

PROGRAM DESCRIPTION I

Program Title REVERSIContributor's Name Valentin Albillo

Address _____

City _____ State/Country _____ Zip Code _____

Program Description, Equations, Variables This program allows the user to play a game of Reversi against an HP-41C. 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.

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 * * * * -
6  - - O O O * * O
7  - - - - O * * -
8  - - - - - O * O

```

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 \emptyset (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.

Reference(s) 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!!!)

This program has been verified only with respect to the numerical example given in *Program Description II*. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

Reversi is played on an 8x8 board. There are two standard openings (see illustrations):

- diagonal opening (left)
- parallel opening (right)

	1	2	3	4	5	6	7	8
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	⊗	○	-	-	-
5	-	-	-	○	⊗	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

One of the players plays the white pieces (represented by the O), the other the black ones (represented by the checkerboard character).

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.

	1	2	3	4	5	6	7	8
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	○	○	-	-	-
5	-	-	-	⊗	⊗	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

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.

	1	2	3	4	5	6	7	8
1	○	⊗	⊗	-	⊗	⊗	⊗	○
2	○	○	○	○	○	○	○	○
3	○	○	○	⊗	○	○	○	○
4	⊗	⊗	○	⊗	○	○	⊗	○
5	⊗	⊗	○	○	⊗	○	⊗	○
6	⊗	○	-	○	○	○	○	⊗
7	⊗	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○

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).

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 8x8 board, including edges (thus being actually a 10x10 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 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.

```

SF 02
XEQ "REVERSI"
DIAG ?
RUN
1 2 3 4 5 6 7 8
1 - - - - -
2 - - - - -
3 - - - - -
4 - - - O - - -
5 - - - O - - -
6 - - - - -
7 - - - - -
8 - - - - -

```

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 O's).

(if you are not using a printer, you need an actual 8x8 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).

```

HP 1ST ?
N
MOVE ?
64
RUN
YOU PLAY 64
FLIP 1 PCES
I PLAY 63
FLIP 1 PCES

```

- 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 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.

```

MOVE ?
CF 02
76 RUN
YOU PLAY 76
FLIP 1 PCES

```

```

1 2 3 4 5 6 7 8
1 - - - - -
2 - - - - -
3 - - - - -
4 - - - O - - -
5 - - O O - - -
6 - - O O - - -
7 - - - - -
8 - - - - -

```

```

I PLAY 66
FLIP 1 PCES

```

```

1 2 3 4 5 6 7 8
1 - - - - -
2 - - - - -
3 - - - - -
4 - - - O - - -
5 - - O O - - -
6 - - O O O O - -
7 - - - - -
8 - - - - -

```

```

 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 X O O X 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

```

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)

```

GAME IS OVER
HP: 49, YOU: 15
I WON

```

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
												22	28

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.)

USER INSTRUCTIONS

SIZE: (HP-41C) 117

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
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?
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
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)			
9	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?
9	You have no legal move: enter: and HP proceeds to compute its move.	Ø	R/S	
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
	(If you play without a printer, remember to actualize the board after HP moves: put a white piece where indicated, then flip your captured pieces.)			MOVE?

USER INSTRUCTIONS

<u>STEP</u>	<u>INSTRUCTIONS</u>	<u>INPUT</u> <u>FUNCTION</u>	<u>DISPLAY</u>
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		GAME IS OVER HP: nn, YOU: mm I WON or YOU WON

Of course, the player with the most pieces at the end of the game wins the game. 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.

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, whenever 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!

67 97 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
01	LBL	"REV		36	STO	72	
	ERSI"			37	9		
02	CLRG		Initialization	38	STO	15	
03	FIX	0		39	CHS		
04	CF	29		40	STO	14	
05	CF	01		41	+		
06	CF	12		42	STO	11	
07	.8188111		Store move	43	CHS		
883			constants	44	STO	10	
08	STO	16		45	11		
09	.8661683			46	STO	13	
138				47	CHS		
10	STO	17		48	STO	12	
11	.1316636			49	61		
633				50	STO	07	
12	STO	18		51	"DIAG ?"		Query for
13	.3684855			52	CF	23	parallel or
158				53	AON		diagonal
14	STO	19		54	PROMPT		opening
15	.4148141			55	RCL	09	
564				56	RCL	08	
16	STO	20		57	FS?C	23	
17	.6553564			58	X<>Y		
346				59	STO	61	
18	STO	21		60	X<>Y		
19	.3435747			61	STO	71	
552				62	XEQ	06	Print board
20	STO	22		63	"HP 1ST		"Who moves
21	.5742472				?"		first?" test
425				64	PROMPT		
22	STO	23		65	AOFF		
23	.7376626			66	FS?C	23	
732				67	GTO	00	
24	STO	24		68	"I MOVE"		HP first
25	.3723268			69	AVIEW		
287				70	SF	29	--First move
26	STO	25		71	LBL	14	of game
27	.7178212			72	"I"		HP's move
812				73	CF	03	
28	STO	26		74	LBL	08	
29	.1772772			75	16.027		
227				76	FS?C	29	First move of
30	STO	27		77	21		game
31	SIGN		1	78	STO	05	
32	STO	62	Initialize	79	LBL	11	
33	STO	09	test constants	80	RCL	IND	Recall constant
34	CHS		-1	05			Constant
35	STO	08		81	X=0?		exhausted?

PROGRAM LISTING

67 97 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
82	GTO	05	Get new constant	127	SF	04	Occupied square
83	SF	05		128	17		flag
84	♦LBL	13		129	+		
85	RCL	10	10	130	STO	00	
86	X↑2			131	RCL	IND	
87	*			00			
88	STO	06		132	X≠0?		Square occupied?
89	INT			133	RTN		Return
90	XEQ	12	Test position	134	CF	05	Empty square
91	FC?C	04	Invalid move?	135	STO	01	Initialize empty
92	GTO	00		136	8.015		square test
93	RCL	06		137	STO	02	
94	FRC			138	RCL	09	
95	X≠0?		If constant not	139	FC?	03	
96	GTO	13	exhausted, re-	140	CHS		
97	FS?	05	cycle	141	STO	04	
98	STO	IND	All squares test-	142	♦LBL	01	
05			ed full? Null	143	RCL	00	
99	♦LBL	05	constant	144	RCL	IND	Is there a flip-
100	ISG	05	Next constant	02			pable neighbor
101	GTO	11		145	+		piece?
102	"NO MOVE		All tested moves	146	STO	03	
"			failed	147	RCL	IND	
103	AVIEW			X			
104	TONE	9		148	RCL	04	Neighbor not
105	PSE			149	X=Y?		flippable
106	♦LBL	00	Your move	150	GTO	12	
107	FS?C	01		151	♦LBL	03	Test the rest of
108	GTO	14		152	LASTX		the pieces in
109	"MOVE ?"			153	ST+	03	the row
110	PROMPT			154	RCL	IND	
111	X=0?		No move?	03			
112	GTO	14	Go to HP's move	155	RCL	04	
113	SF	03		156	X=Y?		
114	"YOU"		Your move	157	GTO	03	
115	X<0?			158	CHS		Space?
116	SF	01		159	X=Y?		Next direction
117	X<0?		HP move for you	160	GTO	12	
118	GTO	00		161	STO	IND	
119	XEQ	12	Test position	00			Flip pieces
120	FC?C	04	Legal move?	162	♦LBL	04	
121	GTO	14		163	LASTX		
122	"ILLEGAL			164	ST-	03	
"				165	RCL	00	
123	AVIEW			166	RCL	03	
124	TONE	9		167	X=Y?		
125	GTO	00		168	GTO	12	
126	♦LBL	12		169	RCL	08	

67 97 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
170	ST*	IND		216	RCL	13	Print horizontal
Y				217	SKPCOL		labels 1-8
171	ST-	01		218	X<>Y		
172	GTO	04		219	ACCHR		
173	♦LBL	12		220	ISG	X	
174	ISG	02	Increment flip	221	GTO	02	
175	GTO	01	test counter	222	PRBUF		
176	RCL	01		223	28.035		
177	X=0?		No valid flips?	224	STO	05	
178	RTN			225	♦LBL	09	Print playing
179	CF	04	End test	226	RCL	04	board row by
180	"F	PLAY		227	ACCHR		row
"				228	RCL	15	
181	RCL	00	Display move	229	SKPCOL		
182	17			230	SF	12	
183	-			231	♦LBL	10	Accumulate row
184	ARCL	X		232	RCL	IND	characters for
185	AVIEW			05			printing
186	FC?	01		233	RCL	09	
187	FC?	03		234	+		
188	BEEP			235	RCL	IND	
189	PSE			X			
190	"FLIP	"		236	ACCHR		
191	ARCL	01		237	RCL	03	
192	"F	PCES"		238	SKPCOL		
193	AVIEW			239	ISG	05	
194	PSE			240	GTO	10	
195	FC?	02	Print board?	241	PRBUF		
196	GTO	06		242	ST+	05	
197	FS?	03	HP's move?	243	CF	12	
198	GTO	12		244	ISG	04	
199	♦LBL	06		245	GTO	09	
200	FC?	55	If no printer,	246	ADV		
201	GTO	12	skip printing	247	FS?	03	
202	ADV		routine	248	GTO	12	
203	31			249	ADV		
204	STO	00		250	ADV		
205	45			251	♦LBL	12	Move counter
206	STO	01	Initialization	252	DSE	07	
207	79		for board	253	RTN		
208	STO	02	printing	254	FC?	02	
209	2.01		routine	255	GTO	12	
210	STO	03		256	FS?C	03	
211	8			257	XEQ	06	
212	SKPCOL			258	♦LBL	12	
213	49.056			259	32		
214	STO	04		260	"GAME IS		
215	♦LBL	02		OVER"			

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

PROGRAM LISTING

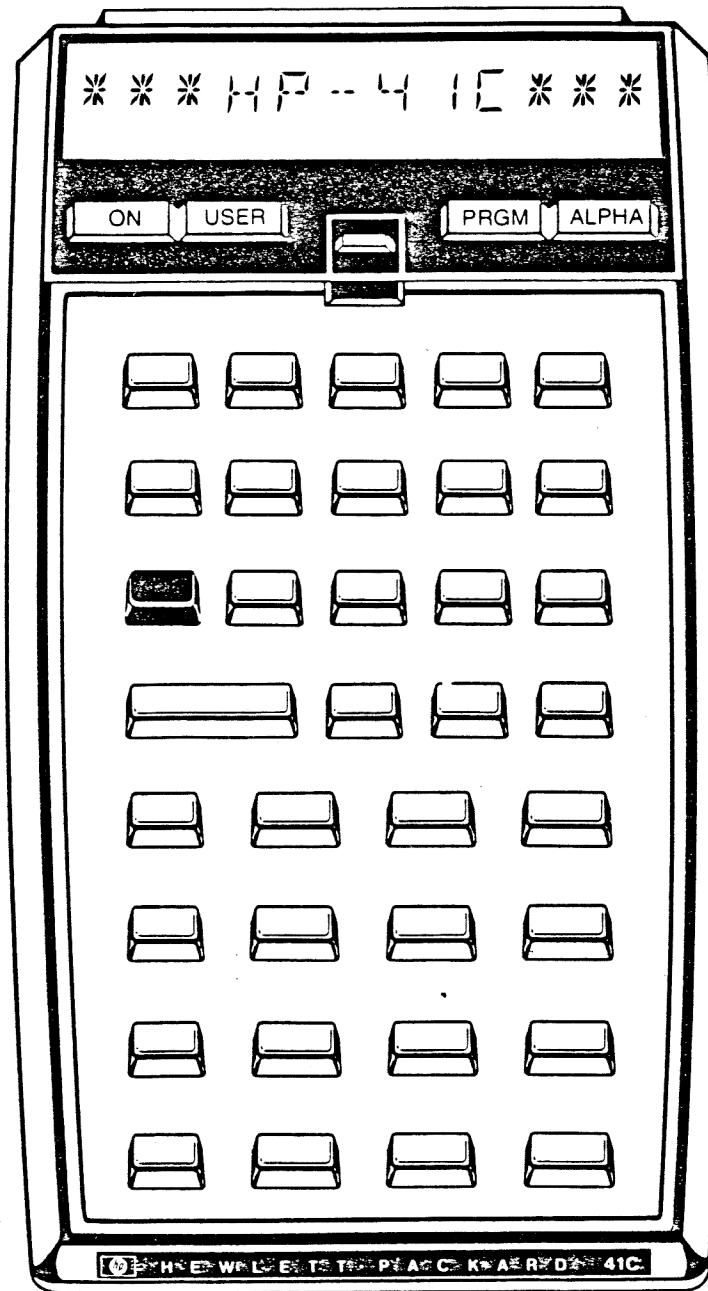
67 97 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
261	28.105						
262	AVIEW						
263	0						
264	◆LBL 07		Total scores				
265	RCL IND						
	Y						
266	+						
267	ISG Y						
268	GTO 07						
269	2						
270	/						
271	X<>Y						
272	RDN						
273	ST- Z						
274	+						
275	ADV						
276	"HP: "		Display scores				
277	ARCL X						
278	"F, YOU:						
	"						
279	ARCL Y						
280	AVIEW						
281	BEEP						
282	ADV						
283	PSE						
284	X=Y?						
285	STOP						
286	"I"						
287	X<Y?						
288	"YOU"						
289	"F WON"		Who won?				
290	PROMPT						
291	.END.						

REGISTERS, STATUS, FLAGS, ASSIGNMENTS

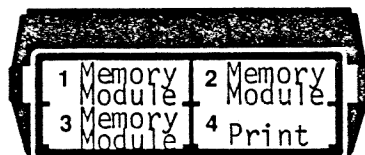
DATA REGISTERS		STATUS																																													
01 } to } Scratch		SIZE 117	TOT. REG. 213																																												
07 } 08 } to } Directions array		ENG _____	FIX _____ SCI _____																																												
15 } 16 } to } Constants array 17 } 27 } 116 } Board		DEG _____	RAD _____ GRAD _____																																												
(Constants array and Board overlap)		FLAGS <table border="1"> <thead> <tr> <th>#</th> <th>INIT S/C</th> <th>SET INDICATES</th> <th>CLEAR INDICATES</th> </tr> </thead> <tbody> <tr> <td>01</td> <td></td> <td>HP playing your pieces</td> <td></td> </tr> <tr> <td>02</td> <td></td> <td>Only 1 board</td> <td>Both boards</td> </tr> <tr> <td>03</td> <td></td> <td>Move tested legal</td> <td></td> </tr> <tr> <td>04</td> <td></td> <td>Move not yet legal</td> <td></td> </tr> <tr> <td>05</td> <td></td> <td>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.</td> <td></td> </tr> <tr> <td>12</td> <td></td> <td>Double wide print</td> <td>Single wide print</td> </tr> <tr> <td>23</td> <td></td> <td>Alpha input</td> <td>No alpha input</td> </tr> <tr> <td>29</td> <td></td> <td>Decimal point</td> <td>Suppress Decimal</td> </tr> <tr> <td></td> <td></td> <td colspan="2">Flag 29 is also set to indicate the first move of the game.</td> </tr> <tr> <td>55</td> <td></td> <td>Printer exists</td> <td>No printer</td> </tr> </tbody> </table>		#	INIT S/C	SET INDICATES	CLEAR INDICATES	01		HP playing your pieces		02		Only 1 board	Both boards	03		Move tested legal		04		Move not yet legal		05		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.		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
#	INIT S/C	SET INDICATES	CLEAR INDICATES																																												
01		HP playing your pieces																																													
02		Only 1 board	Both boards																																												
03		Move tested legal																																													
04		Move not yet legal																																													
05		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.																																													
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																																												

KEYBOARD CARD LABELING



KEYBOARD

SYSTEM
CONFIGURATION



(Printer is optional)

CARD

