	372 -
287 GTO 00	330 10.4	373 ST+ 10
288 AVIEW	331 RTN	374 RTN
289 BEEP	332 LBL 07	375 LBL E
290 LBL 00	333 17	376 RCL 04
291 RCL 11	334 RTN	377 X<> 13
292 RCL 12	335 LBL 21	378 RCL 05
293 10	336 RCL 10	379 2
294 /	337 3	380 /
295 +	338 ENTER	381 ST- IND 13
296 XEQ 50	339 10	382 RDN
297 FIX 00	340 ENTER	383 X<> 13
298 CF 29	341 RCL 12	384 "AAAIIII..."
299 CLA	342 -	385 AVIEW
300 INT	343 Y^X	386 GTO 30
301 ARCL X	344 /	387 END

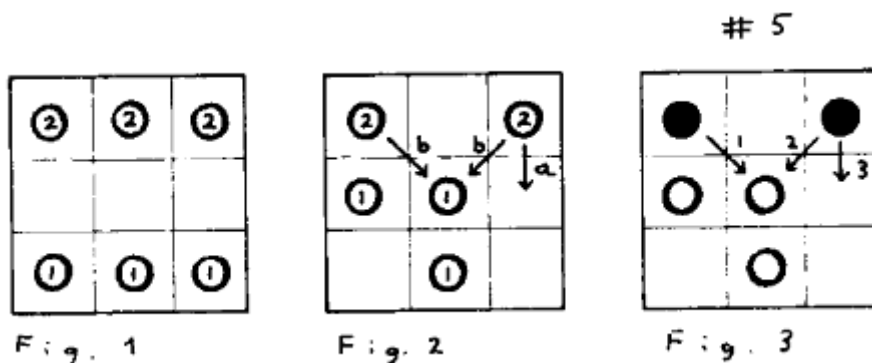
Hexapawn, v2

[Peter Hamer / Edwin Hartingsveldt – PPCCJ V7N4p31 \(May 1980\)](#)

Hexapawn is one of the "classics" among programmable calculator games. It was originally developed by John Rausch for the HP-65 (V2N3). It was subsequently translated by HP for the HP-67 Games Pac and adapted for the HP-25 by Bob Hall (V4N6).

The game's rules are rather simple: on a 3 by 3 board two players have 3 pawns each. The starting position is shown in fig. 1. The legal moves are derived from the pawn in chess: a pawn can advance one step forward (fig. 2a) to an unoccupied square or can capture an opponent's pawn moving forward diagonally (fig. 2b). The machine and the human take turns. A game is won as soon as a player has one of his pawns on the "back" row or when the opponent can't make any legal move.

In previous versions of the game you told the machine that after your move the board configuration was as shown in fig. 2 by looking this position up in a set of possible configurations (like fig. 3) in the program's documentation. This gave you the position's characteristic number. In this case you would input 5 and the machine would respond with a 1, 2 or 3 corresponding to the numbered arrows in fig. 3 (when the machine gave up, you saw a "0").



In our version you input the configuration of fig. 3 as the code number 210 011 200 ("scanning" the board from top left to bottom left, etc.). The machine will answer with 010 021 200, 210021 000 or 210 011020.

The game as described isn't really all that thrilling. It's about just as "mind-boggling" as tic-tac-toe. The interesting part of all these programs was that the machine, when turned on, made its choices at random. As the game progresses the human is expected to "punish" + the machine whenever it makes a stupid move.

Gradually the machine's playing level increases to the point when it's more or less perfect. Thus, the machine learns from its mistakes. In this version the machine is smart enough that, once you tell it that 201 010 220+ 201 020 020 is rather dumb, it automatically avoids making 220 010 201 ~ 020 020 201 because this is essentially the same case (its mirror image!).

In the original game you made this decision yourself because only one configuration in a pair of mirror images was depicted.

Because of changes in the learning algorithm, the new version has a slight edge in learning speed over the original version (after the last example the move 201 220 010 ~ 201 020 020 will also be blacklisted).

To the user it may seem that most of the changes benefit the program's ease-of-use because, after all, both versions play the same game according to the same rules. However if you were to dissect the original version you would find that the program doesn't contain an algorithm for hexapawn at all:

To put it provocatively, if you were to change a few constants and replace the "board configuration tables" in the program's documentation, it would be playing a cybernetic version of tic-tac-toe or possibly even checkers on a small board. This version, on the other hand, consists of a move generating part that simply gives a complete list of legal moves for any meaningful position, and a "learningIt part which decides (together with a random number generator) which of these move to choose.

User Instructions

1) Load program- and data cards.

2) Set up your playing board as in figure I. You can use pennies and dimes as playing pieces - or if you have just bought the HP-41C you may have to find yourself light and dark pebbles.

3) Make your move. Input code for new board position. Press A. (You can have the machine make the first move by inputting the code for fig. I, 201 202 201. This gives a less interesting game because the human tends to win all the time). The machine responds with a new code (its move) or "Error" (if it gives up). The error display is generated internally at step 112 (in case you're in doubt).

4) Optional: Punish the machine for its move by pressing B. Normally you punish for bad moves. You can, however, punish according to whatever criterium you want. Punishing the machine after it has just won is, however, going a bit too far.

5) For your next move, go to 3).

6) For a new game go to 2).

7) Should you want to brainwash your pupil back to the level of an honest playing five-years-old, clear the ten secondary registers. or "point out tactfully to 11 - for the enlightened parents amongst you.

Notes: the machine's accumulated experience can be stored on a datacard. Making this program fit on the basic HP-41C is not going to be easy as it uses all the data registers and all the program space. When translating, watch those DSZ's because the fractional part of RI can be non-zero. The 41C's ability to partition numbers in the display into groups of three using komma's will be quite nice here. Important: The calculator status on your card should be: Rad, Fix 0, flags irrelevant.

Peter Van Den Hamer (3533) & Edwin Van Hartingsveldt
PPC V8N1 p26 ; Jan/Feb 1981

Cybernetic Hexapawn

[Marco de Vries, PP V8N1 p26 ; \(Jan/Feb 1981\)](#)

Peter van den Hamer and Edwin van Hartingsveldt invited the membership to produce a 4IC version of their brilliant "cybernetic hexapawn", published in PPC V7N4 p31. Here is my attempt.

This version will print out the checkerboard and pawns between moves when the printer is connected; it will also play without printer. In addition this version will end the game automatically when one of the players has won by indicating "I WIN" or "YOU WIN". For the rules of the game and the code for indicating moves I refer to the original article.

INSTRUCTIONS:

Size 038; RAM 2; Read program with F11 on

1. Automatic start; turn calculator ON
2. Enter seed when prompted for; default seed is PI, SIN, TAN.
3. After "READY" in display, enter code for move and press [A] or when Black is the first player press [B]
4. Continue moving by entering code and [A]
5. To punish or rather induce learning push [E] before your next move
6. At end of game WDTA for preserving learning and/or push R/S where after machine's default status will be set and the calculator is turned off.
7. To continue for a new game, enter first move and push [A], or push [B]
8. To continue game after calculator has been turned off, pass data Card after the "READY" prompt

Marco J. de Vries (4258)

Program listing:

01*LBL "HEX"	50 "#"	<u>99*LBL 00</u>	148 ST+ 07
02 DEG	51 SF 06	100 8	149 1.2
03 CF 27	52 XEQ 33	101 STO 25	150 RCL 07
04 CF 29	53 ADV	102 RCL 00	151 X=Y?
05 FIX 4	54 RTN	103 E	152 XEQ 07
06 SF 11	55*LBL B	104 %	153 FRC
07 OFF	56 RCL 20	105*LBL 04	154 RCL 21
08*LBL C	57 SF 01	106 STO 07	155 *
09 SF 12	58*LBL A	107 FRC	156 STO 07
10 " HEXAPAWN"	59 " I"	108 .2	157 DSE 25
11 AVIEW	60 X=0?	109 X=Y?	158 GTO 06
12 CLRG	61 GTO 55	110 XEQ 05	<u>159*LBL 08</u>
13 RAD	62 CF 00	111 RCL 07	160 E
14 .031642758	63 STO 22	112 FIX 1	161 ST- 03
15 STO 01	64 STO 00	113 RND	162 ST- 05
16 .637048152	65 FS?C 01	114 RCL 21	163 " YOU"
17 STO 02	66 GTO 88	115 /	164 RCL 03
18 30	67 SF 12	116 DSE 25	165 X=0?
19 STO 09	68 " WHITE"	117 GTO 04	166 GTO 55
20 201201201	69 XEQ 44	118 GTO 08	167 RCL 05
21 STO 20	70*LBL 88	<u>119*LBL 05</u>	168 X<0?
22 STO 00	71 RCL 08	120 8	169 2
23 E1	72 9821	121 RCL 25	170 STO 05
24 STO 21	73 *	122 -	171 FS? 00
25 PI	74 .211327	123 10^X	172 RTN
26 SIN	75 +	124 9	173 STO 25
27 TAN	76 FRC	125 *	174 GTO IND 25
28 "FRC SEED ?"	77 STO 08	126 GTO 03	<u>175*LBL 07</u>
29 PROMPT	78 3	<u>127*LBL 01</u>	176 RCL 24
30 CLA	79 STO 03	<u>128*LBL 02</u>	177 8
31 STO 08	80 *	129 RCL IND 25	178 RCL 25
32 SF 29	81 INT	130 STO 24	179 -
33 FS? 55	82 STO 25	131 STO 04	180 10^X
34 XEQ 99	83 STO 05	132 9	181 *
35 SF 05	84 RCL 20	133 STO 25	182 FRC
36 XEQ 44	85 RCL 22	134 STO 07	183 RCL 21
37 SF 27	86 X#Y?	<u>135*LBL 06</u>	184 *
38 "READY"	87 GTO IND 25	136 RCL 21	185 FRC
39 PROMPT	88*LBL 13	137 ST* 04	186 LASTX
40*LBL E	89 SF 00	138 RCL 00	187 INT
41 2	90 RCL 05	139 RCL 04	188 10^X
42 RCL 25	91 3	140 INT	189 ST+ 00
43 FRC	92 *	141 ST- 04	190 X<>Y
44 RCL 09	93 E	142 10^X	191 RCL 21
45 *	94 +	143 /	192 *
46 FIX 1	95 STO 25	144 INT	193 INT
47 RND	96 XEQ 05	145 RCL 21	194 10^X
48 Y^X	97 XEQ 08	146 /	195*LBL 03
49 ST+ IND 25	98 GTO 13	147 FRC	196 ST- 00

197 ST- 00	250 STOP	303 E3	356 DSE Z
198 RCL 25	251*LBL 09	304 /	357 GTO 18
199 STO 23	252 E3	305 FRC	358 PRBUF
200 RCL 00	253 ST/ 06	306 LASTX	359 3
201 STO 06	254 RCL 06	307 INT	360 STO T
202 XEQ 09	255 FRC	308 E3	361 RDN
203 XEQ 09	256 ST- 06	309 /	362 DSE 37
204 E3	257 *	310 FRC	363 GTO 17
205 *	258 ENTER^	311 LASTX	364 XEQ 16
206 X<>Y	259*LBL 11	312 INT	365*LBL 15
207 RCL 06	260 2	313 E3	366 " YOU"
208 XEQ 11	261 -	314 /	367 26.028
209 RCL 06	262 RCL 21	315 26.034	368 11
210 X<=Y?	263 /	316 STO 37	369 XEQ 11
211 X<>Y	264 FRC	317*LBL 14	370 " I"
212 E6	265 X=0?	318 RDN	371 32.034
213 *	266 GTO 77	319 RCL 21	372 12
214 +	267 RDN	320 STO IND 37	373 XEQ 22
215 +	268 RTN	321 *	374 0
216 SIN	269*LBL 10	322 INT	375 STO 37
217 3 E2	270 RCL 23	323 ST+ IND 37	376 26.034
218 *	271 STO 25	324 RDN	377 E1
219 ABS	272 RCL 22	325 LASTX	378 E1
220 .2	273 STO 00	326 FRC	379*LBL 23
221 -	274 RCL 07	327 STO T	380 RCL IND Y
222 INT	275 RTN	328 ISG 37	381 X#Y?
223 RCL 09	276*LBL 44	329 GTO 14	382 ST+ 37
224 /	277 AVIEW	330 FC? 55	383 RDN
225 RCL 21	278 FC? 55	331 GTO 15	384 ISG Y
226 +	279 PSE	332 SF 12	385 GTO 23
227 STO 25	280 CF 12	333 3	386 RCL 37
228 FRC	281 " "	334 STO 37	387 11
229 RCL 09	282*LBL 33	335 26	388 /
230 *	283 FIX 0	336 124	389 FRC
231 RND	284 RCL 00	337 "--"	390 X=0?
232 2	285 LOG	338*LBL 17	391 GTO "Y"
233 X<>Y	286 INT	339 XEQ 16	392 RCL 37
234 Y^X	287 8	340 ACCHR	393 12
235 RCL IND 25	288 X>Y?	341*LBL 18	394 /
236 X<>Y	289 "0"	342 4	395 FRC
237 /	290 RDN	343 SKPCOL	396 X=0?
238 INT	291 7	344 RDN	397 GTO "M"
239 2	292 X>Y?	345 RCL IND Y	398 ADV
240 /	293 "0"	346 STO L	399 ADV
241 FRC	294 RDN	347 RDN	400 ADV
242 X#0?	295 6	348 XEQ IND L	401 ADV
243 GTO 10	296 X>Y?	349 RDN	402 FC?C 05
244*LBL 77	297 "0,"	350 3	403 RTN
245 SF 05	298 ARCL 00	351 SKPCOL	404 TONE 7
246 SF 12	299 AVIEW	352 RDN	405 TONE 6
247 " BLACK"	300 FS?C 06	353 ACCHR	406 RTN
248 TONE 6	301 RTN	354 ISG Y	407*LBL 10
249 XEQ 44	302 RCL 00	355 RAD	408 7

409 SKPCOL	427 PRBUF	445 AVIEW	463 STO 36
410 RTN	428 RTN	446 CF 00	464 RTN
411*LBL 11	429*LBL 22	447 CF 05	465*LBL 98
412 RCL 35	430 RCL IND Y	448 ADV	466 64
413 ACSPEC	431 X=Y?	449 ADV	467 BLDSPEC
414 RTN	432 GTO 55	450 ADV	468 64
415*LBL 12	433 RDN	451 ADV	469 BLDSPEC
416 RCL 36	434 ISG Y	452 ADV	470 106
417 ACSPEC	435 GTO 22	453 STOP	471 BLDSPEC
418 RTN	436 RTN	454 GTO "HEX"	472 RCL 37
419*LBL 16	<u>437*LBL "M"</u>	455*LBL 99	473 BLDSPEC
420 ACCHR	438 " I"	456 93	474 106
421 ACA	439 GTO 55	457 STO 37	475 BLDSPEC
422 ACCHR	<u>440*LBL "Y"</u>	458 XEQ 98	476 64
423 ACA	441 " YOU"	459 STO 35	477 BLDSPEC
424 ACCHR	442*LBL 55	460 127	478 64
425 ACA	443 BEEP	461 STO 37	479 BLDSPEC
426 ACCHR	444 " WIN"	462 XEQ 98	480 END

Tic-Tac-Toe

N. Michael Johnson; UPL #00948C

It's you against the computer in the simple but tricky and masterful game of Tic Tac Toe. The computer starts on the defensive and grants you first move. You may try as you like to get the 41C in a no-win situation but that is very unlikely since during the playing of the game the 41C makes a random and unpredictable move. I have made it possible to beat but the methods to use are hard to set up and it depends greatly on where the 41C chooses to move. Sometimes these methods work and sometimes they don't. Good Luck! you're going to need it and have fun.

One Memory Module is required. The seed for the random number generator must be a positive whole number greater than one (1)

Sample problem.

1 2 3 4 5 6 7 8 9	X 2 3 4 5 6 7 8 9	X 2 3 4 0 6 7 8 9	X 2 3 4 0 6 7 8 X	X 2 3 0 0 6 7 8 X
X 2 3 0 0 X 7 8 X	X 2 0 0 0 X 7 8 X	X 2 0 0 0 X X 8 X	X 2 0 0 0 X X 0 X	X X 0 0 0 X X 0 X

This is one of those more probable cat's game. Here you can see the # spaces for playing. Each number disappears as its place is taken. The X's represent you the player and the O's represent the computer.

Solution:

Input / Function	Display	Comments
SIZE 012		set minimum size
XEQ "TTT"	NUMBER?	Start program and prompt for seed
123456789, R/S	YOUR MOVE	prompt for the # of the spot you wish
1, R/S	SPOT 1, HUH	verification of your chosen spot
		display will clear as the 41C decides
	TAKING #5	spot the 41 is moving to
	YOUR MOVE	
9, R/S	SPOT 9, HUH	
	TAKING #3	
	YOUR MOVE	
7, R/S	SPOT 7, HUH	
	TAKING #8	
	YOUR MOVE	

2, R/S SPOT 2, HUH last available spot
 CAT'S GAME game is a tie
 GTO . 001, R/S start a new game

Program listing:

1 <u>LBL "TTT"</u>	45 STO 10	89 RCL 01
2 CIST	46 LBL 16	90 RCL 03
3 RCL 00	47 RCL IND 10	91 XEQ 21
4 CLRG	48 X=0?	92 RCL 05
5 X#O?	49 GTO 17	93 RCL 08
6 GTO 00	50 ISG 10	94 XEO 21
7 "SEED?"	51 GTO 16	95 GTO 20
8 PROMPT'	52 "CATS GAME"	96 LBL 03
9 LN1+X	53 AVIEW	97 RCL 01
10 LOG	54 BEEP	98 RCL 02
11 FRC	55 PSE	99 XEQ 21
12 ABS	56 GTO 12	100 RCL 05
13 LBL 00	57 LBL 17	101 RCL 07
14 STO 00	58 -2	102 XEQ 21
15 CF 29	59 SF 05	103 RCL 06
16 LBL 15	60 XEQ 18	104 RCL 09
17 CIST	61 2	105 XEQ 21
18 9	62 LBL 18	106 GTO 20
19 "YOUR MOVE"	63 STO 11	107 LBL 04
20 PROMPT	64 1,009	108 RCL 01
21 X<=0?	65 STO 10	109 RCL 07
22 GTO 15	66 LBL 19	110 XEQ 21
23 X>Y?	67 CLST	111 RCL 05
24 GTO 15	68 RCL IND 10	112 RCL 06
25 RCL IND X	69 X=0?	113 XEQ 21
26 X=0?	70 GTO IND 10	114 GTO 20
27 GTO 10	71 LBL 20	115 LBL 05
28 "SPOT "	72 ISG 10	116 RCL 01
29 ARCL Y	73 GTO 19	117 RCL 09
30 >" TAKEN"	74 FS7C 05	118 XEQ 21
31 AVIEW	75 RTN	119 RCL 02
32 GTO 15	76 GTO 22	120 RCL 08
33 LBL 10	77 LBL 01	121 XEQ 21
34 1	78 RCL 02	122 RCL 03
35 STO INDZ	79 RCL 03	123 RCL 07
36 "SPOT "	80 XEO 21	124 XEQ 21
37 ARCL Z "	81 RCL 04	125 RCL 04
38 "[-, HUH"	82 RCL 07	126 RCL 06
39 AVIEW	83 XEQ 21	127 XEQ 21
40 TONE IND Z	84 RCL 05	128 GTO 20
41 3	85 RCL 09	129 LBL 06
42 XEQ 29	86 XEQ 21	130 RCL 03
43 'CLD	87 GTO 20	131 RCL 09
44 1,009	88 LBL 02	132 XEQ 21

133	RCL 04	186	RCL IND 10	239	GTO 25
134	RCL 05	187	X=0?	240	LBL 09
135	XEQ 21	188	GTO IND 10	241	RCL 06
136	GTO 20	189	LBL 25	242	RCL 08
137	LCL 07	190	ISG 10	243	XEQ 26
138	RCL 01	191	GTO 24	244	GTO 25
139	RCL 04	192	GTO 27	245	LBL 26
140	XEQ 21	193	LBL 01	246	+
141	RCL 03	194	RCL 02	247	2
142	RCL 05	195	RCL 04	248	X#Y?
143	XEQ 21	196	XEQ 26	249	RTN
144	RCL 08	197	GTO 25	250	GTO 28
145	RCL 09	198	LBL 02	251	LBL 27
146	XEQ 21	199	RCL 04	252	CIST
147	GTO 20	200	RCL 09	253	5
148	LBL 08	201	XEQ 26	254	RCL 00
149	RCL 02	202	RCL 06	255	FRC
150	RCL 05	203	RCL 07	256	3
151	XEQ 21	204	XEQ 26	257	1/X
152	RCL 07	205	GTO 25	258	*
153	RCL 09	206	LBL 03	259	E^X
154	XEQ 21	207	RCL 02	260	PI
155	GTO 20	208	RCL 06	261	*
156	LBL 09	209	XEQ 26	262	FRC
157	RCL 01	210	GTO 25	263	ABS
158	RCL 05	211	LBL 04	264	ST+ 00
159	XEQ 21	212	RCL 02	265	10
160	RCL 03	213	RCL 09	266	*
161	RCL 06	214	XEQ 26	267	INT
162	XEQ 21	215	RCL 03	268	X=Y?
163	RCL 07	216	RCL 08	269	GTO 27
164	RCL 08	217	XEQ 26	270	X<=0?
165	XEQ 21	218	GTO 25	271	GTO 25
166	GTO 20	219	LBL 06	272	STO 10
167	LBL 21	220	RCL 02	273	RCL IND X
168	+	221	RCL 07	274	X#0?
169	RCL 11	222	XEQ 26	275	GTO 25
170	X#Y?	223	RCL 01	276	LBL 28
171	RTN	224	RCL 05	277	CIST
172	GTO 28	225	XEQ 26	278	-1
173	LBL 22	226	GTO 25	279	STO IND 10
174	5	227	LBL 07	280	"TAKING SPOT"
175	RCL IND X	228	RCL 04	281	ARCL 10
176	X#0?	229	RCL 08	282	TONE IND 10
177	GTO 23	230	XEQ 26	283	AVIEW
178	RDN	231	GTO 25	284	PSE
179	STO 10	232	LBL 08	285	-3
180	GTO 28	233	RCL 01	286	XEQ 29
181	LBL 23	234	RCL 06	287	GTO 15
182	1,009	235	XEQ 26	288	LBL 29
183	STO 10	236	RCL 03	289	STO 10
184	LBL 24	237	RCL 04	290	RCL 01
185	CLST	238	XEQ 26	291	RCL 02

292 RCL 03
293 XEO 30
294 RCt. 04
295 RCL 05
296 RCL 06
297 XEQ 30
298 RCL 07
299 RCL 08
300 RCL 09
301 XEQ 30
302 RCL 01
303 RCL 04
304 RCL 07
305 XEQ 30
306 RCL 02.
307 RCL 05
308 RCL 08
309 XEQ 30
310 RCL 03
311 RCL 06
312 RCL 09
313 XEO 30
314 RCt. 01
315 RCL 05
316 RCL 09
317 XEQ 30

318 RCL 03
319 RCL 05
320 RCL 07
321 XEQ 30
322 RTN
323 LBL 30
324 +
325 +
326 RCL 10
327 X#Y?
328 RTN
329 3
330 X=Y?
331 GTO 11
332 "YOU LOSE"
333 AVIEW
334 TONE 0
335 TONE 2
336 TONE 1
337 TONE 2
338 TONE 0
339 GTO 12
340 LBL 11
341 "YOU WIN"
342 AVIEW
343 TONE 9

344 TONE 7
345 TONE 8
346 TONE 7
347 TONE 9
348 LBL 12
349 CLST
350 SF 29
351 VIEW X
352 STOP
353 END

3D Tic-Tac-Toe for the HP-41C/CV/CX

HP Co. – Games Solutions Books

This program is Copyright © HP and is used here by permission. It was originally printed in the Games Solution Book. This program was entered and uploaded by [Tony Duell](#). The documentation was entered by Dave Hicks. The Barcode for this program was provided by [Brian Ward](#).

This program is supplied without representation or warranty of any kind. Tony Duell, Hewlett Packard and The Museum of HP Calculators therefore assume no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

Overview

This program pits the HP-41C against a human opponent in a game of 3-D Tic Tac Toe. The rules of this game are simple:

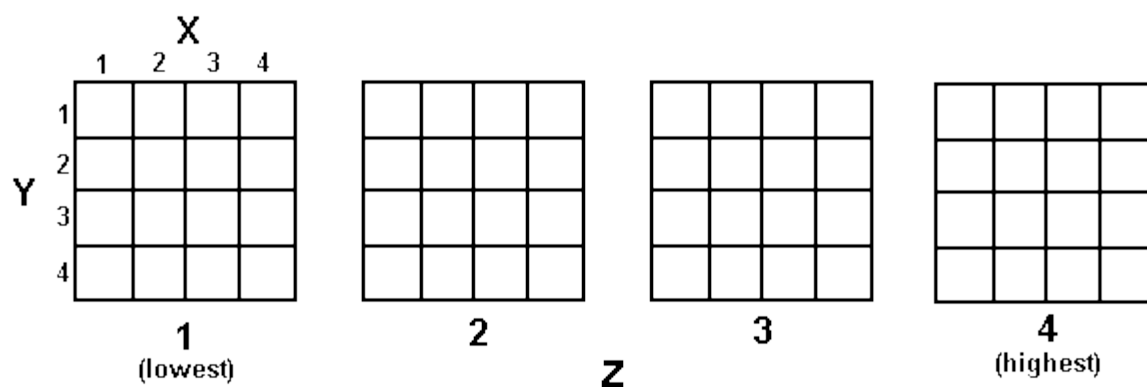
- 1) The board consists of 4 levels, each of which is 4 rows deep and 4 columns across, making a total of 64 squares on a 3 dimensional board.
- 2) Two players move alternately by placing a black or white marker on a square (or making an X or a O on a paper layout of the board). Once a move is made, the piece is never moved or removed. In this game, the human always goes first.
- 3) A player wins by placing four markers in a straight line. The line can lie in more than one level, and diagonals are perfectly legitimate wins.

In short, the game is played just like regular Tic Tac Toe, except that the board has one additional dimension, and is one square bigger in all dimensions. Unlike regular Tic Tac Toe, there is no known winning strategy for the 3-D version. It is a much more complex game which can require considerable skill in a player, allowing for very complicated strategies.

The 41C plays and remembers the game by dividing the board into its 16 component rows and storing an entire row in one register. The registers R0 through R15 are reserved for the game board.

Each square on the board can be characterized by its level=z, its row=y and its column=x. x,y, and z can have values from 1 through 4. When entering moves, make sure they are 3 digit numbers. All three digits must be between 1 and 4 inclusive. Entering a move outside this range may cause the program to make erroneous entries in the board.

The boards look like:



Note: Requires 2 Memory Modules for use on the HP-41C.

Instructions

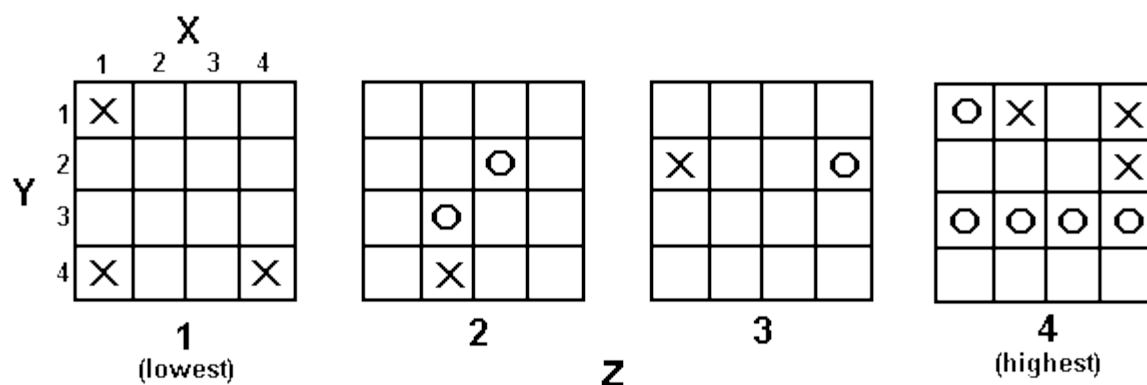
Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Enter program			
2	Initialize		[XEQ] 3DTT	READY
3	Key in your move	xyz	[R/S]	MY MOVE:
	Repeat 3 until you win or lose			

Example

Keystrokes: Display:

[XEQ] [ALPHA] SIZE	
[ALPHA] 026	
[XEQ] [ALPHA]	
3DTT [ALPHA]	READY
242 [R/S]	MY MOVE: 322
414 [R/S]	MY MOVE: 134
123 [R/S]	MY MOVE: 234
441 [R/S]	MY MOVE: 423
141 [R/S]	MY MOVE: 232
214 [R/S]	MY MOVE: 114
424 [R/S]	MY MOVE: 434
111 [R/S]	334, I WIN

The boards look like:



Program listing:

01 LBL "3DTT"

02 FIX 00	32 X#0?	64 XEQ 10	96 LASTX
03 CLRG	33 AVIEW	65 XEQ 04	97 FRC
04 CF 00	34 X#0?	66 FS?C 02	98 .5
05 CF 01	35 GTO A	67 GTO 03	99 -
06 CF 02	36 5	68 RCL 19	100 CHS
07 CF 03	37 RCL 20	69 XEQ 10	101 20
08 CF 29	38 /	70 LBL 03	102 *
09 "READY"	39 1	71 RCL 16	103 10^X
10 AVIEW	40 +	72 INT	104 STO 20
11 LBL A	41 ST+ IND 25	73 1	105 RDN
12 STOP	42 RCL 18	74 -	106 1
13 1 E3	43 X=0?	75 3	107 -
14 /	44 GTO 01	76 /	108 STO 21
15 STO 19	45 SF 00	77 21	109 X<>Y
16 0	46 XEQ 04	78 +	110 1
17 STO 16	47 CF 02	79 ENTER	111 -
18 2	48 CF 03	80 FRC	112 4
19 STO 17	49 CF 00	81 X=0?	113 *
20 RDN	50 .06	82 CF 01	114 STO 22
21 RDN	51 RCL 18	83 .5	115 +
22 XEQ 10	52 FRC	84 X>Y?	116 STO 23
23 STO 25	53 STO 18	85 SF 00	117 RTN
24 RCL 20	54 X>Y?	86 GTO IND Z	118 LBL 04
25 RCL IND 25	55 GTO 02	87 LBL 10	119 1
26 *	56 X=Y?	88 10	120 RCL 22
27 INT	57 GTO 03	89 *	121 XEQ 01
28 1 E2	58 LBL 01	90 INT	122 4
29 /	59 1 E-2	91 LASTX	123 RCL 21
30 FRC	60 ST+ 18	92 FRC	124 XEQ 01
31 "ILLEGAL	61 SF 02	93 10	125 5
MOVE"	62 GTO 03	94 *	126 ENTER
	63 LBL 02	95 INT	127 0

128 XEQ 01	176 +	224 RCL 22	272 STO 25
129 3	177 X<> 25	225 GTO 01	273 1
130 ENTER	178 FS? 01	226 LBL 22	274 STO 24
131 XEQ 01	179 *	227 4	275 XEQ 07
132 0	180 RCL IND 25	228 RCL 21	276 2
133 STO 24	181 +	229 GTO 01	277 XEQ 07
134 RCL 23	182 FS? 01	230 LBL 23	278 3
135 STO 25	183 GTO 01	231 5	279 XEQ 07
136 GTO 02	184 R^	232 ENTER	280 0
137 LBL 01	185 RCL 20	233 0	281 XEQ 08
138 CF 01	186 /	234 GTO 01	282 RCL 22
139 STO 25	187 /	235 LBL 24	283 4
140 RDN	188 LBL 01	236 3	284 STO 24
141 STO 24	189 FRC	237 ENTER	285 *
142 XEQ 02	190 R^	238 GTO 01	286 STO 23
143 SF 01	191 *	239 LBL 25	287 16
144 RCL 24	192 INT	240 SF 01	288 STO 20
145 CHS	193 4	241 0	289 1
146 STO 24	194 X<>Y	242 RCL 23	290 XEQ 08
147 XEQ 02	195 FS? 00	243 LBL 01	291 2
148 RCL 24	196 GTO 01	244 STO 25	292 XEQ 08
149 CHS	197 X>Y?	245 RDN	293 3
150 STO 24	198 RTN	246 STO 24	294 XEQ 08
151 LBL 02	199 GTO 02	247 1 E2	295 0
152 1	200 LBL 01	248 ENTER	296 XEQ 08
153 ST+ 18	201 5	249 1 E8	297 RCL 22
154 1 E2	202 /	250 FS? 00	298 RCL 23
155 ENTER	203 "YOU WIN"	251 X<>Y	299 +
156 ENTER	204 X=Y?	252 FS? 01	300 STO 25
157 RCL IND 25	205 PROMPT	253 STO 20	301 LBL 09
158 X<> 25	206 FRC	254 RCL 20	302 RCL IND 25
159 RCL 24	207 X#0?	255 LBL 05	303 RCL 24
160 +	208 RTN	256 XEQ 19	304 X<=Y?
161 X<> 25	209 LASTX	257 X<> 25	305 GTO 01
162 FS? 01	210 LBL 02	258 RCL 24	306 RCL 25
163 *	211 RCL 17	259 +	307 2
164 RCL IND 25	212 X>Y?	260 X<> 25	308 /
165 +	213 RTN	261 1 E-2	309 FRC
166 X<> 25	214 X<>Y	262 FS? 00	310 X=0?
167 RCL 24	215 SF 02	263 1/X	311 GTO 02
168 +	216 STO 17	264 RCL 20	312 GTO 03
169 X<> 25	217 FC? 00	265 FS? 01	313 LBL 01
170 FS? 01	218 SF 03	266 *	314 RCL 25
171 *	219 RCL 18	267 GTO 05	315 +
172 RCL IND 25	220 STO 16	268 LBL 20	316 16
173 +	221 RTN	269 16	317 X>Y?
174 X<> 25	222 LBL 21	270 STO 20	318 CLX
175 RCL 24	223 1	271 4	319 -

320 STO 25	349 RCL 20	378 /	407 10
321 GTO 09	350 X<=Y?	379 FRC	408 *
322 LBL 08	351 RTN	380 X#0?	409 +
323 STO 25	352 RDN	381 RTN	410 CLA
324 LBL 07	353 STO 20	382 RCL 20	411 ARCL X
325 RCL IND 25	354 RDN	383 1/X	412 1 E3
326 X<> 25	355 STO 22	384 ST+ IND 25	413 /
327 RCL 24	356 RTN	385 LOG	414 FS?C 02
328 +	357 LBL 02	386 2	415 GTO 01
329 X<> 25	358 1 E2	387 /	416 STO 18
330 RCL IND 25	359 XEQ 19	388 5	417 RCL 17
331 +	360 1 E4	389 +	418 3
332 X<> 25	361 XEQ 19	390 RCL 25	419 X>Y?
333 RCL 24	362 1 E8	391 4	420 GTO 01
334 +	363 XEQ 19	392 /	421 FC? 03
335 X<> 25	364 LBL 03	393 INT	422 GTO 01
336 RCL IND 25	365 1 E6	394 LASTX	423 " -, I WIN"
337 +	366 XEQ 19	395 FRC	424 AVIEW
338 X<> 25	367 1 E4	396 4	425 GTO A
339 RCL 24	368 XEQ 19	397 *	426 LBL 01
340 +	369 1 E2	398 1	427 ASTO X
341 X<> 25	370 XEQ 19	399 ST+ IND 25	428 "MY MOVE: "
342 RCL IND 25	371 1 E8	400 +	429 ARCL X
343 +	372 LBL 19	401 X<>Y	430 AVIEW
344 X<> 25	373 STO 20	402 1	431 GTO A
345 RCL 24	374 RCL IND 25	403 +	432 END
346 +	375 *	404 10	
347 X<> 25	376 INT	405 *	
348 INT	377 1 E2	406 +	

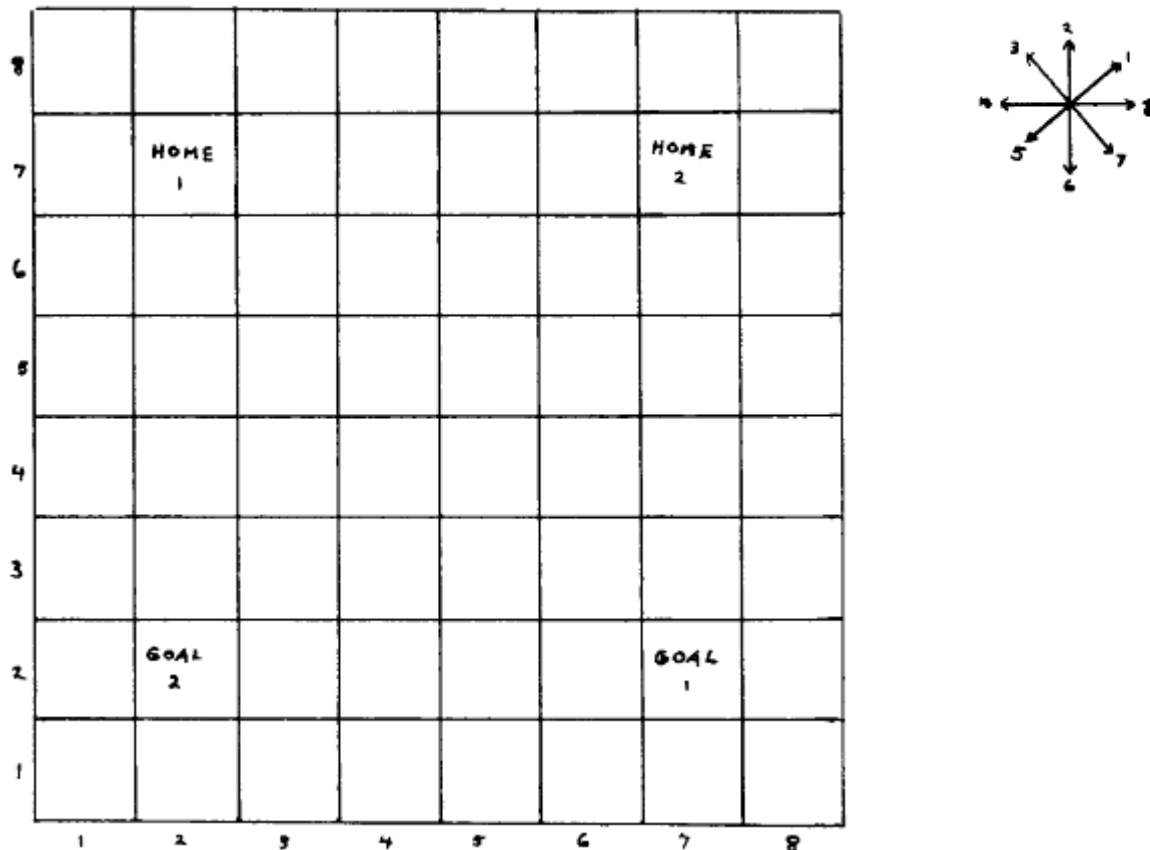
3-Ways Game

[L. Stein – PPCCJ V9N4 p57; \(May/July 1982\)](#)

COMPLETE INSTRUCTIONS FOR 3WAYS

1. XEQ "3WAYS" or press USER B
2. Enter ANY seed for the pseudorandom number generator
3. 64 tiles are chosen by the program. They include 2 "home" tiles, located at coordinates 2,7 and 7,7 on the 8x8 board; 2 "goal" tiles, located at 7,2 and 2,2; 4 blank tiles, located randomly; and 56 combinations of 8 directions taken 3 at a time, located randomly. The number seen during the process of playing board setup is the number of the tile currently being assigned a position on the playing board. (This process slows down as the board gets filled.) A BEEP signals that the board is complete.
4. Enter identifications for the two players, one at a time as prompted, if the default IDs of "ONE" and "TWO" are not desired. More than 3 characters causes later prompts to scroll.
5. Enter a move subject to the following rules:
 - There are 8 move directions, numbered 1 through 8 at 45 degree increments counter-clockwise around the circle beginning at 45 degrees in the upper right-hand quadrant. A move is made by entering the move direction II and hitting R/S.
 - The 1st move may not be made diagonally directly toward the goal; to do so would give the 1st player too much of an advantage. Any other 1st move is OK.
 - After the 1st move, only the 3 directions indicated by the opposing player's tile are allowed.
 - A move must be made if any move is possible. The move prompt shows the opposing player's tile contents and the current player's ID.
 - Illegal moves include
 - i). Moving off any edge of the playing board
 - ii). Moving onto a blank tile. Home tiles are considered to be blank.
 - iii). Moving onto the opposing player's goal
 - iv). Moving onto a square already occupied by the opposing player.
 - f). If no move is possible, or to change the starting player, enter a move of zero.
6. While a tile is being "uncovered", the coordinates of the tile are displayed.
7. Play alternates until a player reaches the goal on the diagonally opposite corner of the playing board.
8. Key A re-prompts for the same move.
9. Key B restarts the game with new tile positions (except for home and goal tiles).

10. Key M restarts the game with the same tile positions. A particular seed produces the same tile arrangement each time it is used



MISCELLANEOUS REMARKS FOR 3WAYS

3WAYS is based on the commercially available (from Aladdin Industries, Inc. PO Box 10444, 703 Murfreesboro Rd., Nashville, Tennessee 37210) game called "Trippples". 3WAYS is one possible implementation of the very flexible rules which come with the game.

In general, the player with the best memory has an advantage if a physical playing board is not used. This advantage is nullified if a playing board, such as the one included here, is used with transparent place-markers (so tile contents can be seen through them), and if each tile's contents is written in the appropriate square as it becomes known.

The game, as it stands, requires 3 HP41C Memory Modules. It is not optimized for any particular parameter, but is a compromise for speed, bytes and features, while using a very basic HP41C system. One thing is paid particular attention: it is almost foolproof. All stated rules are implemented except that a "pass" may be made at any time instead of its being automatic when forced. Forced moves are not made automatically either.

Many improvements are possible when the PPC ROM, Synthetic Programming, the printer, X functions, etc. are used. In particular, I can think of appropriate uses, off-hand, for PPC ROM routines BC, DR, IR, RK, RN, SE, SK and TN.

Specific improvements needed, besides speed, bytes and printing mentioned earlier, include automatic forced and "pass" moves, which would be a big step toward implementing player-

vs. HP41 games; a simple-minded machine strategy would be to minimize the radial distance to the goal among all possible moves available at the time. A little interest could be thrown in by arbitrating randomly ties resulting from the above strategy. An added strategy could be to avoid losing-moves, if possible. One might also want to program the avoidance of "taking chances" (moving to an "unknown" tile), wherever possible in the event that the human player is adjacent to the goal. Increasing the distance between home and goal tiles might be desired, as well.

Flag Status:

REG. #	CONTENTS	FLAG	MEANING/REASON USED
81	1.1	01	#1's move
82	0.1	02	#2's move
83	-0.9	06	Blank tile entry in progress
84	-1.0	07	1st move in progress
85	-1.1	08	Playing board has been set up
86	-0.1	09	2nd move in progress
87	0.9	22	Tested for illegal seed entry
88	1.0	27	USER off during seed entry
		28	Comma used in display
		29	Omits extraneous commas

3-Ways Data Register Usage

REG. #	USAGE	REG. #	USAGE
00	Scratch counter	01	Tile at pos'n 1,1
02	Tile at pos'n 1,2	03	1,3
04	1,4	05	1,5
06	1,6	07	1,7
08	1,8	09	2,1
10	#2's goal tile	11	2,3
12	Tile at pos'n 2,4	13	2,5
14	2,6	15	#1's home tile
16	2,8	17	Tile at pos'n 3,1
18	3,2	19	3,3
20	3,4	21	3,5
22	3,6	23	3,7
24	3,8	25	4,1
26	4,2	27	4,3
28	4,4	29	4,5
30	4,6	31	4,7
32	4,8	33	5,1
34	5,2	35	5,3
36	5,4	37	5,5
38	5,6	39	5,7
40	5,8	41	6,1
42	6,2	43	6,3
44	6,4	45	6,5
46	6,6	47	6,7
48	6,8	49	7,1
50	#1's goal tile	51	7,3
52	Tile at pos'n 7,4	53	7,5
54	7,6	55	#2's home tile
56	7,8	57	Tile at pos'n 8,1
58	8,2	59	8,3
60	8,4	61	8,5
62	8,6	63	8,7
64	8,8	65	" MOVE"

REG. #	USAGE DURING SETUP	USAGE DURING MAIN PGM.
66	MSD of current tile	Current tile #
67	Middle digit of current tile	#1's current tile #
68	LSD of current tile	#2's current tile #
69	Random tile #	#1's 1st legal dir.
70	Seed	#1's 2nd legal dir.
71	Current tile contents	#1's 3rd legal dir.
72	ID #1	ID #1
73	ID #2	ID #2
74	Current tile #	#1's current pos'n
75		#2's current pos'n
76		#2's 1st legal dir.
77		#2's 2nd legal dir.
78		#2's 3rd legal dir.
79		Tentative new pos'n
80		Scratch storage

Program listing:

01*LBL "3WAYS"	36 CF 27	71 -4	106 GTO 02
02*LBL 10	37 PROMPT	72 STO 55	107 RDN
03 FS? 08	38 FC?C 22	73 ISG 74	108 STO IND 69
04 GTO 03	39 GTO 10	74 1.006	109 ISG 74
05 "SIZE=089?"	40 SF 27	75 STO 66	110 ISG 00
06 AVIEW	41 RAD	76 2.007	111 GTO 02
07 " MOVE"	42 SIN	77 STO 67	112 CF 06
08 ASTO 65	43 ABS	78 3.008	113 GTO 02
09 "ONE"	44 DEG	79 STO 68	114*LBL 00
10 ASTO 72	45 ASIN	80 SF 06	115 RCL IND 69
11 "TWO"	46 90	81 5.008	116 X=0?
12 ASTO 73	47 /	82 STO 00	117 GTO "S"
13 1.1	48 STO 70	83*LBL 02	118 ISG 69
14 STO 81	49 1.064	84 VIEW 74	119 GTO 00
15 .1	50 STO 00	85 RCL 66	120 XROM "R"
16 STO 82	51 STO 74	86 INT	121 GTO 00
17 -.9	52 FIX 0	87 E2	122*LBL "S"
18 STO 83	53 0	88 *	123 RCL 71
19 -1	54*LBL 01	89 RCL 67	124 STO IND 69
20 STO 84	55 STO IND 00	90 INT	125 XROM "NT"
21 -1.1	56 ISG 00	91 E1	126 GTO 02
22 STO 85	57 GTO 01	92 *	127*LBL "MP"
23 -.1	58 VIEW 74	93 +	128 AOFF
24 STO 86	59 -1	94 RCL 68	129 FC? 08
25 .9	60 STO 50	95 INT	130 GTO 10
26 STO 87	61 ISG 74	96 +	131 BEEP
27 E	62 VIEW 74	97 STO 71	132 1.002
28 STO 88	63 -2	98 XROM "R"	133 STO 00
29*LBL 03	64 STO 10	99 XROM "RN"	134 FIX 0
30 CF 28	65 ISG 74	100 FC? 06	135 "NEW ID?"
31 CF 29	66 VIEW 74	101 GTO 00	136 AVIEW
32 CF 01	67 -3	102 RCL 00	137 PSE
33 CF 02	68 STO 15	103 CHS	138 "N=0, Y=1."
34 AOFF	69 ISG 74	104 RCL IND 69	139 E
35 "SEED? "	70 VIEW 74	105 X<0?	140 PROMPT

141 X=Y?	194 X<0?	247*LBL 08	300 INT
142 GTO 06	195 GTO 11	248 RCL 80	301 STO 80
143 X=0?	196 INT	249 RCL IND 00	302 69
144 GTO 05	197 LASTX	250 X=Y?	303 FS? 01
145 GTO "MP"	198 X#Y?	251 GTO 09	304 76
146*LBL 06	199 GTO 11	252 ISG 00	305 RCL 80
147 "ID NO. "	200 STO 80	253 GTO 08	306 STO IND Y
148 ARCL 00	201 8	254 GTO 11	307 LASTX
149 AON	202 X<>Y	255*LBL 09	308 FRC
150 PROMPT	203 X>Y?	256 7.2	309 E1
151 RCL 00	204 GTO 11	257 FS? 01	310 *
152 71	205 FS? 07	258 2.2	311 INT
153 +	206 XROM "FM"	259 RCL 79	312 STO 80
154 ASTO IND X	207 75	260 X=Y?	313 70
155 ISG 00	208 FS? 01	261 GTO "WG"	314 FS? 01
156 GTO 06	209 74	262 2.2	315 77
157 AOFF	210 RCL IND X	263 FS? 01	316 RCL 80
158*LBL 05	211 RCL 80	264 7.2	317 STO IND Y
159 SF 07	212 80	265 RCL 79	318 LASTX
160 CF 09	213 +	266 X=Y?	319 FRC
161 15	214 X<>Y	267 GTO "WN"	320 E1
162 STO 67	215 RCL IND Y	268 RCL 79	321 *
163 55	216 +	269 INT	322 STO 80
164 STO 68	217 STO 79	270 E	323 71
165 2.7	218 FIX 1	271 -	324 FS? 01
166 STO 74	219 VIEW 79	272 8	325 78
167 7.7	220 FIX 0	273 *	326 RCL 80
168 STO 75	221 INT	274 RCL 79	327 STO IND Y
169 SF 01	222 X=0?	275 FRC	328 CF 09
170 CF 02	223 GTO 12	276 E1	329 FS? 07
171*LBL "EM"	224 8	277 *	330 SF 09
172 FC? 08	225 X<Y?	278 +	331 CF 07
173 GTO 10	226 GTO 12	279 STO 66	332*LBL "SW"
174 CLA	227 RCL 79	280 RCL IND X	333 FS? 01
175 FS? 07	228 FRC	281 X<0?	334 GTO 04
176 GTO 07	229 X=0?	282 GTO "BT"	335 SF 01
177 67	230 GTO 12	283 75	336 CF 02
178 FS? 01	231 .8	284 FS? 01	337 GTO "EM"
179 68	232 X<Y?	285 74	338*LBL "IM"
180 RCL IND X	233 GTO 12	286 RCL 79	339*LBL 11
181 ARCL IND X	234 74	287 STO IND Y	340 "ILLEGAL"
182*LBL 07	235 FS? 01	288 68	341 FS? 07
183 " " "	236 75	289 FS? 01	342 " 1ST"
184 73	237 RCL IND X	290 67	343 FS? 09
185 FS? 01	238 RCL 79	291 RCL 66	344 " 2ND"
186 72	239 X=Y?	292 STO IND Y	345 ARCL 65
187 ARCL IND X	240 GTO "OC"	293 68	346 AVIEW
188 ARCL 65	241 FS? 07	294 FS? 01	347 PSE
189 PROMPT	242 GTO 09	295 67	348 AOFF
190 FS? 09	243 76.078	296 RCL IND X	349 GTO "EM"
191 XROM "SM"	244 FS? 01	297 RCL IND X	350*LBL "SM"
192 X=0?	245 69.071	298 E2	351 X#0?
193 GTO "SW"	246 STO 00	299 /	352 RTN

353 15	382 TONE 6	409 GTO 11	437 1.064
354 FS? 01	383 TONE 9	410*LBL "WG"	438 STO 69
355 55	384 "NEW GAME?"	411 "WRONG	439 RTN
356 RCL IND X	385 AVIEW	GOAL"	<u>440*LBL "NT"</u>
357 68	386 PSE	412 AVIEW	441 ISG 74
358 FS? 01	387 "M=SAME	413 PSE	442 X<> X
359 X<>Y	BOARD,"	414 GTO 11	443 ISG 68
360 FS? 01	388 AVIEW	<u>415*LBL "FM"</u>	444 RTN
361 67	389 PSE	416 5	445 ISG 67
362 RCL IND X	390 "B=NEW	417 FS? 01	446 GTO "R8"
363 RCL Z	BOARD."	418 7	447 ISG 66
364 X=Y?	391 PROMPT	419 RCL 80	448 GTO "R7"
365 GTO 11	392 GTO 10	420 X=Y?	449 SF 08
366 RTN	<u>393*LBL "OC"</u>	421 GTO 11	450 GTO "MP"
367*LBL 04	394 "OCCUPIED"	422 RTN	<u>451*LBL "RN"</u>
368 CF 01	395 AVIEW	<u>423*LBL "R8"</u>	452 RCL 70
369 SF 02	396 PSE	424 RCL 67	453 9821
370 GTO "EM"	397 GTO 11	425 1.001	454 *
<u>371*LBL "WN"</u>	398*LBL "OB"	426 +	455 .211327
372 73	399*LBL 12	427 STO 68	456 +
373 FS? 01	400 "OFF BOARD"	428 RTN	457 FRC
374 72	401 AVIEW	<u>429*LBL "R7"</u>	458 STO 70
375 CLA	402 PSE	430 RCL 66	459 64
376 ARCL IND X	403 GTO 11	431 1.001	460 *
377 "` WINS"	<u>404*LBL "BT"</u>	432 +	461 INT
378 AVIEW	405 TONE 5	433 STO 67	462 ST+ 69
379 TONE 3	406 "BLANK TILE"	434 XROM "R8"	463 END
380 TONE 6	407 AVIEW	435 RTN	
381 TONE 9	408 PSE	<u>436*LBL "R"</u>	

Matrix Game for HP-41CX

Kai Schröder, http://www.achim-und-kai.de/kai/hp41cx/matrixspiel_e.html

Game theory is a fascinating field and one topic are the so-called *two person zero-sum games*. A game is called a *zero-sum game*, if the gains of one player is the loss of the other one and vice versa. If the game is played sufficiently - infinity times - often, loss and gains of both players will become zero. The stimulus now is, that the game isn't played infinity but only finite times. In this case there is a winner and a loser.

The playing field here is a matrix, whose elements are covered with different positive and negative numbers, which are generated with a random number generator. Now the player looks for a row, which is "advantageous" to him, and the HP-41CX calculates a column, respectively. If the so defined element is positive the player wins this round. In case it is negative the HP-41CX is the winner. If the element equals zero, the game ends drawn.

In the simple method used by me the game can have a "value" different from zero. Now, if you would play this game ad infinitum, one of the players will win, and from this fact the "value" of the game is determined. Ideally no one would win - this would be a "zero"-sum game.

An Example For Better Understanding:

Let the following matrix be given:

	C ₁	C ₂
r ₁	1	0
r ₂	1	-1

In this case row 1 would be advantageous to the player, because one time he wins ($[r_1, c_1] = 1$) and in the other the game would end in a draw ($[r_1, c_2] = 0$). If he chooses row 2 he would win if his opponent chooses column 1 ($[r_2, c_1] = 1$), but he loses if the other one chooses column 2 ($[r_2, c_2] = -1$).

From the point of view of the other player column 2 is the right one, for the game ends drawn ($[r_1, c_2] = 0$) or he wins ($[r_2, c_2] = -1$). Column 1 is not acceptable for him, because he loses in both cases.

In the above example for the first player it's advantageous to choose row 1, and for the other one column 2 is favorable. If both players make their choices concerning these considerations the game will end in a draw ($[r_1, c_2] = 0$). This matrix is more advantageous to the first player, for he wins in two cases, loses in one and in one other the game ends drawn - correspondingly the other player is in disadvantage.

The above matrix is a very simple one. It's of more interest and more challenging, if the matrix becomes larger in size and the number's values increase. Then it's no longer easy to decide which row or column is the most advantageous or which chooses the other player !

Interestingly, there exists an optimal strategy for solving this problem (and the program masters it, of course ;-)) ! Noteworthy is, that a choice is possible, which on first sight looks "unfavorable". But, of course, one has to take into account the possible opponent's choices and the arising consequences ! ;-)

I don't want to go into detail here concerning the theory of *two person zero-sum games*. (In case I should have much time I will write some lines. ;-)) There is plenty of literature available, for example G. Owen, **Game Theory**, [Springer Verlag](#).

Course of Game:

Starting the program the first matrix is generated. In the display appears "1. MATRIX" and then "1. ROW:". Now the elements of this row are displayed separated by commas. These numbers must be written on a sheet of paper by the player, because the HP-41CX's display cannot contain the whole matrix but only one row per time. ;-) If the matrix is fully constructed "READY" is displayed.

Now the HP-41CX calculates the column which is the most favorable for it and the player makes his choice of a row, respectively. After a new BEEP the row must be entered. Then it is displayed, which column was computed by the HP-41CX and afterwards the element determined by row and column. If the value of this element is positive the player wins this round. In case the value is zero "DRAW" is displayed. The HP-41CX wins, if the value is negative. In the last case the game ends at once and the number of tries and the degree of difficulty are displayed.

In the first or second case the player is asked, whether he wants to continue. In the display appears "CONTIN ? Y/N". Now "Y" must be pressed to continue the game, otherwise "N". If the last round ended in a draw, a new matrix of same level is generated. If the player had won the last round, the difficulty is increased. There are 21 levels (degrees of difficulty) for the matrices.

ERWSP must be the first program in extended memory (in CAT 4 ERWSP must be first displayed). There is *no* need for jumping by hand to any addresses !

After ERWSP is copied into extended memory the main program GAME9 can be loaded. Before starting the program **SIZE 122** must be executed. Additionally, GAME9 creates the data file "DATA" with size 28 registers in extended memory, which is removed at program termination.

Program listing:

01 LBL "ERWSP"	15 2	29 4	43 LBL 02
02 RCL 04	16 X=Y?	30 X#Y?	44 ISG 07
03 2	17 GTO 02	31 GTO 01	45 ""
04 X#Y?	18 ISG 04	32 RCL 07	46 7
05 GTO 01	19 ""	33 3	47 STO 06
06 ISG 04	20 GTO "HP"	34 X=Y?	48 GTO "HP"
07 ""	21 LBL 02	35 GTO 02	49 LBL 01
08 GTO "HP"	22 ISG 07	36 ISG 04	50 X<>Y
09 LBL 01	23 ""	37 ""	51 5
10 X<>Y	24 5	38 5	52 X#Y?
11 3	25 STO 06	39 STO 06	53 GTO 01
12 X#Y?	26 GTO "HP"	40 3	54 RCL 07
13 GTO 01	27 LBL 01	41 STO 07	55 3
14 RCL 07	28 X<>Y	42 GTO "HP"	56 X=Y?

57 GTO 02	120 7	183 LBL 01	23 CF 29
58 ISG 04	121 STO 06	184 ISG 07	24 26
59 ""	122 GTO "HP"	185 ""	25 SEEKPT
60 3	123 LBL 02	186 9	26 TIME
61 STO 07	124 ISG 07	187 STO 06	27 E2
62 5	125 ""	188 GTO "HP"	28 /
63 STO 06	126 9	189 LBL 11	29 SAVEX
64 GTO "HP"	127 STO 06	190 RCL 07	<u>30 LBL "HP"</u>
65 LBL 02	128 GTO "HP"	191 4	31 2,025
66 ISG 07	129 LBL 01	192 X=Y?	32 STO 10
67 ""	130 ISG 07	193 GTO 01	33 LBL 14
68 7	131 ""	194 X<>Y	34 RCL 10
69 STO 06	132 7	195 5	35 SEEKPT
70 GTO "HP"	133 STO 06	196 X=Y?	36 ,
71 LBL 01	134 GTO "HP"	197 GTO 02	37 SAVEX
72 X<>Y	135 LBL 11	198 TONE 0	38 ISG 10
73 6	136 X<>Y	199 TONE 3	39 GTO 14
74 X#Y?	137 8	200 TONE 5	40 27
75 GTO 11	138 X#Y?	201 TONE 7	41 SEEKPT
76 RCL 07	139 GTO 01	202 TONE 9	42 GETX
77 3	140 RCL 07	203	43 ISG X
78 X=Y?	141 4	"CONGRATS"	44 ""
79 GTO 01	142 X=Y?	204 AVIEW	45 X<>Y
80 X<>Y	143 GTO 02	205 PSE	46 SEEKPT
81 4	144 ISG 04	206 PSE	47 X<>Y
82 X=Y?	145 ""	207 GTO "END"	48 SAVEX
83 GTO 02	146 4	208 LBL 02	49 SIGN
84 ISG 04	147 STO 07	209 ISG 07	50 STO 08
85 ""	148 7	210 ""	51 RCL 05
86 3	149 STO 06	211 11	52 CLA
87 STO 07	150 GTO "HP"	212 STO 06	53 " "
88 5	151 LBL 02	213 GTO "HP"	54 ARCL X
89 STO 06	152 ISG 07	214 LBL 01	55 "└.
90 GTO "HP"	153 ""	215 ISG 07	MATRIX:"
91 LBL 02	154 9	216 ""	56 AVIEW
92 ISG 07	155 STO 06	217 9	57 RCL 04
93 ""	156 GTO "HP"	218 STO 06	58 X^2
94 9	157 LBL 01	219 GTO "HP"	59 LASTX
95 STO 06	158 X<>Y	220 END	60 +
96 GTO "HP"	159 9		61 E1
97 LBL 01	160 XY#?	<u>01 LBL "GAME9"</u>	62 +
98 ISG 07	161 GTO 11	02 "MATRIX	63 E3
99 ""	162 RCL 07	GAME"	64 /
100 7	163 4	03 AVIEW	65 11
101 STO 06	164 X=Y?	04 CLRG	66 +
102 GTO "HP"	165 GTO 01	05 CF 05	67 STO 09
103 LBL 11	166 X<>Y	06 "DATA"	68 RCL 04
104 X<>Y	167 5	07 28	69 E3
105 7	168 X=Y?	08 CRFLD	70 /
106 X#Y?	169 GTO 02	09 E	71 ISG X
107 GTO 11	170 ISG 04	10 SEEKPT	72 STO 10
108 RCL 07	171 ""	11 11	73 LBL 03
109 3	172 4	12 SAVEX	74 RCL 08
110 X=Y?	173 STO 07	13 X<>Y	75 " "
111 GTO 01	174 7	14 STO 05	76 ARCL X
112 X<>Y	175 STO 06	15 ISG X	77 "└. ROW :"
113 4	176 GTO "HP"	16 ""	78 AVIEW
114 X=Y?	177 LBL 02	17 STO 04	79 CLA
115 GTO 02	178 ISG 07	18 STO 07	80 LBL 00
116 ISG 04	179 ""	19 ISG X	81 XEQ 16
117 ""	180 11	20 ""	82 RCL 06
118 4	181 STO 06	21 STO 06	83 MOD
119 STO 07	182 GTO "HP"	22 FIX 0	84 ISG X

85	""	144	RCL 02	206	RTN	269	STO O
86	STO IND 09	145	X>Y?	207	LBL 34	270	X<>Y
87	RCL 07	146	GTO 40	208	STO M	271	MOD
88	-	147	RCL 03	209	RDN	272	ST- O
89	ARCL X	148	XEQ 37	210	XEQ 38	273	LASTX
90	LASTX	149	X<> Z	211	X<>Y	274	ST/ O
91	+	150	XEQ 37	212	XEQ 38	275	CLX
92	DSE X	151	E3	213	RCL M	276	X<> O
93	""	152	/	214	SIGN	277	y<>Y
94	RCL 08	153	+	215	LBL 05	278	ISG Y
95	15	154	RCL 04	216	RDN	279	""
96	+	155	ISG X	217	RCL IND Y	280	ISG X
97	,	156	""	218	LAST X	281	""
98	SEEKPT	(NOP)		219	*	282	RTN
99	X<>Y	157	E5	220	ST+ IND Y	283	LBL 37
100	SAVEX	158	/	221	ISG Y	284	X<> 04
101	RCL Z	159	+	222	""	285	ISG X
102	RCL 10	160	XEQ 39	223	ISG Z	286	""
103	INT	161	RCL IND M	224	GTO 05	287	ST- 04
104	XEQ 18	162	ST* 00	225	RTN	288	*
105	ISG 10	163	X=0?	226	LBL 35	289	ST+ 04
106	GTO 01	164	GTO 31	227	XEQ 38	290	X<> L
107	GTO 02	165	1/X	228	X<>Y	291	DSE X
108	LBL 01	166	RCL M	229	XEQ 38	292	X<> 04
109	"┐,"	167	INT	230	INT	293	E1
110	ISG 09	168	XEQ 36	231	E3	294	+
111	GTO 00	169	RDN	232	/	295	RTN
112	LBL 02	170	STO 01	233	X<>Y	296	LBL 39
113	TONE 5	171	XEQ 33	234	INT	297	STO M
114	AVIEW	172	RCL 01	235	+	298	STO N
115	ISG 08	173	ST- 01	236	RCL 04	299	STO O
116	""	174	RCL 02	237	ISG X	300	RCL IND X
117	ISG 09	175	X=Y?	238	""	301	ABS
118	SIGN	176	GTO 32	239	E6	302	ENTER^
119	STO IND	177	XEQ 35	240	/	303	ENTER^
09		178	RCL 00	241	+	304	RDN
120	RCL 04	179	CHS	242	REGSWAP	305	LBL 07
121	ST- 10	180	STO 00	243	RTN	306	CLX
122	STOP	181	LBL 32	244	LBL 38	307	RCL IND Z
123	ISG 09	182	ISG 01	245	RCL 04	308	ABS
124	GTO 03	183	""	246	ISG X	309	X>Y?
125	BEEP	184	RCL 04	247	""	310	GTO 01
126	"	185	RCL 01	248	*	311	R^
READY"		186	X>Y?	249	11	312	X>Y?
127	AVIEW	187	GTO 30	250	+	313	GTO 02
128	SIGN	188	RCL 02	251	RCL X	314	RDN
129	STO 00	189	X=Y?	252	RCL 04	315	LBL 06
130	1,003	190	GTO 32	253	ISG X	316	ISG Z
131	CLRGX	191	RCL 01	254	""	317	GTO 07
132	LBL 30	192	RCL 03	255	ST- Z	318	X<>Y
133	ISG 02	193	XEQ 37	256	SIGN	319	R^
134	LBL 31	194	RDN	257	-	320	RTN
135	ISG 03	195	RCL IND T	258	E3	321	LBL 01
136	""	196	CHS	259	/	322	X<>Y
(NOP)		197	XEQ 34	260	+	323	CLX
137	RCL 04	198	GTO 32	261	RTN	324	RCL Z
138	ISG X	199	LBL 33	262	LBL 36	325	STO M
139	""	200	XEQ 38	263	11	326	GTO 06
(NOP)		201	X<>Y	264	-	327	LBL 02
140	RCL 03	202	LBL 04	265	RCL 04	328	CLX
141	X>Y?	203	ST* IND Y	266	ISG X	329	RCL T
142	GTO 40	204	ISG Y	267	""	330	STO N
143	RCL 04	205	GTO 04	268	X<>Y	331	X<>Y

332	RDN	395	GETX	458	GETX	519	GTO 20
333	GTO 06	396	GETX	459	X>0?	520	BEEP
334	LBL 40	397	XEQ 15	460	GTO 02	521	X=0?
335	CLD	398	STO M	461	LBL 01	522	GTO 01
336	RCL 04	399	RDN	462	13	523	"
337	RCL X	400	STO N	463	SEEKPT	524	CONGRATS"
338	E3	401	RDN	464	GETX	525	GTO 02
339	/	402	4	465	LBL 02	526	LBL 01
340	ISG X	403	RCL 04	466	3	527	" DRAW"
341	STO 10	404	X<=Y?	467	SEEKPT	528	SF 05
342	ISG Y	405	GTO 02	468	X<>Y	529	LBL 02
343	"	406	-	469	LBL 12	530	AVIEW
344	X<>Y	407	CHS	470	GETX	531	PSE
345	STO 09	408	E3	471	X=Y?	532	AON
346	LBL 08	409	/	472	GTO 13	533	"CONT?"
347	RCL 09	410	ISG X	473	RDN	534	Y/N"
348	RCL 10	411	STO 10	474	ISG 09	535	PROMPT
349	INT	412	RDN	475	"	536	FS? 49
350	*	413	RCL N	476	GTO 12	537	GTO 21
351	E1	414	RCL M	477	LBL 13	538	78
352	+	415	LBL 10	478	BEEP	539	AOFF
353	RCL IND X	416	GETX	479	" ROW ?"	540	ATOX
354	RCL 10	417	XEQ 15	480	PROMPT	541	X=Y?
355	2	418	ISG 10	481	STO 10	542	GTO "END"
356	+	419	GTO 10	482	FC?C 22	543	FS?C 05
357	SEEKPT	420	GTO 01	483	GTO 13	544	GTO "HP"
358	X<>Y	421	LBL 02	484	X<=0?	545	ISG 05
359	SAVEX	422	R^	485	GTO 13	546	"
360	LASTX	423	R^	486	RCL 04	547	" " (BE)
361	XEQ 19	424	RCL N	487	X<Y?	548	ASTO b
362	+	425	RCL M	488	GTO 13	549	LBL 16
363	2	426	LBL 01	489	X<>Y	550	26
364	SEEKPT	427	E	490	15	551	SEEKPT
365	X<>Y	428	STO 09	491	+	552	GETX
366	SAVEX	429	RDN	492	,	553	R-D
367	ISG 10	430	13	493	SEEKPT	554	FRC
368	GTO 08	431	SEEKPT	494	X<>Y	555	R-D
369	RCL 04	432	RDN	495	SAVEX	556	FRC
370	ST- 10	433	SAVEX	496	" HP-	557	FS? 06
371	LBL 09	434	RDN	497	41CX:"	558	E^X
372	RCL 10	435	SAVEX	498	AVIEW	559	FRC
373	2	436	RDN	499	PSE	560	26
374	+	437	SAVEX	500	" COLUMN	561	SEEKPT
375	XEQ 19	438	XEQ 16	501	"	562	X<>Y
376	LASTX	439	9	502	ARCL 09	563	SAVEX
377	XEQ 19	440	MOD	503	AVIEW	564	E6
378	X#0?	441	2	504	PSE	565	*
379	1/X	442	X<=Y?	505	RCL 09	566	INT
380	*	443	GTO 01	506	XEQ 17	567	RTN
381	RCL 10	444	15	507	ISG X	568	LBL 17
382	2	445	SEEKPT	508	"	569	DSE X
383	+	446	GETX	509	RCL 07	570	"
384	SEEKPT	447	X<=0?	510	-	571	E
385	X<>Y	448	GTO 11	511	" ["	572	XEQ 19
386	SAVEX	449	GTO 02	512	ARCL 10	573	X<>y
387	ISG 10	450	LBL 01	513	"┐,"	574	Y^X
388	GTO 09	451	X<>Y	514	ARCL 09	575	,
389	RCL 04	452	5	515	"┐] = "	576	XEQ 19
390	ST- 10	453	X<=Y?	516	ARCL X	577	XEQ 19
391	3	454	GTO 01	517	AVIEW	578	X<>Y
392	SEEKPT	455	LBL 11	518	PSE	579	ST/ Y
393	GETX	456	14		PSE		X<>Y
394	GETX	457	SEEKPT		X<0?		INT

580 E	606 LBL 21	631 R^	655 X=Y?
581 XEQ 19	607 BEEP	632 RTN	656 GTO 01
582 MOD	608 "	633 LBL 20	657 "┐ TRIES"
583 RTN	BATTERY"	634 TONE 8	658 GTO 02
584 LBL 18	609 SF 11	635 TONE 7	659 LBL 01
585 XEQ 17	610 OFF	636 TONE 5	660 ">" TRY"
586 X<>Y	611 AVIEW	637 TONE 3	661 LBL 02
587 ST* Z	612 STOP	638 TONE 2	662 RCL 05
588 *	613 LBL 15	639 TONE 0	663 PSE
589 ,	614 X<Y?	640 TONE 10	664 AVIEW
590 XEQ 19	615 X<>Y	641 TONE 1	665 " " (1
591 XEQ 19	616 R^	642 "	space)
592 X<>Y	617 X<Y?	SORRY, "	666 ARCL X
593 -	618 X<>Y	643 PSE	667 ">".
594 +	619 R^	644 AVIEW	MATRIX"
595 ,	620 X<Y?	645 PSE	668 PSE
596 XEQ 19	621 X<>Y	646 LBL "END"	669 AVIEW
597 SEEKPT	622 RDN	647 "END OF	670 FIX 4
598 X<>Y	623 X<Y?	GAME"	671 SF 29
599 SAVEX	624 X<>Y	648 AVIEW	672 "DATA"
600 RTN	625 RDN	649 27	673 PURFL
601 LBL 19	626 X<Y?	650 SEEKPT	674 CF 06
602 SEEKPT	627 X<>Y	651 GETX	675 CLA
603 CLX	628 R^	652 CLA	676 CLRG
604 GETX	629 X<Y?	653 ARCL X	677 CLST
605 RTN	630 X<>Y	654 E	678 END

Labyrinth (French)

Whodunit – Swap Disks

No documentation is available. It uses the Card Reader (!)

Program listing:

01*LBL "LABY"	43 X=Y?	85 RCL 22
02 1.02	44 GTO 01	86 STO 24
03 "LABYRINTHE-1"	45 SF 05	87*LBL 04
04 AVIEW	46*LBL 02	88 E
05 XEQ 10	47 RCL 21	89 ST+ 29
06 XROM 30,03	48 E	90 "I"
07 XEQ 10	49 FS? 05	91 ARCL 25
08*LBL A	50 -	92 ARCL 26
09 SF 27	51 FC? 05	93 ARCL 27
10 "ATTENTION"	52 +	94 "I"
11 AVIEW	53 STO 23	95 ARCL 28
12 0	54 XEQ 11	96 "I "
13 STO 29	55 E	97 ARCL 29
14 STO 30	56 " "	98 TONE 0
15 FIX 0	57 X=Y?	99 AVIEW
16 CF 29	58 "*"	100*LBL 08
17*LBL 01	59 FS? 05	101 E
18 RCL 00	60 ASTO 25	102 ST+ 30
19 X<0?	61 FS?C 05	103 PSE
20 CHS	62 GTO 02	104 GTO 08
21 E	63 ASTO 27	105*LBL "DR"
22 X>Y?	64 RCL 21	106 SF 05
23 GTO 00	65 STO 23	107*LBL "GA"
24 "ALEA 0<X<1?"	66 SF 05	108 TONE 9
25 PROMPT	67*LBL 03	109 AVIEW
26 STO 00	68 RCL 22	110 RCL 21
27 GTO 01	69 E	111 E
28*LBL 00	70 FS? 05	112 FS? 05
29 1.008	71 -	113 +
30 XEQ 09	72 FC? 05	114 FC?C 05
31 STO 21	73 +	115 -
32 STO 23	74 STO 24	116 STO 23
33 1.018	75 XEQ 11	117 GTO 05
34 XEQ 09	76 E	118*LBL "AR"
35 STO 22	77 " "	119 SF 05
36 STO 24	78 X=Y?	120*LBL "DE"
37 XEQ 11	79 "*"	121 TONE 9
38 9	80 FS? 05	122 AVIEW
39 X=Y?	81 ASTO 26	123 RCL 22
40 GTO 01	82 FS?C 05	124 E
41 X<>Y	83 GTO 03	125 FS? 05
42 E	84 ASTO 28	126 +

127 FC?C 05	156 TONE 0	185 "CHARGER OFF"
128 -	157 BEEP	186 AVIEW
129 STO 24	158 RCL 29	187 TONE 0
130*LBL 05	159 E3	188 SF 11
131 XEQ 11	160 *	189 OFF
132 9	161 RCL 30	190 RTN
133 X=Y?	162 /	191*LBL 09
134 GTO 07	163 INT	192 FRC
135 X<>Y	164 "GAGNE,SCORE="	193 ST- L
136 E	165 ARCL X	194 E3
137 X=Y?	166 AVIEW	195 ST* Y
138 GTO 06	167 FIX 2	196 X<> L
139 RCL 23	168 SF 29	197 X>Y?
140 STO 21	169 STOP	198 X<>Y
141 RCL 24	170*LBL 11	199 -
142 STO 22	171 RCL 23	200 ISG Y
143 SF 05	172 E	201 STO X
144 GTO 02	173 -	202 RCL 00
145*LBL 06	174 10^X	203 ST* Y
146 "IMPOSSIBLE"	175 RCL IND 24	204 X<> L
147 5	176 *	205 +
148 ST+ 30	177 FRC	206 INT
149 AVIEW	178 E1	207 RCL 00
150 RCL 21	179 *	208 ACOS
151 STO 23	180 INT	209 FRC
152 RCL 22	181 RTN	210 STO 00
153 STO 24	182*LBL 10	211 RDN
154 GTO 04	183 FC? 49	212 END
155*LBL 07	184 RTN	

Domino

Whodunit – Swap Disks

No documentation is available. It uses the Card Reader (!)

Program listing:

01*LBL "DOMINO"	45 FC?C 22	89 GTO 04	133 RCL 18
02*LBL 11	46 GTO 02	90*LBL 03	134 STO [
03 12	47 ABS	91 XEQ 17	135 XEQ 16
04 XROM "INIT"	48 INT	92*LBL 04	136*LBL 03
05 "I* "	49 STO 19	93 DSE 07	137 FC?C 01
06 ASTO d	50 FS? 03	94 GTO B	138 GTO 05
07 7	51 STO 17	95 "YOU WIN"	139 XEQ 17
08 STO 07	52 LASTX	96 GTO 09	140 CF 02
09 STO 08	53 FRC	97*LBL 10	141*LBL 06
10 STO 09	54 E1	98 PROMPT	142 CF 03
11 ST+ 09	55 *	99 ABS	143 "ME:"
12 E3/E+	56 STO 20	100 INT	144 ARCL 19
13 STO 17	57 FS? 03	101 STO [145 >","
14*LBL 00	58 STO 18	102 LASTX	146 ARCL 20
15 CLA	59 LASTX	103 FRC	147 RCL 18
16 ARCL 17	60 ST+ Z	104 E1	148 RCL 17
17 >"# DOM=?"	61 +	105 *	149 X=Y?
18 XEQ 10	62 ISG IND Y	106 STO \	150 "" EXT. "
19 ISG 17	63 ""	107 LASTX	151 X=Y?
20 GTO 00	64 X#Y?	108 ST+ Z	152 ARCL X
21 CLX	65 ISG IND X	109 +	153 CLAXON
22 STO 17	66 ""	110 X=Y?	154 AVIEW
23 "ME 1ST?"	67 FS?C 03	111 SF 00	155 PSE
24 AON	68 GTO 04	112 1.1	156 RCL 20
25 STOP	69 RCL 19	113 ST+ IND Y	157 10^X
26 AOFF	70 RCL 20	114 FC? 00	158 ST- IND 19
27 FC?C 23	71 *	115 ST+ IND Z	159 RCL 19
28 GTO A	72 LASTX	116 RCL \	160 10^X
29 6 E-3	73 +	117 10^X	161 X=Y?
30 STO [74 RCL 19	118 ST+ IND [162 SF 00
31*LBL 01	75 +	119 RCL [163 X#Y?
32 XEQ 16	76 RCL 17	120 10^X	164 ST- IND 20
33 ISG [77 -	121 FC?C 00	165 RCL 19
34 GTO 01	78 LASTX	122 ST+ IND \	166 RCL 20
35 RCL 19	79 RCL 18	123 RTN	167 E1
36 STO 17	80 *	124*LBL B	168 ST+ Z
37 RCL 20	81 LASTX	125 RCL 17	169 +
38 STO 18	82 +	126 STO [170 LASTX
39 GTO 06	83 X#Y?	127 RCL 18	171 1/X
40*LBL A	84 GTO 03	128 X=Y?	172 ST- IND Z
41 CF 27	85 "EXT.="	129 SF 00	173 FC?C 00
42 "YOU:"	86 PROMPT	130 XEQ 16	174 ST- IND Y
43 CF 22	87 STO 17	131 FS?C 00	175 DSE 08
44 PROMPT	88 STO 18	132 GTO 03	176 GTO A

177 "I WIN"	212 "NULL GAME"	247 +	282 RCL 17
178 GTO 09	213*LBL 09	248 7	283 RCL 19
179*LBL 02	214 BEEP	249 RCL IND Y	284 X#Y?
180 "YOU TAKE"	215 PROMPT	250 STO]	285 GTO 03
181 AVIEW	216 GTO 11	251 X<Y?	286 RCL 20
182 RCL 17	217*LBL 03	252 GTO 03	287 STO 17
183 RCL 18	218 "I TAKE"	253 FRC	288 RTN
184 X=Y?	219 XEQ 10	254 .2	289*LBL 03
185 SF 00	220 SIGN	255 X<=Y?	290 RCL 18
186 E-2	221 ST- 09	256 ST+]	291 X#Y?
187 ST+ IND Z	222 ST+ 08	257 GTO 04	292 GTO 03
188 FC?C 00	223 GTO B	258*LBL 03	293 RCL 20
189 ST+ IND Y	224*LBL 16	259 FRC	294 STO 18
190 RCL 09	225 .	260 .1	295 RTN
191 X=0?	226 STO 21	261 X=Y?	296*LBL 03
192 GTO 03	227 6 E-3	262 ST+]	297 RCL 20
193 SIGN	228 STO \	263 RCL IND \	298 X#Y?
194 ST- 09	229 RCL IND [264 FRC	299 GTO 03
195 ST+ 07	230 STO a	265 X#0?	300 RCL 19
196 GTO A	231*LBL 07	266 ST+]	301 STO 18
197*LBL 03	232 RCL a	267*LBL 04	302 RTN
198 "NO TURN"	233 INT	268 RCL 21	303*LBL 03
199 AVIEW	234 E1	269 RCL]	304 R^
200 SF 02	235 /	270 X<=Y?	305 X#Y?
201 GTO B	236 STO a	271 GTO 08	306 GTO 03
202*LBL 05	237 FRC	272 STO 21	307 RCL 19
203 RCL 09	238 X#0?	273 RCL [308 STO 17
204 X#0?	239 GTO 03	274 INT	309 RTN
205 GTO 03	240*LBL 08	275 STO 19	310*LBL 03
206 "I PASS"	241 ISG \	276 RCL \	311 "CHEATER"
207 AVIEW	242 GTO 07	277 INT	312 SF 27
208 PSE	243 RTN	278 STO 20	313 GTO 09
209 FC?C 02	244*LBL 03	279 SF 01	314 END
210 GTO A	245 RCL \	280 GTO 08	
211 PSE	246 E1	281*LBL 17	

Health Check

Whodunit – Swap Disks

No documentation is available. It uses the Card Reader (!)

Program listing:

01*LBL	46 PSE	95 "`X"	137 GTO 02
"CHECKUP"	47 "-CITADIN"	96 XEQ 15	138 TONE 9
02	48 FC? 06	97 3	139 41
"*LONGEVITE*"	49 "`E"	98 FC?C 05	140 X=Y?
03 AVIEW	50 XEQ 15	99 ST+ 05	141 SF 05
04 PSE	51 2	100 "AVEZ VOUS	142 RTN
05 CLRG	52 FC?C 05	..."	143*LBL 00
06 "AGE"	53 ST- 05	101 AVIEW	144 "NOMBRE DE
07 TONE 3	54 "-SEUL"	102 TONE 3	CIGAR"
08 PROMPT	55 FC? 06	103 PSE	145 >"ETTE"
09 ABS	56 "`E"	104 PSE	146 AVIEW
10 INT	57 XEQ 15	105 "FAIT DES	147 >"S PAR JOUR
11 STO 00	58 FS?C 05	ETUDES"	?"
12 "SEXE"	59 GTO 00	106 >"S"	148 TONE 3
13 XEQ 15	60 RCL 05	107 AVIEW	149 PROMPT
14 CF 05	61 4	108	150 ABS
15 CF 06	62 -	>"UPERIEURES"	151 E1
16 20	63 STO 01	109 XEQ 15	152 /
17 -	64*LBL 00	110 2	153 3
18 X=Y?	65 "-ACTI"	111 FC?C 05	154 Y^X
19 GTO 00	66 FC? 06	112 ST+ 05	155 SQRT
20 3	67 "`VE"	113 "UNE	156 STO 02
21 SF 06	68 FS? 06	PROFESSION "	157 "NOMBRE DE
22 ST- 05	69 "`F"	114 >"IN"	VERRE"
23*LBL 00	70 XEQ 15	115 AVIEW	158 >"S D'ALCOO"
24 8	71 2	116	159 AVIEW
25 "TAILLE"	72 FC?C 05	>"DEPENDANTE"	160 >"L PAR JOUR
26 TONE 3	73 ST+ 05	117 XEQ 15	?"
27 PROMPT	74 "-SPORTI"	118 3	161 TONE 3
28 ABS	75 FC? 06	119 FC?C 05	162 PROMPT
29 E	76 "`VE"	120 ST+ 05	163 ABS
30 -	77 FS? 06	121 "FAIT UN	164 2
31 110	78 "`F"	BILAN D"	165 /
32 *	79 XEQ 15	122 >"E SA"	166 LASTX
33 "POIDS"	80 3	123 AVIEW	167 -
34 TONE 3	81 FC?C 05	124 >"NTE	168 STO 08
35 PROMPT	82 STO 06	ANNUEL"	169 "NOMBRE
36 X<>Y	83 "-SATISFAIT"	125 XEQ 15	D'HEURES"
37 /	84 FC? 06	126 2	170 >" DE
38 E1	85 >"E"	127 FC?C 05	SOMME"
39 *	86 >" EN AMOUR"	128 STO 04	171 AVIEW
40 -	87 XEQ 15	129 GTO 00	172 >"IL PAR
41 STO 07	88 2	130*LBL 15	NUIT?"
42 "VOUS ETES	89 FC?C 05	131 "`?"	173 TONE 3
..."	90 ST+ 05	132 AVIEW	174 PROMPT
43 AVIEW	91>" -ANXIEU"	133 TONE H	175 ABS
44 TONE 3	92 FC? 06	134*LBL 02	176 2
45 PSE	93 >"SE"	135 GETKEY	177 /
	94 FS?C 06	136 X=0?	178 4

179 -	221 AVIEW	265 1.5	307 >"MOINS"
180 STO 03	222 PSE	266 RCL 08	308 AVIEW
181 "COMBIEN AVEZ-VO"	223 FIX 0	267 X<Y?	309 PSE
182 >"US D"	224 CF 29	268 GTO 00	310*LBL 00
183 AVIEW	225 "** "	269 "VOUS BUVIEZ MOI"	311 RCL 01
184 >"E GRANDS- PAR"	226 ARCL 09	270 >"NS"	312 X=0?
185 AVIEW	227 "` ANS**"	271 AVIEW	313 GTO 00
186 "` ENTS DECEDES"	228 BEEP	272 PSE	314 "VOUS ROMPIEZ VO"
187 AVIEW	229 AVIEW	273*LBL 00	315 AVIEW
188 >" APRES L'AGE"	230 PSE	274 RCL 06	316 >"TRE SOLITUDE"
189 AVIEW	231 PSE	275 X#0?	317 AVIEW
190 >" DE 80 ANS ?"	232 E	276 GTO 00	318 PSE
191 TONE 3	233 LASTX	277 "VOUS FAISIEZ DU"	319*LBL 00
192 PROMPT	234 "S"	278 >" SPORT"	320 CLA
193 70	235 X<=Y?	279 AVIEW	321*LBL 01
194 RCL 00	236 GTO 01	280 PSE	322 >"IL NE VOUS ARR"
195 .24	237 +	281*LBL 00	323 >"IVE AU"
196 *	238 "VOUS POURRIEZ A"	282 RCL 04	324 AVIEW
197 +	239 "` VOIR "	283 X#0?	325 >"CUN ACCIDENT"
198 +	240 AVIEW	284 GTO 00	326 AVIEW
199 RCL 05	241 "` UN BONUS DE "	285 "VOUS VOUS FAISI"	327 >" FACHEUX AVA"
200 +	242 AVIEW	286 "` E"	328 AVIEW
201 RCL 07	243 ADATE	287 AVIEW	329 "` NT L'AN "
202 ABS	244 "` ANS SI ..."	288 >"Z EXAMINER R"	330 DATE
203 RCL 02	245 AVIEW	289 AVIEW	331 E2
204 +	246 PSE	290	332 *
205 RCL 08	247 CHS	"` EGULIEREMENT"	333 FRC
206 ABS	248 RCL 07	291 AVIEW	334 E4
207 +	249 X>Y?	292 PSE	335 *
208 RCL 03	250 GTO 00	293*LBL 00	336 RCL 00
209 ABS	251 "VOUS PERDIEZ DU"	294 "VOUS DORMIEZ "	337 -
210 +	252 "` POIDS"	295 1.4	338 RCL 09
211 RCL 01	253 AVIEW	296 CHS	339 INT
212 -	254 PSE	297 RCL 03	340 +
213 RCL 06	255*LBL 00	298 X>Y?	341 ARCL X
214 -	256 5	299 GTO 00	342 AVIEW
215 RCL 04	257 RCL 02	300 >"PLUS"	343 PSE
216 -	258 X<Y?	301 AVIEW	344 FIX 2
217 -	259 GTO 00	302 PSE	345 CLOCK
218 STO 09	260 "VOUS CESSIEZ DE"	303*LBL 00	346 END
219 "ESPERANCE DE VI"	261 >" FUMER"	304 .9	
220 >"E"	262 AVIEW	305 X>Y?	
	263 PSE	306 GTO 00	
	264*LBL 00		