Notes on the back story of this letter:

At this time I was still very enthusiastic about anything and everything having to do with *HP* calcs programming (mostly with the new *HP-41C* and its awesome synthetic instructions,) so I was creating and submitting materials to **Richard Nelson** for publication in the *PPC Calculator Journal* in rapid-fire style, e.g., this particular *9-page* letter was the *third* one I sent him within a month.

I began this letter praising the recent *June* double issue while at the same time complaining about the lack of *HP-67* programs (which members had asked for), despite *Mr. Nelson* having previously stated that he couldn't publish *HP-67* programs because none were submitted, which just wasn't true, as I told and showed him in my usual direct style (I was *very* young at the time ...)

The letter also included two significant contributions, namely (1) a synthetic routine for the *PPC ROM*, *Clear* (*all*) *Assignments*, which had been requested by several members because of its obvious usefulness (and it wasn't included, of course), and (2) the very first version for any machine of my *Othello* program, which was a long, optimized, polished program which would play quite fast a non-trivial game against the user, optionally printing a nice labeled board if a printer was available.

I was *very* proud of my *Othello* program, there was nothing remotely like it and so I thought that it would cause a sensation among the members of *PPC* and even could be used by *HP* to promote the *HP-41C*, using it as a demo program at events, which was the case locally: my friends and acquaintances loved it and *HP Spain* did use it for the purpose. In time, it was featured in the *Wall Street Journal Magazine* and *HP Corvallis* asked me personally to submit it to the *Users' Program Library USA* (which I promptly did, receiving a nice - unasked for - reward for it; see the letters.)

However, despite the fact that I sent *Mr*. *Nelson* the documented program ready for him to cut & paste + good photocopies to prevent printouts fading + the complete printout of a full sample game + magnetic cards (+ the mailing cost, all of which were very hard for my meager finances), he *never* sent me a note or comment, and he *never* published it in *PPC CJ*, nor did he bother to tell me *why*.

I was (again !) demolished by this, and after waiting almost a year (!) for its publication, which never happened, I began to consider other places where my materials would get a warmer reception and I'd get some feedback, which was the case with **John McGechie** and the members of the *Melbourne Chapter of PPC*, the utter polar opposites of *Mr. Nelson* and his *PPC CJ*, thus marking the beginning of the end of my attachment to *PPC* proper.

Valentin Albillo, 21-12-2021

Richard Nelson Editor, PPC Journal 2541 W. Camden Place U. S. A. Valentin Albillo (4747) Padre Rubio, 61 - 2º C Madrid 29 SPAIN

Dear Richard :

I've just received the June ISSUE (capital letters, no less !) . I was thinking something wrong was happening with the post, as the issue didn't arrived. But finally it did, and it was worth the wait: I can express my satisfaction in 3 languages at least: Hats off ! (or) Chapeau ! (or); Hay que descubrirse!. Only one thing wrong with the June issue : where is -

the 67/97 material ? There is not a single program for the 67 (apart from some routines, and adaptable material), despite the 64 pages of the issue. This time you can't say you haven't got programs from the members. In about 2 months, I myself have con tributed 3 programs for the 67/97, they were:

- ELLIPTIC LOWPASS FILTER DESIGN

- A CHESS PROGRAM
- FOURIER SERIES HARMONIC ANALYSIS

If I (who do not have a 67, but a 41c) have sent 3 programs, other members should have sent dozens of programs ! Why are them not published ? The wand may be very interesting to us 41c owners, but think about the 67,97,34c owners: they have no use at all for the pages full of bar codes. Don't get angry, it's only an opinion.

Well, let's change of topic. Here included are 2 contributions:

- a) <u>1 routine for the Custom ROM</u>: it is CA, Clear Assignments. It seems to be very useful to have a routine to automatically clear all asignments at once, without having to clear them all manually, one at a time. Several members requested this. This routine does that, clears from 0 to 70 assignments in 9 seconds, uses no register, no flag, and is only 112 bytes long. Magnetic card included.No input req.
- b) <u>OTHELLO</u>, a computer game for the 41c : I'm sure you must have seen, (or even played) some computer programs to play OTHELLO at commercial computer shows, etc. It seems to be the most popular computer game, apart from chess. Even the HP-85 included a program which plays OTHELLO in one of its packs.

The program here included plays OTHELLO against the user. It includes full graphics capabilities (will print the board), is autonomous (no card reader or data cards required), it is printer compatible (runs with or without printer), includes all options: diagonal or parallel openings, checks user moves for legality, can play for the user, can play against itself, and includes all handling of the board, the winner, etc.

The level of play is astonishingly good for a calculator: defeats easily beginners, even experienced people. Execution is very fast: no more than 25 minutes for a whole game (no printer). Calculator remembers already played squares, so execution gets faster as the game progresses.

I have used it very much at shows and demos . All people who saw the 41c playing OTHELLO was amazed. Most of them were defeated by the program, and everybody wanted to have a game against the 41c. Maybe you could use it too for demo purposes, or to shut up all those Apple, etc, owners a little.

A paper tape with a game recorded onto it is included, for you to see how neatly operates the printer in conjuction with the 41c for games. The program itself is included recorded onto 3 magnetic cards.

I hope the membership enjoys (and perhaps, improves) the program. It is a definite answer to all those people wanting not mastermind-type games.

That's all. I include both the originals and a good photocopies, to overcome fading of printer tapes. Use the better ones.

Till the next letter . Sincerely

P.S.: the OTHELLO program does not use synthetic functions at all, so it can be submitted to the HP Library, if anyone wishes, or used at HP shows.

- this is an input for the ROM PROGRESS column : it is a routine to be considered for the PPC Custom ROM, a Clear Assign ments routine

The 41C has many clearing options: CLD, CLX, CLST, CLA, CLRG, CL Σ , and CLP, but it lacks two very useful options: one is the "clear assignments options", and the other the "clear all programs" option. The former is specially desired, as it is very keystroke busy to clear the assignments one at a time, if MEMO-RY LOST can't be tolerated. Bill Barnet suggests loading a status card with a single assignment recorded on it. This works, but a card reader is required. Some other members point out the need for a "clear assignments" software option, as in V7N5P13, so I offer the following routine to be considered for its in - clusion in the PPC Custom ROM . It is called CA (clear assignments at once, requiring no input at all from the user.

01*LBL "CA"	32 35
02 CLA	33 X>Y?
03 RCL c	34 X<>Y
04 STO E	35 0
05 ASHF	36 STO e
06 XEQ 01	37 STO T
07 RCL E	38 "*ix "
08 INT	39 RCL E
09 1	40*LBL 00
10 *	41 X<> c
11 16	42 X<>Y
12 *	43 STO IND Z
13 +	44 X<>Y
14 CLA	45 X<> c
15 X<>Y	46 DSE Z
16 X<> E	47 GTO 00
17 "F12345"	48*LBL 02
18 X<> N	49 CLST
19 "F6"	50 CLA
20 X<> N	51 RTN
21 CLA	52*LBL 01
22 STO E	53 "F****"
23 X<>Y	54 RCL E
24 XEQ 01	55 FRC
25 256	56 10
26 *	57 *
27 +	58 .END.
27 + 28 192 29 X=Y? 30 GTO 02 31 -	CAT 1 LEL'CA ENT 112 EYTES

-This routine is 58 lines, 112 bytes, so it exactly fits onto a single side of a card. It uses no register, no flag. It clears the stack and ALPHA registers upon execution. It will clear from 0 to 70 assignments within 9 seconds. Once executed, no assignments will be active. PACK is not neccessary, the registers used to store the assignments are automatically allocated as program memory, once the execution is completed. If there were any function (not program) assigned (at least 1), all asignments, whe ther functions or programs, are cleared. But if only programs are asigned, they are not removed by CA. This is done intentionally: you clear the assignments to gain program or data memory; as the assignments of user programs do not take out memory, it is unneccessary to delete them, they are deleted as soon as you clear the whole program.

HOW TO USE : simply, XEQ "CA" . All assignments are removed. Note : line 38 is F501690BF0BF ; line 53 is F67F0000000001

- <u>WARNINGS</u>: do not interrupt program execution, as this causes MEMORY LOST, as does trying to single step thru the program.
- lines 41 & 45 are needed to ensure that Rc is restored before the GTO 00 at line 47 is attempted. Otherwise, the GTO 00 will search the LBL 00 passing the .END., on its first execution, causing real trouble.
- program is totally safe. There is no danger of altering or deleting anything but the assignments registers. Even if program memory is absolutely full, and there are none assig nments made, or a single assignment, or any number of them, they will be deleted without altering the .END.

EXAMPLE : assume a just MASTER CLEARED machine.

load CA ; do not perform GTO ...; -assign : ASN SIN 11 (Σ +) ASN COS 12(1/x) ASN TAN 13(SQRT)

press RIN, switch to PRGM mode, to see OO REG 29, to RUN m. if we delete the assignments, this will give us back 2 registers, so we must see OO REG 31. Let's see: (RUN mode)

XEQ "CA" → 0.0000

switch to PRGM mode, to see OO REG 31, so it works. Now, keys 11,12,13 have recovered their original functions.

VALENTIN ALBILLO (4747)

Most members of PPC do like computer games. Some of them have joined PPC hoping to get some challenging ones to implement on their calculators. Even someone asked for a chess playing program for the 41c.

Here presented is a 41c program that plays OTHELLO . It is not chess, but it's closer to chess than to mastermind type programs. The present program includes all the features required: it plays quite well, and will easily defeat a beginner; it should provide a challenging level of play for everyone, however. It includes full graphics capabilities : if

YOU PLAY 57 FLIP 2 PCES								
	1	2	3	4	5	6	7	8
1				~****			-	••••
2			-					
3	ŗ,	Û	Û	-	-	•		
4	•**=	-	O	Ο	Ū			
5			0	8	\gtrsim	\gtrsim	$\stackrel{\scriptstyle \times}{\scriptstyle \times}$	
6		-	0	O	C	\otimes	\otimes	Ū
7			- and		0	\sim	3	
8			-			0	×	0

a printer is present, it will print the board. It is autonomous: no data cards required, no card reader needed. It is very fast: the 41c performs 30 moves in about 25 minutes. The running speed increases as the game progresses. It is printer compa tible: runs with or without printer. It offers both standard diagonal or parallel openings, you select who makes the first move, too . 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 for you against itself. Even can play a game

against itself. No synthetic functions used.

OTHELLO (also known as reversi) is a game very often found in computers. The HP-85 includes a program to play OFHELLO in one of its packs. Maybe some of you have already played a game against some computer (HP-85, Apple, TRS-80). Here is a brief outline of the rules:

OTHELLO is played in an 8x8 board. There are two standard openings (see illustrations); - diagonal opening

	1	2-	3	4	5	6	7	8
1								
2								
3	• ~~		-			•***	801-847	
4		-	-	0	O			
5			-	X		•	-	
6	-	•				•••••		
7						-		
8								

PARALLEL OPENING

800000000

- parallel opening

One of the players plays the white pieces (represented by the D), and the other the black pieces (represented by the).

To make a move, the player places his piece in an empty location (represented by a dash -), taking into account that: - it should 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

colour. This is , any number of pieces enclosed between the played piece and another piece of the same colour, are flipped: they become of the capturer's colour. No empty squares can be enclosed, only full rows of enemy's pieces can be flipped. The row can be in any direction: diagonal, horizontal, or vertical. If more than one row is enclosed at the same time, all are flipped.

Some examples should make it clear. 8 -----For instance, look at the diagonal opening: if black plays to 64 (6 vert, 4 horiz), then the white piece at 54 is between the 2 black

DIRGONAL OPENING

12345678

pieces at 44 and 64, so it is flipped: the white piece at 54 becomes black. Now look at this illustration: if white -12345678 plays at 14, the black pieces at 12, 13 10※※-※※※0 are enclosed between the just played white 200000000 piece at 14 and the piece at 11, so they 300000000 are flipped. Simultaneously, the black -4 ※ ※ O ※ O 0 ※ O pieces at 15,16,17 are between the piece $5 \otimes \otimes O \otimes O \otimes O \otimes O$ just played at 14, and the piece at 18, 6 20-0000 so they are flipped too. 7 \$ 0 0 0 0 0 0 0

2 _____ 5 - - - 0 🚿 - - -6 -----7 -----

In the same board position, if black plays at 05, it will flip the white pieces at 62,53,43,33,23,64,65,66,67, because there is another black piece at the end of each row of white pieces, and there are no empty squares in the rows.

PROGRAM CHARACTERISTICS : the OTHELLO program is exactly 672 by-

tes long, so it fits exactly onto 3 magnetic cards. Key it in exactly as listed, as a single extra byte makes neccessary a 4th card. The program is optimized for running speed: each location on the board is stored onto a single storage register, for máximum speed, so a minimum SIZE 117 is needed. 3 RAMs are required, thus, to run the program, lea ving a port to plug in the card reader or the printer.

Registers 00 thru 07 are used as scratch. Registers 08-15 contain the directions array. Registers 16-27 are used to store the constants array. Registers 17 thru 116 store the 8x8 board, including the edges, thus being a ||0x10| board.(The board array and the constants array may overlap without trouble, saving 11 registers). The white pieces (HP's) are stored as +1, black ones as -1, and the empty squares and the edges, as 0 (the edges may be any number, except ± 1 .). Program uses flags 1,2,3,4 . If flag 3 is set, your move is being tested for legality, or HP is playing your pieces against itself. If flag 4 is set, the present move is not proved to be legal yet. If flag 1 is set, HP is playing your own pieces for you. If flag 2 is set, and the printer is present, the board will not be printed after your moves (except if you make the last move). All flags are controlled by the program except flag 02, which is user-dependant.

INSTRUCTIONS : load the program. You play black, HP plays white. 1) if you want to use the printer, plug it in. Otherwise, remove it now. Set the mode switch at NORM position.

- 2) if you have the printer, and want to suppress board printing after your moves, SF 02. The board will now be printed only after HP's moves, saving time and paper. This can be done at any moment during the game. To print both boards, CF 02.
- 3) Make sure you have a minimum SIZE 117. Then, XEQ "OTHELLO" \rightarrow \rightarrow DIAG ? (prompts for the desired opening)
- 4)-if you want to play the diagonal opening, simply : R/S -if you want the parallel opening, press N R/S

(everything that appears in the display is printed as well, for you to keep a record of the game. The board is printed now, reflecting the opening you selected)

> HP 1ST ? (prompts for the first player)

- 5)-if you want HP to make the first move, simply R/S → I MOVE -if you want to make the first move, press N R/S → MOVE ?

where nn are the vert/horiz coordinates of the square where it moves to, and p is the number of your pieces flipped. If a printer is present, the resulting board position is printed now, and your move is requested: > MOVE ? (If no printer, actualize the board yourself, putting a white piece where indicated, then flipping the pertinent pieces) -if no move is possible for HP (all moves are illegal), it will display NO MOVE, (beep), then asks for your move.

- 7) if you move (you have been prompted by MOVE ?), there are several possibilities:
 - you have no legal move: press β R/S , and HP will make it s next move.
 - you want the machine to play for you: press -1 R/S (or any other negative number), and HP will play your pieces against it s own. Once found, your movement will be dis played (and beeped, recalling that HP perform the task),

and HP will proceed to make its move. This is very useful if you don't know what to play: let the machi ne select your move for you, hoping it will be good. Or, if you are unsure about you having a legal move or not: let the machine play your pieces, and if the re is a legal move for you, it will be found. Otherwise, NO MOVE is displayed. Another use is to have the machine play against itself the whole game: input -1 always.

- you want to make a move, say xy (vert/horiz): xy R/S

(your move is checked for legality. If it was not legal, ILLEGAL (beep) is displayed, and you are prompted for another move) If your move is legal, then:

> YOU PLAY XY → FLIP p PCES

is displayed. If printer is present and flag 02 is clear, the board is printed. Otherwise, HP proceeds directly to compute its move. If no printer, actualize the board manually.

- Warning : your move must be of the form xy, with x ranging from 1 to 8, both included, and the same for y. \emptyset and negative numbers are allowed too.
- 8) Once the last player makes the last move, the board is printed, and : \rightarrow GAME IS OVER (the counting takes place)

HP: xx, YOU: yy > HP or YOU WON

where xx is the number of white pieces at the end of the game, yy is the number of black pieces. of course, THE PLAYER WITH MOST PIECES AT THE END OF THE GAME WINS THE GAME. This is, if HP has 24 pieces on the board, and you have 40, you won. If both have 32, it is a tie, and the HP or YOU WON message is not displayed. The game generally ends when the board is full of pieces. It can also end if no player can make a legal move, but this is very unlikely.

SAMPLE GAME

SAMPLE GAME		MOVE ?
SF 02 XEQ "OTHELLO" DIAG ? RUN	left : typical beginning of a game: diagonal opening	CF 02 76 RUH YOU PLAY 76 FLIP 1 PCES
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	<u>right</u> : flag 02 is cleared, so the board is printed after every move	1 2 3 4 5 6 7 8 1
HP 1ST ? N RUN	I play first I play 64	I PLAY 66 FLIP 1 PCES
64RUNYOU PLAY 64FLIP 1 PCES1 PLAY 63FLIP 1 PCES	flag 02 is set, so the board is not printed after my moves. HP plays 63. and the board	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & - & - & - & - & - & - \\ 2 & - & - & - & - & - & - & - \\ 3 & - & - & - & - & - & - & - & - \end{array}$	is printed	5 X X 0 6 0 0 0 0 7 X 8
4 = = 0 = = - = - = - = - = -	(Don't worry if you misse play. Simply, use backar MOVE ? display, and the be in the X-register)	d the I PLAY xx dis row to clear the last HP's move will

10716790	and of a same	
12343678	end of a game	TEST GADE
	-	to test that the program
2.0.%		is correctly loaded: dia-
3000 \$ 00 \$ 0		gonal opening, HP first
		ME HP
$5 \times 0000 \times 0$		- 65
6 % 0 0 0 0 0 0 0 0 0		46 33
7 ¥8000000	I am not very sure	64 63 only the
\$ 0 0 0 0 0 0 0 0 8	whether I have a	43 66 moves are
	legal move or not,	72 53 shown, not
MOVE ?	so I leave the ma-	67 81 the flipped
-1 RUN	chine to make my	42 68 pieces.
NO NOVE	move. It reports	75 36
1 PLAY 28	NO MOVE, and plays	35 84
FLIP 8 PCES	28, flipping 8	06 51
	pieces (22,23,24,	31 56
	25,26,27,37,46)	27 10
12345678	the final hound	
	the linal board	53 50
200000000	is princed,	(6 41
300080000		61 34
4 * * n nn * 0		62 (4
5 \$ \$ 0 0 0 8 0		24 13
6 20000000		2) 10
7 * 6 6 6 6 6 6 6		20 52
200000000		32 47
ಲ್ ಬ್ಯಾಗೆ ಇವರೆ ಇವರೆ ಇವರೆ ಸಾರ್ವ ಇವರ ಇವರ	the counting of	23 74
	the pieces is	15 73
COME IS OVED	performed, and	17 31
GREE IO UTEN	I lose	30 40
UD- NO VOIL (5		70 02
nr 40) 100- 10	HP won	11 87
1 1011		12 11
1 MUN		0 21
		77 88
		22 28
	Final score :	17 47 HP WON

REMARKS : running times (not including your thinking times)

- a typical game (30 HP's moves):
- without printer : 25 minutes
- printer, SF 02 : (1 board)60 minutes
- printer, CF 02 : (2 id.)75 minutes

as you may see, printer slows down significantly the running 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. Execution speed gets faster as the game goes.

- no moves are random, so the same game is played if you make the same moves. This is useful: if you made a mistake that allowed HP to won, you can repeat the game once again, this time avoiding the error, to see who wins now.

- do not make changes to the program unless you are very sure what you are doing. Some parts are very critical.

- level of play is quite good. Children are consistently defeated, as are unexperienced people. Any improvements to the strategy are welcomed, however.

- there are several ways of making room for improvements, or to fit the program into 2 RAMs. 1) delete lines 68,69, change line 01 to LBL"0", line 260 to "OK", and shorten other alpha comments; this saves about 27 or 30 bytes at almost no cost. 2) if you have no printer, or do not want printing of the board, delete lines: 06,62,195 thru 251,254,255,256,257,258, and change line 49 to 60 instead of 61. This saves about 116 bytes. 3) you may use data cards: delete lines 07 thru 30, and insert in their place: 07 16.027, 08 RDTAX. This saves 148 bytes, but a card reader is needed, and you must load a data card with the constants stored on it at the beginning.

VALENTIN ALBILLO (4747)

01	LBL"OTHELLO"	69	AVIEW	137	STO 02	205 45
02	CIRG	70	SF 29	138	RCL 09	206 900 01
03	FIX O	71		139	FC? 03	207 79
05	CF 29	72	CF 03	140		208 STO 02
06	CF 12	74	LBL 08	141	LBL 01	209 2.01
07	.8188111883	75	16.027	143	RCL OO	210 510 05
08	STO 16	76	FS?C 29	144	RCL IND 02	212 SKPCOL
09	.8661683138	77	21	145	+	213 49.056
10	STO 17	78	STO 05	146	STO 03	214 STO 04
11	•13 16 636633	79	LBL 11	147	RCL IND X	215 LBL 02
12	STO 18	80	RCL IND 05	148	RCL 04	216 RCL 13
13	• 3604055150	01		149	X¥Y?	217 SKPCOL
14	STU 19 A1A81A156A	82	GIU 05 SF 05	150	URL 03	218 X()Y
16	•4140141J04 STO 20	84	LRL 13	152	LAST X	219 ACCHR
17	.6553564346	85	$\frac{10}{\text{RCL}}$ 10	153	ST + 03	220 ISG X
18	STO 21	86	x12	154	RCL IND 03	221 GIU UZ
19	• 3435747552	87	x	155	RCL 04	223 28.035
20	STO 22	88	STO 06	156	X=Y?	224 STO 05
21	•5742472425	89	INT	157	GTO 03	225 LBL 09
22	STO 23	90	XEQ 12	158	CHS	226 RCL 04
23	•7376626732	91	FC ?C 04	159	X≠Y?	227 ACCHR
24	STO 24 -	92	GPO OO	160	GPO 12	228 RCL 15
25	•3723260207	93	RCL U6	101		229 SKPCOL
20	510 2)	94 95	x.40?	162	LAST X	230 SF 12
28	STO 26	96	GTO 13	164	ST = 03	231 <u>LBL 10</u>
29	.1772772227	97	FS? 05	165	RCL OO	232 ROL 100
30	STO 27	98	STO IND 05	166	RCL 03	234 +
31	SIGN	9 9	LBL 05	167	X=Y?	235 RCL IND X
32	STO 62	100	ISG 05	168	GTO 12	236 ACCHR
33	STO 09	101	GTO 11	169	RCL 08	237 RCL 03
34	CHS	102	'NO MOVE'	170	ST _X IND Y	238 SKPCOL
35	SI'U 08	103	AVIEN TONE O	171	SI = 01	239 ISG 05
30	9	104	PSE	172	LBL 12	240 GI'O 10
38	STO 15	106	LBL 00	174	TSG 02	241 PRBUE
39	CHS	107	FS?C 01	175	GTO 01	242 SI + 0 J
40	STO 14	108	GTO 14	176	RCL O1	244 ISG 04
41	+	109	"MOVE ?"	177	X=0?	245 GTO 09
42	STO 11	110	PROMPT	178	RIN	246 ADV
43	CHS	111	X=0?	179	CF 04	247 FS? 03
44	SPO 10	112	GIU 14 ST O2	100	PLAY "	248 GTO 12
45	STO 13	11/	IIYOTIII	182	17	249 ADV
40	CHS	115	x(0?	183	-	200 AUV 251 IBI 10
48	STO 12	116	SF 01	184	ARCL X	252 DSE 07
49	61	117	X(0?	185	AVIEW	253 RTN
50	STO 07	118	GTO 08	186	FC? 01	254 FC? 02
51	"DIAG ?"	119	XEQ 12	187	FC? 03	255 GTO 12
52	CF 23	120	FC ?C 04	188	BEEP	256 FS?C 03
53	AON	121	GTO 14	189	PSE	257 XEQ 06
54	PRUMPI'	122	"LLIPGAL"	190	APCI OI	258 <u>IBL 12</u>
22 56	RCL 09	123	TONE O	171	IL DORSI	259 32
57	FS2C 23	125	GTO OO	193	AVTEW	260 "GAME 15 OVE
58	x()y	126	LBL 12	194	PSE	201 20.100 262 AUTEW
59	STO 61	127	SF 04	195	FC? 02	263.0
60	X()Y	128	17	196	GTO 06	264 LBL 07
61	STO 71	129	+	197	FS? 03	265 RCL IND Y
62	XEQ 06	130	STO OO	198	GTO 12	266 +
63	"HP 1ST ?"	131	KOT TND QO	199	TRT 06	267 ISG Y
64 65	PROMPT'	132	RTN	200	FUT 55	268 GTO 07
66	ESPC 22	134	CF 05	207	ADV 12	269 2
67	GTO 00	135	STO 01	202	31	210 / 271 x()v
68	"I MOVE"	136	8.015	204	STO OO	272 RDN

288 "YOU" 278 "+, YOU: " 283 PSE 273 ST-Z 279 ARCL Y 284 X=Y? 289 "F WON" 274 + 275 ADV 280 AVIEW 285 STOP 290 PROMPT 276 "HP: " 281 BEEP 286 "I" 291 END 277 ARCL X 282 ADV 287 X(Y? 291 lines, 672 bytes (3 cards), SIZE 117 (3 RAMs) registers: 00 thru 116 ; flags: 01 thru 04 ; labels: 00 thru 14 run times: execution gets faster as program progresses. A move at the beginning may take at most 60-70 seconds; in the middle, 30-40 seconds; at the end, 12-18 seconds. no printer : 30 moves in 25 minutes printer, SF 02: id in 60 minutes printer, CF 02: id in 75 minutes VALENTIN ALBILLO (4747)