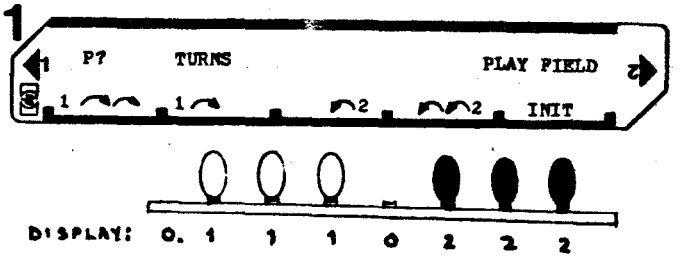
STEP	KEY ENTRY 1	KEY CODE
1	LBL E	21 15
	CL REG	16-53
	INT	16 34
	X = 0	16-43
5	GTO 0	22 00
	ABS	16 31
	4	04
	X≤Y	-41
	X>Y	16-34
10	GTO 0	22 00
	STO D	35 14
	STO B	35 12
	2	02
	x	-35
15	1	01
	+	-55
	ST I	35 46
	STO E	35 15
	2	02
20	LBL I	21 01
	STO(i)	35 45
	DSZ	16 25 46
	GTO 1	22 01
	RCL E	36 15
25	RCL D	36 14
	-	-45
	ST I	35 46
	0	00
	STO(i)	35 45
30	DSZ	16 25 46
	I	01
	LBL 2	21 02
	STO(i)	35 45
	DSZ	16 25 46
35	GTO 2	22 02
	RCL E	36 15
	ST I	35 46
	CLx	-51
	LBL 3	21 03
40	I	01
	0	00
	÷	-24
	RCL(i)	36 45
	+	-55
45	DSZ	16 25 46
	GTO 3	22 03
	1	01
	0	00
49	÷	-24
	RCL E	36 15
	ST I	35 46
	DSP(i)	-63 45
	R↓	-31
	F0 ?	16 23 00
55	-x-	-14
	RTN	24
	LBL B	21 12
	SF 1	16 21 01
	LBL C	21 13
60	1	01
	GTO f d	22 16 14
	LBL A	21 11
	SF 1	16 21 01
	LBL D	21 14
65	2	02
	LBL f d	21 16 14
	STO D	35 14
	1	01
	RCL A	36 11
70	+	-55
	STO A	35 11
	RCL E	36 15
	ST I	35 46
	LBL 4	21 04
75	RCL(i)	36 45
	X = 0	16-43
	GTO 5	22 05
	DSZ	16 25 46
	GTO 4	22 04
80	LBL 5	21 05
	RCL I	36 46
	STO C	35 13
	RCL D	36 14
	F1 ?	16 23 01
85	CHS	-22
	+	-55
	ST I	35 46
	RCL(i)	36 45
	2	02
90	X≠Y	-41
	F1 ?	16 23 01
	1	01
	X≠Y	16-32
	GTO 0	22 00
95	0	00
	STO(i)	35 45
	CLx	-51
	RCL C	36 13
	ST I	35 46
100	R↓	-31

STEP	KEY ENTRY 2	KEY CODE
1	STO(i)	35 45
	CF 1	16 22 01
	RCL B	36 12
	ST I	35 46
5	LBL 6	21 06
	2	02
	RCL(i)	36 45
	X≠Y	16-32
	GTO f e	22 16 15
10	DSZ	16 25 46
	GTO 6	22 06
	RCL B	36 12
	1	01
	+	-55
15	ST I	35 46
	RCL(i)	36 45
	X≠0	16-42
	GTO f e	22 16 15
	SF 2	16 21 02
20	LBL f e	21 16 15
	RCL E	36 15
	ST I	35 46
	CLx	-51
	GSB 3	23 03
25	F2 ?	16 23 02
	F2 ?	16 23 02
	R/S	51
	PAUSE	16 51
	RCL A	36 11
30	DSP 0	-63 00
	F0 ?	16 23 00
	-x-	-14
	F0 ?	16 23 00
	SPACE	16-11
35	RTN	24
	LBL f a	21 16 11
	F0 ?	16 23 00
	GTO 7	22 07
	SF 0	16 21 00
40	1	01
	R/S	51
	LBL 7	21 07
	CF 0	16 22 00
	0	00
45	R/S	51
	LBL f b	21 16 12
	RCL A	36 11
	RTN	24
49	LBL 0	21 00
50	CF 1	16 22 01
	CLx	-51
	÷	-24
	R/S	51
55		
60		
65		
70		
75		
80		
85		
90		
95		
100		

1. PEGS BY: Craig A. Pearce 311

2. BY:



67
97

DISPLAY: 0. 1 1 1 0 2 2 2

Program Description, Equations, Variables The HP-97 (or 67) will set up a play field consisting of double the number of 'pegs' the user wishes to play in this game. The play field consists of 2 different colored pegs, the white represented by the no. '1', the black by the no. '2'. When a player selects a number between 1 and 4 inclusive, the game is initialized with 'n' number of both 'white' and 'black' pegs, the black (#2) pegs on the right of a zero, the white (#1) pegs on the left side of the zero. This zero between them represents an open space on play field.

The object of the game is for the player to move these pegs, 1 at a time, into the open space, until the positions of all the pegs are reversed with the #2 (black) pegs on the left, and the #1 (white) pegs on the right. The trick is, the #1 pegs can ONLY move to the right, by ONLY 1 or 2 places. Likewise, the #2 pegs can ONLY move left, one or two places. The position that any peg can move to is the open space (the '0'). Thus, with one or two pegs of each 'color' the game is not all that hard, but just try it with 3 or 4 pegs of each 'color'.

The 97/67 will count the number of moves made, and indicate a win by pausing the display, then stopping with the number of moves showing on the display. At anytime, the user can call up the number of moves made so far, recall the play field, or reinitialize. If an illegal move is attempted, the 67/97 will say so, and the user can clear the error, and recall the original display of pegs, but with a loss of one turn.

This game is playable on either the HP-67 or HP-97, with the option of switching the printer on or off, for 97 play.

Operating Limits and Warnings The user can only enter the numbers between 1 and 4, inclusive. Entering a number outside this range will result in an "Error" message. Inputing numbers containing fractions will be corrected to the correct form. The same is true for negative inputs.

The play field is to the RIGHT of the decimal point. Ignore the zero to the left of the point.

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Load program (sides 1 and 2);			
2	Select the no. of pegs in the game and input this number (1 thru 4);	n	E	0.play-field
3	To move a '1' peg one space right;		B	
	-or-			
3	To move a '1' peg two spaces right;		A	
	-or-			
3	To move a '2' peg one space left;		C	
	-or-			
4	To move a '2' peg two spaces left;		D	
4	Optional-Recall turns at any time;		f 1	
5	Optional-To alternately switch the printer on and off;		f 2	1---On 0---Off
6	If an error is made, to clear error & return present state of play field to display;		CLx	
7	Optional-At any time, to recall the current state of the play field to the display;		f 3	
8	Repeat step 3 until game is won or lost. Final, winning field will be displayed, followed by no. of turns, if game is won.		f 4	
9	For a new game, go to step 2.			