

Notes on the back story of this letter:

This 8-page letter is one of the very last letters I sent to **Richard Nelson** before I went abroad to fulfill my military obligations, and when I eventually came back many months later I was all too busy with real-life issues to dedicate any more time to calculators and related matters. By then I had already sold my **HP-41C** (for the last time) and even my beloved **SHARP PC-1211** (which I later regretted, *again*, and this time I didn't even need the money ! ...) and so the **HP-11C** (presented to me, not bought by me) was the one and only calc I kept (it *still is* with me, as new !), which was more than enough. Thus, my then-strong calc hobby went into hibernation for many years, which was for the best as things **PPC** would get utterly *sour* and *nasty* soon after, and thankfully I was spared it all.

As this present letter was essentially the last one where I would "talk" to *Mr. Nelson* (one-way communication, as always) I threw "*political correction*" aside and went into mild "*rant mode*", giving him a piece of my mind for the last time, in particular asking (among other things) for the criteria used to select which materials would be published and which wouldn't ever make it. I also asked about the blatant *bias* on publishing time and again the umpteen ever-so-slightly-improved version of this or that *tool* in detriment of publishing real *applications* created (or not) using said tools. Anyway, I didn't care for or expect an answer (there was none, of course) as I was going to leave it all behind immediately.

Just for the sake of it, I also included one last contribution for its possible publication in *Mr. Nelson's PPC CJ*, namely my "**Chess 5x5**" program for the **HP-41C**. Actually, I didn't care in the least whether it would be eventually published there or not, as *PPC CJ* was the *last* place I submitted it to: I had already submitted it to *Melbourne Chapter's Technical Notes* and both the *US* and *Europe HP Users' Libraries* for good, and all of them featured it (and also included it in the "**HEWLETT-PACKARD HP-41 USERS' LIBRARY SOLUTIONS**" **Games II** book !!), so even if *Mr. Nelson* decided not to publish it in *PPC CJ* I wouldn't mind at all, my last barbed remark notwithstanding.

End of an age for me.

Valentin Albillo, 13-06-2022

Richard Nelson
Editor, PPC Journal
2541 W. Camden Place
Santa Ana, CA 92704
U.S.A.

Valentin Albillo (4747)
Padre Rubio, 61-2ºC
Madrid 29
SPAIN

June 13, 1981

Dear Richard:

I could ask the standard "how are you?", but I guess that now that the ROM is finished, the word best describing your condition should be "relaxed", or the like.

Do you remember me? It must be difficult to remember particular members among some 7000 numbers, but I've made some contributions in the past to the PPC Calculator Journal, though just a few of them received the high honor of being published there. For unknown reasons, some of the best never appeared; the most relevant example probably being the program "Othello", submitted for its publication nearly a year ago, and never published. The program was, fortunately, published by the Melbourne Chapter PPC TN, and it generated a big deal of interest, even some members created bar code for it. I recently received a letter from Corvallis (see attached photocopy), asking for permission to include the program in the American Users' Library. As it seems, someone in Melbourne took the care of sending the program to them and, as they say in the letter, it generated a lot of interest.

Now, I use to always say what I think, regardless of it being adequate to standard social behaviour or not, and the question is (crudely exposed): if Corvallis is interested, why you aren't? Don't you think the membership will enjoy the program? Why don't you write an article in a future PPC telling the membership the criteria used in editor's censorship? Is a normal ("normal" means non-synthetic) program preferred to a synthetic one? Is an article on the last stupidity on synthetics preferred to a program? Are we ("we" means most PPC members) creating programs that serve to create another programs in a never-ending loop? I am pointing now at those Key Assignments, Byte Loaders, etc, that are very useful as tools to do something with them, and the question is: why don't stop perfecting, improving, revising to a limit the tools, and begin to do something with them?

Here included is some contribution. It is:

CHESS 5x5 : This program allows the user to play a game of chess against the 41c. Actual chess is played using actual rules, as all standard rules are implemented, including pawn promotion. The board is 5x5 instead of 8x8, due to memory and speed limitations, but this hardly matters, as both players have all pieces of standard chess, with the same powers and limitations, arranged in the same order: that's a king, queen, bishop, knight, rook, and a row of pawns. The program makes extensive use of the graphic capabilities of the printer, and prints the board using visually beautiful BLDSPEC characters. If a printer is not present, it runs the same, however. The level of play is surprisingly good for such a tiny program (921 bytes), and the average time for a machine move is some 5 minutes. Additionally, a synthetic routine can be added that speeds the execution if printer present. The machine easily checkmates you if you don't play fine enough. Magnetic cards included (9 tracks+1 track+1 data card: MCHES and P are independent, separate programs. SIZE EXACTLY TO 097).

That's all. I think this is the 1st and possibly the last chess program written for the 41c, and will surely interest to the membership. Let's hope it does not follow the same destiny as poor "Othello".

Yours sincerely:

(4747)

Note: both originals & good photocopies included, to overcome printouts fading.

= MCHES5 - A mini-chess program for the 41c =

This program allows 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 rules for chess are implemented, including pawn promotion. The program is absolutely printer-compatible, but if a printer is present, it will print the board. If a printer is used, you may have the board printed after every move, or only after HP's moves, to save paper.

1 MOVE
FROM 15 TO 45, CHECK

```

1 2 3 4 5
1  A A A A A
2  A A A A A
3  A A A A A
4  A A A A A
5  A A A A A

```

I originally wrote an 8x8 game, but:

- an 8x8 board cannot be printed using BLDSPC characters because of printer buffer limitations. A board could be printed using numbers to identify each piece, or characters, but - even the best attempt was much worse and un-

recognizable than the present version.

- 8x8 game took the full memory of a 41c with 4 single density RAMs, so that, unless you have a quad RAM or a - 41cV, neither the printer nor the card reader could be plugged , making very difficult to load and run the program.

-8x8 game, using the same playing lo - gic as this 5x5 version, took several hours for each move, playing very weak, and thus - making the game uninteresting.

On the other hand, this 5x5 version - provides the following advantages:

- the 5x5 board is printed using BLDSPCs, so you can clearly see the position.
- Though the board is 5x5, you still have all pieces of conventional chess, arranged in - the same order: king, queen, bishop, knight, - rook, and a row of pawns, all with the same powers and restrictions of standard game.

-this 5x5 version fits in 3 sdRAMs, and takes just 5 minutes (average) per move. Also, due to the reduced size, a typical game - takes an average of 20 moves (8x8 games take some 40 moves), making the game "faster" and more active. The armies get into battle very soon. And also, as the number of positions and moves is less than in 8x8, the level of play is much better, so that the 41c plays a weak, but non-trivial game. It can even checkmate you if you don't play fine enough !!

```

1 2 3 4 5
1  A A A A A
2  A A A A A
3  A A A A A
4  A A A A A
5  A A A A A

```

1 MOVE
FROM 42 TO 53

```

1 2 3 4 5
1  A A A A A
2  A A A A A
3  A A A A A
4  A A A A A
5  A A A A A

```

FROM?

```

1 2 3 4 5
1  A A A A A
2  A A A A A
3  A A A A A
4  A A A A A
5  A A A A A

```

HP 1ST?		
N		RUN
FROM?	43	RUN
TO?	33	RUN

1 MOVE
FROM 22 TO 33

- All standard chess rules are implemented, with the following exceptions:

- as the king is already in a corner (see standard initial position above), no castling is necessary: it is protected by the queen
- as there is only one empty row between the pawns, a pawn may advance just one position on its first move (not 1 or 2 as in standard 8x8)
- no capture "en passant" is allowed

All other rules are the same: pawn promotion is allowed: if a pawn reaches the opposite side, it becomes any desired piece (except king or pawn, of course), as in standard chess. Such an example is given in the illustration at the left: HP moves - the pawn in 42 (standard matrix row/column notation) to 53, thus taking the white bishop at 53, becomes a queen - (see illustration) and gives check (not indicated). In case of - pawn promotion, HP will always select a queen, but you may choose any desired piece.

PROGRAM CHARACTERISTICS

This program, called MCHESS, is composed - of 3 separate files: MCHESS, P, and 5. The main file, MCHESS, is independent of the others, and if you do not intend to use a printer, just load MCHESS (921 bytes, 9 tracks). However, if you have a printer and want printing of the board, load the P (print board) routine, too. Neither MCHESS nor P - use synthetics at all. But you may have noticed that plugging in the printer slows down the execution - speed by some 35%. So, you may want to also load the routine called 5: this is a synthetic routine that toggles flag 55 on and off. It is called at proper times from the main file (if printer), so disabling the printer when not necessary. The operation of 5 is transparent to the user: the program runs the - same with or without it, but it runs appreciably - faster if 5 is used. In fact, it runs almost as - fast as if no printer is plugged in. Both routines, P and 5, fit together in exactly 112 bytes, and can be stored on the 10th track, so the whole program takes 5 cards.

```

1 2 3 4 5
1 11111 111
2 11111 111
3 11111 111
4 11111 111
5 11111 111

```

I MOVE
FROM 42 TO 53

```

1 2 3 4 5
1 11111 111
2 11111 111
3 11111 111
4 11111 111
5 11111 111

```

FROM?

In addition, a separate - data card is used, that contains all, BLDSPEC characters used by the prin - ter, as well as other useful constants.

You should prepare this data card previously to running the program. The card is used at the beginning of each game. It contains numeric constants in R18 thru R36 and BLDSPEC alpha constants in R37 thru R49. That's 32 registers in all, a whole card. If you do not have a printer, store an alpha A in lieu of the BLDSPEC characters.

The program needs SIZE 097 to run. Very important: the SIZE must be exactly 097. If it - is smaller or larger, it won't work. It has to be no other than 097.

```

1 2 3 4 5
1 11111 111
2 11111 111
3 11111 111
4 11111 111
5 11111 111

```

I MOVE
FROM 43 TO 53

```

1 2 3 4 5
1 11111 111
2 11111 111
3 11111 111
4 11111 111
5 11111 111

```

REMARKS: -you can select whether the board is printed after every move (SF 00) or just after HP moves (CF 00) at any time during the game.

-If a printer is present, the P routine should be present, too. The 5 routine is recommended, then, to speed program execution. But if you do not want synthetics, just load the alternate IBL "5", END, which will do nothing, and no changes to the main file will be needed.

-If some HP move results in a check being given to your king, - the machine places the word CHECK after its move. There are 2 exceptions to this rule: (see illustrations above)

- a) if a pawn promoted to a queen by HP results in a check, this is not indicated (right illust.)
- b) if HP moves a piece that, while not giving check by itself, leaves your king under attack from some other HP piece, the check isn't indicated, too. (left illust.)

-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 thing, HP will actually take your king on its move.

-HP's average "thinking" time is 5 minutes per move. This is the average for a non-printer game. But actual times vary very much with the specific position. The time increases quadratically - with the number of possible moves for HP and linearly with the number of your responses. Minimum times are 15 seconds, typical are 3 or 4 minutes, average is 5 minutes, and maximum some 15 minutes. A little patience is needed! However, a whole game - should last no more than 1½ hours or so.

-do not make any changes to the program, or it may not work. Specially do not insert subroutines, all 6 levels are used up

INSTRUCTIONS

MCHESS-VERSION 2

```

 1 2 3 4 5
1 ♔ ♕ ♖ ♗ ♘
2 ♙ ♙ ♙ ♙ ♙
3 ♚ ♚ ♚ ♚ ♚
4 ♜ ♜ ♜ ♜ ♜
5 ♞ ♞ ♞ ♞ ♞

```

HP 1ST?

N

FROM?

43

RUN

RUN

TO?

33

RUN

```

 1 2 3 4 5
1 ♔ ♕ ♖ ♗ ♘
2 ♙ ♙ ♙ ♙ ♙
3 ♚ ♚ ♚ ♚ ♚
4 ♜ ♜ ♜ ♜ ♜
5 ♞ ♞ ♞ ♞ ♞

```

I MOVE

FROM 24 TO 33

```

 1 2 3 4 5
1 ♔ ♕ ♖ ♗ ♘
2 ♙ ♙ ♙ ♙ ♙
3 ♚ ♚ ♚ ♚ ♚
4 ♜ ♜ ♜ ♜ ♜
5 ♞ ♞ ♞ ♞ ♞

```

FROM?

- SIZE 097 (exactly !!!)
- if you want to print the board, load the P and 5 routines and GTO..
- load the MCHESS main file (do not press GTO.. if your program is compiled, to avoid new recompilation and delays when searching for labels)

- XEQ "MCHESS" → CARD

-load the data card (2 sides); the machine turns itself off, to allow you unplug the card reader and plug the printer instead. (if you do not have printer, or have a quad you can delete the OFF instruction)

-once the printer is plugged in and ON (and set to NORM position) turn on the 41c. The program starts immediately. If printer, the board is printed now reflecting the initial position (see illust. at left):

the pieces are arranged as follows:

black (HP): king, queen, bishop, knight, rock
pawn, pawn, pawn, pawn, pawn

white (you): pawn, pawn, pawn, pawn, pawn
king, queen, bishop, knight, rock

you always play white, and HP plays black, but you can select who moves first:

HP 1ST? , if you want HP to make the first move, simply press R/S. If you want to make the first move, press N, R/S

a) HP moves: it will display I MOVE , and scroll the message to the right while it thinks its move. Then finally, if a printer is present, it prints

FROM xx TO yy , where xx is the row/column position of the piece it moves , and yy is the location where it moves to.

If the piece gives check, FROM xx TO yy, CHECK is displayed (see REMARKS for exceptions). Then the board is printed. (if you have no printer, the execution stops to allow you to see and perform the move on the board. If you fail to see the whole message, turn to ALPHA to see it again. Press R/S to continue). Then, you are prompted for your move with:

FROM? , enter the xx coordinates of the location where your piece stands, then R/S

TO? , enter the coordinates of the location where it moves to. The machine, once you press R/S, proceeds to compute its own move. However, if you just promoted your pawn, after the TO you will be prompted for the piece you want - the pawn to become: PIECE?, enter the code for the selected piece:

the codes are;

queen: 5.09 , bishop: 4.03 , knight: 3.03
rook : 2.05

whenever you are prompted by PIECE? enter the code for the desired piece (most likely a queen, 5.09). If the machine promotes one of its pawns, it will always select a queen. If you are playing without a printer, remember this fact and place a black - queen where the promoted black pawn stands. Of course, you may promote any number of your pawns, and choose any desired piece, so you can have 3 - knights if you want to.

I MOVE

FROM 12 TO 52, CHECK

```

 1 2 3 4 5
1 ♔ ♕ ♖ ♗ ♘
2 ♙ ♙ ♙ ♙ ♙
3 ♚ ♚ ♚ ♚ ♚
4 ♜ ♜ ♜ ♜ ♜
5 ♞ ♞ ♞ ♞ ♞

```

FROM?

-always remember to properly actualize the board after HP moves if no printer is being used.

- b) your moves : just answer the FROM?,TO?, (PIECE?) prompts with the respective coordinates, as stated previously, but you must remember:

-your moves are not tested to be legal. So, please, play carefully and do not make any illegal move, which would spoil the game. Remember that, though HP announces most checks given to your king, there are two exceptions, and if you fail to notice the check and let your king under attack, HP will take your king on its next move

-the machine recognizes the situation when you give checkmate, and displays the proper message CHECKMATE, YOU WON on its move. But see c) end of a game, below.

The game continues, alternating moves between you and HP.

c) end of a game

- a) if you give checkmate : the machine recognizes the fact, - and displays (prints) and beeps the message:

I MOVE → CHECKMATE → YOU WON

- b) stalemated position: a stalemated position is a position - in which either you or HP cannot make any legal move on - their respective turn. HP recognizes the stalemate by it - self if, and only if it has no possible move at all (legal or not). In this case, it displays:

STALEMATE , the game is a tie

however, if HP cannot make any legal move (and its king - is not under check), but has "illegal moves" (such as moving the king to a position under attack from your pieces) it displays the CHECKMATE → YOU WON message. This is not - so, because HP's king is not under attack, so the position is a stalemated one, and the game is a tie, too

If you cannot make any legal move, but your king is not - under check given by HP, the position is a stalemate, too. Just answer the FROM? prompt with \emptyset , then R/S:

FROM? , \emptyset R/S → STALEMATE appears. A tie.

- c) if HP checkmates you : should HP happen to checkmate you, it does not recognize the situation by itself. Just answer the FROM? prompt requesting your move with a minus one:

FROM? , -1 R/S → CHECKMATE → I WON , HP won

DATA CARD : you should prepare a data card with the following - contents:

R18 = 1	, R26 = -7	, R34 = 20.023	, R42=b.pawn
R19 = -1	, R27 = -11	, R35 = 16.023	, R43=dot.square
R20 = -10	, R28 = 19	, R36 = 16.023	, R44=white pawn
R21 = -8	, R29 = 17	, R37 = black king	, R45=w.rook
R22 = 10	, R30 = 7	, R38 = black queen	, R46=w.knight
R23 = 8	, R31 = 11	, R39 = black bishop	, R47=w.bishop
R24 = -19	, R32 = 16.019	, R40 = black knight	, R48=w.queen
R25 = -17	, R33 = 24.031	, R41 = black rook	, R49=w.king

that's 32 (1 card) registers in all. Registers 37 thru 49 contain the BLDSPEC characters used to generate the board. If you do not have a printer, store an alpha A into R37 thru R49. If - you have a printer, you can construct your own BLDSPEC characters and store them in the registers as shown above (the character for the black king in R37, etc). If you want your board to look like the one in the illustrations, use these BLDSPEC nos.:

R37 = black king = 0 96 122 127 122 96 0

SAMPLE GAME & RUNNING TIMES

Sample game, you first (times = T)

remarks: -as you can see, the combination printer+routine 5 slows down the program by a factor of 1.1, while using the printer without 5 slows it down by a factor of 1.52. -this is not a standard average sample game: most - games are faster than this one. For instance, the move 15-14 which takes 14'40'' to be found is a maximum. The 41c had to ex plore nearly 1000 alternatives (i.e: moves) to find it, so - the time had to be large. In that position, HP had 26 possible options, each option having at least 19 responses from you.etc.

left & right : typical
beginning of a game

and HP computes its move now, giving check with its pawn, when taking the white one at location 42

1 1 1 1 1 1
2 1 1 1 1 1
3 1 1 1 1 1
4 1 1 1 1 1
5 1 1 1 1 1

	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1

your move?
setting flag
OO causes the
board to be
printed now

FROM 31 TO 42, CHECK

	1	2	3	4	5
1	全	全	全	全	全
2	全	全	全	全	全
3	全	全	全	全	全
4	全	全	全	全	全
5	全	全	全	全	全

43 RUN

T02

27 RUN

<pre> 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ FROM? 41 RUN TO? 31 RUN 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ I MOVE FROM 33 TO 42, CHECK 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ FROM? 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ I MOVE FROM 53 TO 31 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ FROM? 0 RUN STALEMATE </pre>	<pre> left typical sequence of a game: HP has just moved your move? you move your pawn from 41 to 31 flag 00 was set so the board is printed reflecting your move. Now HP takes your knight at 42 with its pawn and gives check to your king the board your move? left in this posi- tion, you are left with the king and a - pawn at 34. HP moves its bishop from 53 to 31 but, as the board reflects, now you cannot move at all, because all - possible moves aren't legal. It is a stale- mate, as your king is not under check. </pre>	<pre> right: end of a game. You move your knight from 54 to 33, giving check to the black king at 21 (the piece at 12, which is the queen is also - under attack) the board is printed, and HP acknow- ledges the checkmate. right: end of a game. HP takes your rook at 55 with its queen at 35, and gi- ves check. As the bo- ard shows, you cannot defend your king, so you've been checkmated the machine acknowled- ges this. right: end of a game your queen, at 22, defended by your bishop at 33, gives checkmate to the king at 12 </pre>	<pre> 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ FROM? 54 RUN TO? 33 RUN 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ I MOVE CHECKMATE YOU WON 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ I MOVE FROM 35 TO 55, CHECK 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ FROM? -1 RUN CHECKMATE I WON 1 2 3 4 5 1 ♠ ♠ ♠ ♠ ♠ 2 ♠ ♠ ♠ ♠ ♠ 3 ♠ ♠ ♠ ♠ ♠ 4 ♠ ♠ ♠ ♠ ♠ 5 ♠ ♠ ♠ ♠ ♠ I MOVE CHECKMATE YOU WON </pre>
--	--	---	--

That will be all for now. Excuse the quite long documentation, but the complexity of the program made it unavoidable. As always, I hope you will like it.

VALENTIN ALBILLO (4747)

01+LBL "MCHES"	68 XEQ 06	135 BEEP	202 -	269 RCL 02	336 RCL 08	403 RCL IND X	470 .4	01+LBL "P"
02 CLRG	69 STO 00	136 AVIEW	203 46	270 8	337 XXY?	404 STO 14	471 FS? 03	02 ADV
03 FIX 0	70 "TO?"	137 FS? 55	204 +	271 XEQ 09	338 RTN	405+LBL 28	472 -	03 SF 12
04 CF 29	71 PROMPT	138 XEQ "P"	205 RTN	272 FS? 18	339 SF 08	406 RCL 11	473 FS? 02	04 9
05 10.049	72 XEQ 06	139 FC? 55	206+LBL 12	273 RTN	340 9	407 STO 15	474 XEQ 13	05 SKPCOL
06 RDTAX	73 STO 01	140 STOP	207 STO 03	274 X<=0?	341 ST+ 06	408+LBL 29	475 RCL 06	06 49.053
07 SF 11	74 CLX	141 GTO 99	208 ABS	275 RTN	342 RTN	409 RCL IND 14	476 X<>Y	07 STO 13
08 OFF	75 X<> IND 00	142+LBL 07	209 CF 05	276+LBL 12	343+LBL 13	410 ST+ 15	477 -	08+LBL 00
09 9	76 STO IND 01	143 60	210 CF 06	277 CF 08	344 FS? 06	411 RCL 15	478 RCL 00	09 ACCHR
10 STO 16	77 XEQ 07	144 RCL 01	211 CF 07	278 STO 07	345 GTO 13	412 XEQ 08	479 X<>Y	10 2
11 ST- 17	78 FC? 55	145 X>Y?	212 2	279 FRC	346 RCL 03	413 FS? 18	480 X<=Y?	11 SKPCOL
12 6.5	79 GTO 00	146 RTN	213 X>Y?	280 1 E2	347 30	414 GTO 00	481 SF 19	12 X<>Y
13 STO 92	80 FS? 00	147 2	214 GTO 13	281 *	348 -	415 X>0?	482 X<=Y?	13 ISG X
14 ST- 56	81 XEQ "P"	148 RCL IND 01	215 X<>Y	282 STO 06	349 RCL IND X	416 GTO 00	483 RTN	14 GTO 00
15 5.09	82+LBL 00	149 X>Y?	216 30	283 RCL Z	350 STO 01	417 CF 10	484 RCL 09	15 PRBUF
16 STO 93	83 "I MOVE"	150 RTN	217 +	284 STO 08	351+LBL 03	418 X=0?	485 X<>Y	16 56.06
17 ST- 57	84 AVIEW	151 "PIECE?"	218 XEQ IND X	285 .4	352 RCL 08	419 SF 10	486 XXY?	17 STO 15
18 4.03	85 CF 17	152 PROMPT	219 RCL IND X	286 FS? 07	353 STO 10	420 XEQ 13	487 STO 09	18+LBL 01
19 STO 94	86 FS? 55	153 STO IND 01	220 STO 04	287 ST- 06	354+LBL 10	421 FS? 19	488 RTN	19 RCL 13
20 ST- 58	87 SF 17	154 RTN	221+LBL 14	288 FS? 06	355 RCL IND 01	422 RTN	489+LBL 13	20 ACCHR
21 1.01	88 FS? 55	155+LBL 08	222 RCL 02	289 XEQ 12	356 ST+ 10	423 FS? 01	490 .5	21+LBL 02
22 STO 03	89 XEQ "5"	156 ABS	223 STO 05	290 FC? 07	357 RCL 10	424 GTO 00	491 +	22 2
23 STO 04	90 PI	157 2	224+LBL 15	291 XEQ 13	358 XEQ 00	425 FS? 10	492 RCL Z	23 SKPCOL
24 STO 05	91 STO 09	158 XXY?	225 RCL IND 04	292 RCL 00	359 FS? 10	426 GTO 29	493 60	24 RCL IND 15
25 STO 06	92 96.055	159 RTN	226 ST+ 05	293 RCL 06	360 GTO 00	427+LBL 00	494 X<>Y	25 INT
26 STO 07	93 STO 02	160 92	227 RCL 05	294 X<=Y?	361 XEQ 12	428 ISG 14	495 CF 04	26 43
27 ST- 65	94 CHS	161 RCL 13	228 XEQ 00	295 RTN	362 X=Y?	429 GTO 28	496 X<=Y?	27 +
28 ST- 66	95 STO 00	162 XXY?	229 FS? 18	296 RCL 03	363 RTN	430 RTN	497 SF 04	28 RCL IND X
29 ST- 67	96+LBL 11	163 RTN	230 GTO 00	297 FS? 08	364 FS? 05	431+LBL 36	498 RCL Z	29 ACSPEC
30 ST- 68	97 RCL IND 02	164 -5.09	231 X<0?	298 -5.09	365 GTO 00	432 SF 03	499 9	30 ISG 15
31 ST- 69	98 SIGN	165 STO IND 13	232 GTO 00	299 STO IND 00	366 LASTX	433+LBL 33	500 FC? 04	31 GTO 02
32 INT	99 X=0?	166 RTN	233 CF 09	300 CLX	367 X=0?	434 SF 01	501 CLX	32 4.009
33 -	100 GTO 00	167+LBL 04	234 X=0?	301 STO IND 02	368 GTO 10	435+LBL 32	502 +	33 ST+ 15
34 STO 95	101 LASTX	168 ASTO X	235 SF 09	302 56.096	369+LBL 00	436+LBL 34	503 RTN	34 ISG 13
35 ST- 59	102 X<0?	169 "CHECKMATE"	236 XEQ 12	303 STO 11	370 ISG 01	437+LBL 35	504+LBL 09	35 GTO 01
36 2.05	103 XEQ 12	170 AVIEW	237 FS? 05	304 STO 09	371 GTO 03	438 RTN	505 +	36 ADV
37 STO 96	104+LBL 00	171 BEEP	238 GTO 00	305 CF 19	372 RTN	439+LBL 12	506+LBL 00	37 ADV
38 ST- 60	105 DSE 02	172 CLA	239 FS? 09	306+LBL 21	373+LBL 13	440 SF 02	507 CF 18	38 ADV
39 "A"	106 GTO 11	173 ARCL X	240 GTO 15	307 RCL IND 11	374 RCL 08	441 RCL 11	508 SF 25	39 CF 12
40 50.091	107 FS? 17	174 "I WON"	241+LBL 00	308 SIGN	375 RCL 22	442 RCL 17	509 RCL IND X	40 RTN
41 SIGN	108 XEQ "5"	175 PROMPT	242 ISG 04	309 X=0?	376 XEQ 00	443 XEQ 09	510 SIGN	41+LBL "5"
42+LBL 01	109 RCL 09	176+LBL 05	243 GTO 14	310 GTO 00	377 RCL 08	444 FS? 18	511 FS?C 25	42 RCL d
43 RCL IND L	110 PI	177 "STALEMATE"	244 RTN	311 LASTX	378 8	445 1	512 X=0?	43 STO [
44 X=0?	111 X=Y?	178 BEEP	245+LBL 36	312 X<=0?	379+LBL 00	446 X=0?	513 SF 18	44 ARCL 19
45 ASTO IND L	112 GTO 05	179 PROMPT	246 SF 07	313 GTO 00	380 XEQ 09	447 XEQ 13	514 LASTX	45 ARCL 20
46 ISG L	113 "YOU"	180+LBL 09	247+LBL 33	314 XEQ 07	381 FS? 18	448 FS? 19	515 END	46 RCL [
47 GTO 01	114 -25	181 INT	248 SF 05	315 FS? 19	382 RTN	449 RTN		47 STO d
48 SREG 74	115 RCL 00	182 ENTER↑	249+LBL 32	316 GTO 04	383+LBL 12	450 RCL 11		48 FC?C 15
49 CLX	116 XXY?	183 ENTER↑	250+LBL 34	317+LBL 00	384 INT	451 RCL 20		49 SF 15
50 ASTO 79	117 GTO 04	184 9	251+LBL 35	318 ISG 11	385 6	452 XEQ 00		50 RCL d
51 FS? 55	118 CLX	185 /	252 RTN	319 GTO 21	386 X*Y?	453 FS? 19		51 STO [
52 XEQ "P"	119 X<> IND 12	186 INT	253+LBL 13	320 RCL 09	387 RTN	454 RTN		52 "I-12"
53 CF 23	120 STO IND 13	187 +	254 SF 06	321 STO 00	388 .41	455 RCL 11		53 RCL \
54 AON	121 XEQ 00	188 51	255 RCL 02	322 RCL 02	389 ST+ 06	456 RCL 21		54 STO d
55 "HP 1ST?"	122 "FROM "	189 -	256 9	323 STO 12	390 RDH	457+LBL 00		55 END
56 PROMPT	123 RCL 12	190 ARCL X	257 XEQ 09	324 RCL 08	391 RTN	458 XEQ 09	LBL"P	
57 AOFF	124 XEQ 09	191 RTN	258 FS? 18	325 STO 13	392+LBL 07	459 FS? 18	LBL"5	
58 FC?C 23	125 "I TO "	192+LBL 06	259 1	326+LBL 04	393 CF 01	460 RTN	END	112 BYTES
59 GTO 00	126 RCL 13	193 ENTER↑	260 X=0?	327 RCL 03	394 CF 02	461 X*0?	LBL"MCHES"	
60+LBL 99	127 XEQ 09	194 ENTER↑	261 XEQ 12	328 STO IND 02	395 CF 03	462 X>0?	END	921 BYTES
61 "FROM?"	128 RCL 00	195 1	262 RCL 02	329 RCL 07	396 2	463 RTN		
62 PROMPT	129 FRC	196 -	263 RCL 22	330 STO IND 08	397 X>Y?	464+LBL 13		
63 "I"	130 RCL 22	197 5	264 XEQ 09	331 RTN	398 GTO 12	465 FRC	STATUS:	
64 X<0?	131 *	198 /	265 FS? 18	332+LBL 12	399 X<>Y	466 ABS	SIZE= 097	
65 GTO 04	132 FRC	199 INT	266 CLX	333 .5	400 30	467 1 E2	Σ= 74	
66 X=0?	133 X*0?	200 2	267 X>0?	334 ST+ 06	401 +	468 *	DEG	
67 GTO 05	134 "I, CHECK"	201 /	268 XEQ 12	335 92	402 XEQ IND X	469 FS? 03	FIX 0	