

Program Title CHESS 5x5
 Contributor's Name VALENTIN ALBILLO
 Address PADRE RUBIO, 61-20C
 City MADRID 29 State/Country SPAIN Zip Code _____

Program Description, Equations, Variables 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.
 I MOVE FROM 15 TO 45, CHECK 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.

```

  1 2 3 4 5
  1 [ ] [ ] [ ] [ ] [ ]
  2 [ ] [ ] [ ] [ ] [ ]
  3 [ ] [ ] [ ] [ ] [ ]
  4 [ ] [ ] [ ] [ ] [ ]
  5 [ ] [ ] [ ] [ ] [ ]
  
```

FROM?

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 takes 7. The board could be printed using numbers to identify each piece, 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 (4 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.

Necessary Accessories 3 memory modules, card reader, and optionally, printer

Operating Limits and Warnings -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 be EXACTLY 097 (no more, no less)
 -do not make any changes to the program, unless you want it to have bugs. Specially, do not add any subroutines: all 6 levels are used up.

Reference(s) Martin Gardner described the 5x5 version of chess in one of his remarkable books on Recreative Mathematics.

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.

(CONTINUATION FORM)

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

- 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.
- Though the board is 5x5, you still have all pieces of - conventional chess, arranged in the same order (see illustration at the left) : king, queen, bishop, knight, rook, and a row of pawns. All pieces have the same powers and - restrictions as in standard chess.
- 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 moves), making the game "faster", 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 !!

the following 3 exceptions:

- as the king is already in a corner (see standard initial position at the upper left), no castling is necessary.
- 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.

As you may see, these exceptions are mostly due to the size of the board. 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 upper right: HP moves its pawn in 42 (standard row/column matrix notation: 4 is vertical, 2 is horizontal, numbers) to 53, thus taking the white bishop at that location (by the way, you play white always), - becomes a queen (see printout) and gives check (not shown). In - case of pawn promotion, HP always selects a queen, but you may - chose any desired piece.

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 printouts at both left and right of these lines):

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

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 next move !

```

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

```

```

HP 1ST?
N          RUN
FROM?

SF 00
41      RUN

T0?
31      RUN

```

```

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

```

```

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

```

I MOVE
FROM 42 TO 53

```

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

```

I MOVE
FROM 43 TO 53

```

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

```

```

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

```

I MOVE
FROM 42 TO 53

```

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

```

(CONTINUATION FORM)

Here are some characteristics of this program: This program, called MCHESSE, is composed of 2 separate program files: MCHESSE itself, and P. The main file, MCHESSE, is independent of the other, and if you do not intend to use a printer, all you need is to load MCHESSE alone (921 bytes, 9 tracks). However, if you have a printer and want printing of the board, you should load, separately, the P subroutine (P stands for print board). MCHESSE calls, if printer present, P at proper times. You can notice the convenience of having the print board routine separated from the main program: those users without printer do not waste memory. Also, you can create your own subroutine, without having to change the main program.

The P routine fits into 1 track, the 10th track, so the whole combination MCHESSE+P takes 5 full magnetic cards. You'll notice that using the printer slows down program execution. See "Execution times"

In addition, a separate data card is used that contains all BLDSPEC characters used by the P routine to print the pieces, as well as other useful constants. You must load this data card at the beginning of each game. Here are the contents of the card: (registers 18 thru 49, that's 32 registers in all)

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

as you may see, R18 thru R36 contain numeric constants, while R37 thru R49 contain alpha BLDSPEC characters, which represent each piece. If a user without a printer were to create these card, he would have to load any alpha characters instead of the BLDSPEC characters, because the program require that registers 37 thru 49 be loaded with alphas to run properly, whether it prints or not.

If you want to, you can construct your own BLDSPEC characters and store them in their appropriate register (ie, the character for the black king must be in R37, etc). The board you can see in the printouts uses the following BLDSPEC numbers for each character:

R37=black king	= 0,96,122,127,122,96,0	R49=white one=	112,95,69,64,69,95,112
R38= id. queen	= 0,96,114,127,114,96,0	R48= id.	= 112,95,77,64,77,95,112
R39= id. bishop	= 0,100,110,123,110,100,0	R47= id.	= 110,91,81,68,81,91,110
R40= id. knight	= 0,108,102,119,126,108,0	R46= id.	= 110,83,89,72,65,83,126
R41= id. rook	= 0,102,124,126,124,102,0	R45= id.	= 103,89,67,65,67,89,103
R42= id. pawn	= 0,96,102,126,102,96,0	R44= id.	= 96,95,89,65,89,95,96

R43= 85,0,65,0,65,0,85

You can select whether the printer prints the board after every move, or just after HP moves. To select the first option, simply set flag 00 (SF 00). To select the second option (which saves paper and time), clear flag 00 (CF 00). This can be done at any time, when the machine is at a halt. By the way, if the printer is plugged in, the P routine should be present, too.

HP's average "thinking" time is 5 minutes per move. This is an average for non-printer game. Actual times vary very much with the position, from a minimum of some 15 seconds, to a typical 3 or 4 minutes, average 5 minutes, and maximum of some 15 minutes. However, a whole game should last no more than 1½ hour. If a printer is used, multiply these times by the factor 1.52 (52% slower)

(CONTINUATION FORM)

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 being 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 - and 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	so, the king is considered to have a value of 50, the
queen = 5.09	queen is 9, rook is five, bishop & knight are of the-
bishop = 4.03	same value, and pawn is worth 1. This is accordingly-
knight = 3.03	to the standard chess valuation for pieces.
rook = 2.05	
pawn = 1.01	Empty locations have a \emptyset 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)
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 evaluation routine:

-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 used thus, resembles the alpha-beta algorithm used in computer chess programs.

```

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

```

HP 1ST?

N

FROM?

41 RUN

TO?

SF 00

31 RUN

```

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

```

I MOVE

FROM 22 TO 31

```

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

```

FROM?

43 RUN

TO?

33 RUN

```

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

```

I MOVE

FROM 31 TO 42, CHECK

```

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

```

SAMPLE GAME : let's try it. Make sure you have a size of EXACTLY 097 (The size must be exactly 097, no more, no less, because the error flag is used to detect nonexistent registers above 096, which saves 20 registers. If the size were greater than 097, some of these registers would exist, causing errors). Load the printer P routine (if you have no printer, skip this procedure), then press: (printer in NORM)

(shift) GTO .., see PACKING momentarily

Now load the main file MCHESSE. Press:

XEQ (alpha) MCHESSE (alpha), see CARD

pass the data card thru the card reader. As soon as both tracks are read, the machine turns itself off, to allow you unplug the card reader, and plug the printer instead. Once this is done, (if you have no printer ignore this), turn on the machine. The program starts immediately:

-see printout at the left, the initial position is printed. Pieces are:

black (HP) : king, queen, bishop, knight, rook
paw, paw, paw, paw, paw

white (you): paw, paw, paw, paw, paw
king, queen, bishop, knight, rook

-you are prompted with HP 1ST? to know who makes the first move. You want to make the move, so press: /S.

N and then R/S → FROM?

-enter the location where your piece is:

41 then R/S → TO?

-enter the location where it moves to :

31 then R/S → I MOVE is displayed while HP thinks.

(in the printout, flag 00 was set before entering the 31, to force printing of the board always)

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 prints both at the left and at the right: you advance your pawn at 43, then HP captures your pawn at 42, giving check. You move your king, and HP (see printout at top right) moves once more its pawn to 53, capturing your bishop and being promoted to a queen (See the 2nd black queen at 53 !). You now decide to capture the pawn at 24 with your pawn at 33 (previously flag 00 was cleared, to avoid printing the board after - your move), 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 queen at 12, and the bishop at 13 impedes your retreat. You input -1, and HP acknowledges the victory with a happy CHECKMATE → I WCN message. Better luck next time !

FROM?

51 RUN

TO?

41 RUN

```

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

```

I MOVE

FROM 42 TO 53

```

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

```

FROM?

CF 00

33 RUN

TO?

24 RUN

I MOVE

FROM 53 TO 52, CHECK

```

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

```

FROM?

-1 RUN

CHECKMATE

I WCN

(CONTINUATION FORM)

```

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

END of Game examplesleft

in the position shown, you move your knight 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 deduces you have - given checkmate. So it displays the I MOVE, CHECKMATE and YOU WON messages.

SAMPLE GAME & TIMES

if desired, test that your program is correctly loaded by running this game:

Check is indicated with a + sign, and numbers in - brackets represent the co

de of pieces obtained by pawn promotion (answers to the PIECE? prompt). Times are given, too. You play first:

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, HP 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.

right

in this position, HP moves and wins. Black move its queen from 35 to 55 taking your rook at 55, and giving 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 and displays the CHECKMATE, I WON.

right

in this other position, you are 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 0 as your move, and the machine acknowledges the stalemate, displaying STALEMATE. A tie

```

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
FROM 53 TO 31

```

1 2 3 4 5
1 ♔ ♖ ♗ ♘ ♙
2 ♜ ♞ ♝ ♞ ♟
3 ♠ ♢ ♣ ♤ ♡
4 ♚ ♛ ♙ ♜ ♞
5 ♠ ♢ ♣ ♤ ♡

```

FROM? 0 RUN
STALEMATE

USER INSTRUCTIONS

SIZE: 097 exact (HP-41C)				
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	SIZE EXACTLY TO 097. If you want to print the board, load the P routine. Then, press: and load the MCHES program . Press:		GTO .. XEQ (alpha) MCHES (alpha)	PACKING CARD machine off
2	pass the data card thru the card reader if desired, unplug the card reader, and plug the printer on its place. If printer present, set it to NORM position and turn it on. Then, turn on the 41c:			
3	(YOU ALWAYS PLAY WHITE, HP PLAYS BLACK)		ON	program starts immediately
3 or 3	if you want HP to make the first move if you want to make the first move	N	R/S R/S	HP 1ST? I MOVE FROM?
	note: if a printer is present, and you want to print the board just after HP moves, simply press :			
	if you want to always print the board : this can be done at any moment the machine is halted.		CF 00 SF 00	
4	<u>IF YOU MOVE</u> (FROM? is in the display)			
5	-enter the xy coordinates where your - piece stands. X stands for row, Y for column (standard matrix notation) :	xy	R/S	TO?
	-enter the xy coordinates of the location where it moves to :	xy	R/S	I MOVE or PIECE?
	-if the PIECE? prompt appears, you've just promoted your pawn. Enter the code for the selected piece. Codes are: queen: 5.09 , bishop: 4.03 knight: 3.03 , rook : 2.05	code	R/S	I MOVE
or 5	-if you cannot make any legal move, but your king is not under check, enter: you're stalemated. It's a tie.	ø	R/S	STALEMATE
or 5	-if your king is under check and you can't save him, you're checkmated. Enter: HP won this one.	-1	R/S	CHECKMATE I WON
6	<u>IF HP MOVES</u> (I MOVE is in the display) I MOVE is scrolled in the display while it thinks its move, and then:			
7	HP makes a move; where xx is the row/column position of the piece it moves, and yy is the location where it moves to. CHECK is displayed if your king is now under check.			FROM xx TO yy (, CHECK

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	(if you have no printer, remember to actualize the board, moving the black pieces as HP indicates. If HP promotes a pawn, place a black queen on its place. The execution stops with the FROM xx TO yy on the display for you to notice. If you fail to see the whole-message, turn alpha on to see it-again, then alpha off. Press R/S to continue)			
	Then you are prompted for your move with:			FROM?
or 7	-You have checkmated HP, which acknowledges the fact and displays:			CHECKMATE YOU WON
or 7	-HP has no move at all (legal or not) but its king is not under check either. It displays: A tie			STALEMATE
or 7	-HP's king is not under check, and it has no legal moves, but has some illegal ones, such as moving its king to a square under attack from your pieces. It displays: but this is not so, because its king is not under attack. The actual result must be STALEMATE, so please, notice this and concede the tie, will you? Thank you			CHECKMATE YOU WON
<u>Notes:</u>				
-everything that appears on the display is printed as well, and the board is printed after every move if flag 00 is set, and only after HP moves if clear. Remember that you can set or clear flag 00 from the keyboard as often as you like, whenever the machine is halted.				
-your moves are not tested for legality. Do not cheat, please, or you will ruin the game. Be careful not to make mistakes. For instance, though HP will never make illegal moves, if your king is under check and you forget the fact, HP will take it on its next move. Also remember that though HP uses to announce checks there are two exceptions. Remember, too, that castling, advancing pawns 2 locations forward, and capturing "en passant" are not allowed.				
-do not turn off the machine while it is thinking its move, or otherwise running. You can turn it off whenever it is halted, then later resume with the game. If you turn off the machine while it runs, you'll generate errors due to some flags losing its status at turn on.				

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
	01*LBL "MCH			46	ISG L		
	ESS"			47	GTO 01		
	02 CLRG			48	ΣREG 74		
	03 FIX 0			49	CLΣ		
	04 CF 29			50	ASTO 79		
	05 18.049			51	FS? 55		
	06 RDTAX			52	XEQ "P"		
	07 SF 11			53	CF 23		
	08 OFF			54	RON		
	09 9			55	"HP 1ST?"		
	10 STO 16						
	11 ST- 17			56	PROMPT		
	12 6.5			57	AOFF		
	13 STO 92			58	FC?C 23		
	14 ST- 56			59	GTO 00		
	15 5.09			60*LBL 99			
	16 STO 93			61	"FROM?"		
	17 ST- 57			62	PROMPT		
	18 4.03			63	"I"		
	19 STO 94			64	X<0?		
	20 ST- 58			65	GTO 04		
	21 1.01			66	X=0?		
	22 STO 83			67	GTO 05		
	23 STO 84			68	XEQ 06		
	24 STO 85			69	STO 00		
	25 STO 86			70	"TO?"		
	26 STO 87			71	PROMPT		
	27 ST- 65			72	XEQ 06		
	28 ST- 66			73	STO 01		
	29 ST- 67			74	CLX		
	30 ST- 68			75	X<> IND		
	31 ST- 69			00			
	32 INT			76	STO IND		
	33 -			01			
	34 STO 95			77	XEQ 07		
	35 ST- 59			78	FC? 55		
	36 2.05			79	GTO 00		
	37 STO 96			80	FS? 00		
	38 ST- 60			81	XEQ "P"		
	39 "A"			82*LBL 00			
	40 50.091			83	"I MOVE"		
	41 SIGN			84	AVIEW		
	42*LBL 01			85	CF 17		
	43 RCL IND			86	FS? 55		
L				87	SF 17		
	44 X=0?			88	FS? 55		
	45 ASTO IND			89	XEQ "5"		
L				90	PI		

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
91	STO 09			135	BEEP		
92	96.055			136	AVIEW		
93	STO 02			137	FS? 55		
94	CHS			138	XEQ "P"		
95	STO 00			139	FC? 55		
96	LBL 11			140	STOP		
97	RCL IND			141	GTO 99		
02				142	LBL 07		
98	SIGN			143	60		
99	X=0?			144	RCL 01		
100	GTO 00			145	X>Y?		
101	LASTX			146	RTN		
102	X<0?			147	2		
103	XEQ 12			148	RCL IND		
104	LBL 00			01			
105	DSE 02			149	X>Y?		
106	GTO 11			150	RTN		
107	FS? 17			151	"PIECE?"		
108	XEQ "5"			152	PROMPT		
109	RCL 09			153	STO IND		
110	PI			01			
111	X=Y?			154	RTN		
112	GTO 05			155	LBL 08		
113	"YOU"			156	ABS		
114	-25			157	2		
115	RCL 00			158	X<Y?		
116	X<Y?			159	RTN		
117	GTO 04			160	92		
118	CLX			161	RCL 13		
119	X<> IND			162	X<Y?		
12				163	RTN		
120	STO IND			164	-5.09		
13				165	STO IND		
121	XEQ 08			13			
122	"FROM "			166	RTN		
123	RCL 12			167	LBL 04		
124	XEQ 09			168	ASTO X		
125	"F TO "			169	"CHECKMA		
126	RCL 13			TE"			
127	XEQ 09			170	AVIEW		
128	RCL 00			171	BEEP		
129	FRC			172	CLA		
130	RCL 22			173	ARCL X		
131	*			174	"F WON"		
132	FRC			175	PROMPT		
133	X=0?			176	LBL 05		
134	"F, CHEC			177	"STALEMA		
K"				TE"			

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
178	BEEP			224	LBL 15		
179	PROMPT			225	RCL IND		
180	LBL 09			04			
181	INT			226	ST+ 05		
182	ENTER↑			227	RCL 05		
183	ENTER↑			228	XEQ 08		
184	9			229	FS? 18		
185	/			230	GTO 00		
186	INT			231	X<0?		
187	+			232	GTO 00		
188	51			233	CF 09		
189	-			234	X=0?		
190	ARCL X			235	SF 09		
191	RTN			236	XEQ 12		
192	LBL 06			237	FS? 05		
193	ENTER↑			238	GTO 00		
194	ENTER↑			239	FS? 09		
195	1			240	GTO 15		
196	-			241	LBL 00		
197	5			242	ISG 04		
198	/			243	GTO 14		
199	INT			244	RTN		
200	2			245	LBL 36		
201	/			246	SF 07		
202	-			247	LBL 33		
203	46			248	SF 05		
204	+			249	LBL 32		
205	RTN			250	LBL 34		
206	LBL 12			251	LBL 35		
207	STO 03			252	RTN		
208	ABS			253	LBL 13		
209	CF 05			254	SF 06		
210	CF 06			255	RCL 02		
211	CF 07			256	9		
212	2			257	XEQ 09		
213	X>Y?			258	FS? 18		
214	GTO 13			259	1		
215	X<>Y			260	X=0?		
216	30			261	XEQ 12		
217	+			262	RCL 02		
218	XEQ IND			263	RCL 22		
X				264	XEQ 09		
219	RCL IND			265	FS? 18		
X				266	CLX		
220	STO 04			267	X>0?		
221	LBL 14			268	XEQ 12		
222	RCL 02			269	RCL 02		
223	STO 05			270	8		

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
271	XEQ	09		316	GTO	04	
272	FS?	18		317	LBL	00	
273	RTN			318	ISG	11	
274	X<=0?			319	GTO	21	
275	RTN			320	RCL	09	
276	LBL	12		321	STO	00	
277	CF	08		322	RCL	02	
278	STO	07		323	STO	12	
279	FRC			324	RCL	08	
280	1 E2			325	STO	13	
281	*			326	LBL	04	
282	STO	06		327	RCL	03	
283	RCL	Z		328	STO	IND	
284	STO	08		02			
285	.4			329	RCL	07	
286	FS?	07		330	STO	IND	
287	ST-	06		08			
288	FS?	06		331	RTN		
289	XEQ	12		332	LBL	12	
290	FC?	07		333	.5		
291	XEQ	13		334	ST+	06	
292	RCL	00		335	92		
293	RCL	06		336	RCL	08	
294	X<=Y?			337	X<Y?		
295	RTN			338	RTN		
296	RCL	03		339	SF	08	
297	FS?	08		340	9		
298	-5.09			341	ST+	06	
299	STO	IND		342	RTN		
08				343	LBL	13	
300	CLX			344	FS?	06	
301	STO	IND		345	GTO	13	
02				346	RCL	03	
302	56.096			347	30		
303	STO	11		348	-		
304	STO	09		349	RCL	IND	
305	CF	19		X			
306	LBL	21		350	STO	01	
307	RCL	IND		351	LBL	03	
11				352	RCL	08	
308	SIGN			353	STO	10	
309	X=0?			354	LBL	10	
310	GTO	00		355	RCL	IND	
311	LASTX			01			
312	X<=0?			356	ST+	10	
313	GTO	00		357	RCL	10	
314	XEQ	07		358	XEQ	08	
315	FS?	19		359	FS?	18	

☐ 67 ☐ 97 ☒ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
360	GTO	00		406	RCL	11	
361	XEQ	12		407	STO	15	
362	X=Y?			408	LBL	29	
363	RTN			409	RCL	IND	
364	FS?	05		14			
365	GTO	00		410	ST+	15	
366	LASTX			411	RCL	15	
367	X=0?			412	XEQ	08	
368	GTO	10		413	FS?	18	
369	LBL	00		414	GTO	00	
370	ISG	01		415	X>0?		
371	GTO	03		416	GTO	00	
372	RTN			417	CF	10	
373	LBL	13		418	X=0?		
374	RCL	08		419	SF	10	
375	RCL	22		420	XEQ	13	
376	XEQ	00		421	FS?	19	
377	RCL	08		422	RTN		
378	0			423	FS?	01	
379	LBL	00		424	GTO	00	
380	XEQ	09		425	FS?	10	
381	FS?	18		426	GTO	29	
382	RTN			427	LBL	00	
383	LBL	12		428	ISG	14	
384	INT			429	GTO	28	
385	6			430	RTN		
386	X≠Y?			431	LBL	36	
387	RTN			432	SF	03	
388	.41			433	LBL	33	
389	ST+	06		434	SF	01	
390	RDN			435	LBL	32	
391	RTN			436	LBL	34	
392	LBL	07		437	LBL	35	
393	CF	01		438	RTN		
394	CF	02		439	LBL	12	
395	CF	03		440	SF	02	
396	2			441	RCL	11	
397	X>Y?			442	RCL	17	
398	GTO	12		443	XEQ	09	
399	X<>Y			444	FS?	18	
400	30			445	1		
401	+			446	X=0?		
402	XEQ	IND		447	XEQ	13	
X				448	FS?	19	
403	RCL	IND		449	RTN		
X				450	RCL	11	
404	STO	14		451	RCL	20	
405	LBL	28		452	XEQ	00	

LINE KEY ENTRY

(CONTINUATION PAGE) KEY ENTRY

453 FS? 19
454 RTN
455 RCL 11
456 RCL 21
457*LBL 00
458 XEQ 09
459 FS? 18
460 RTN
461 X=0?
462 X>0?
463 RTN
464*LBL 13
465 FRC
466 ABS
467 1 E2
468 *
469 FS? 03
470 .4
471 FS? 03
472 -
473 FS? 02
474 XEQ 13
475 RCL 06
476 X<>Y
477 -
478 RCL 00
479 X<>Y
480 X<=Y?
481 SF 19
482 X<=Y?
483 RTN
484 RCL 09
485 X<>Y
486 X<Y?
487 STO 09
488 RTN
489*LBL 13
490 .5
491 +
492 RCL Z
493 60
494 X<>Y
495 CF 04
496 X<=Y?
497 SF 04
498 RCL Z
499 9
500 FC? 04

501 CLX
502 +
503 RTN
504*LBL 09
505 +
506*LBL 08
507 CF 18
508 SF 25
509 RCL IND
X
510 SIGN
511 FS?C 25
512 X=0?
513 SF 18
514 LASTX
515 END

LBL*MCHESS

END 921 BYTES

LINE KEY ENTRY

(CONTINUATION PAGE) KEY ENTRY

```
01*LBL "P"
02 ADV
03 SF 12
04 9
05 SKPCOL
06 49.053
07 STO 13
08*LBL 00
09 ACCHR
10 2
11 SKPCOL
12 X<>Y
13 ISG X
14 GTO 00
15 PRBUF
16 56.06
17 STO 15
18*LBL 01
19 RCL 13
20 ACCHR
21*LBL 02
22 2
23 SKPCOL
24 RCL IND
15
25 INT
26 43
27 +
28 RCL IND
X
29 ACSPEC
30 ISG 15
31 GTO 02
32 4.009
33 ST+ 15
34 ISG 13
35 GTO 01
36 ADV
37 ADV
38 ADV
39 CF 12
40*LBL "5"
41 END
```

LBL'P

LBL'5

END

83 BYTES

```
Print board
routine
```

REGISTERS, STATUS, FLAGS, ASSIGNMENTS

DATA REGISTERS				STATUS							
<div>00 } ... } scratch 15 } 16 +9 } 17 -9 } 18 +1 } rock king & 19 -1 } queen 20 -10 } 21 -8 } 22 10 } bishop 23 8 } 24 -19 } 25 -17 } 26 -7 } 27 -11 } knight 28 19 } 29 17 } 30 7 } 31 11 } 32 16.019 rock 33 24.031 knight 34 20.023 bishop 35 16.023 queen 36 16.023 king 37 black king 38 id.queen 39 id.bishop 40 id.knight 41 id.rook 42 id.pawn 43 dot.sq. 44 white pawn 45 id.rook 46 id.knight 47 id.bishop 48 id.queen 49 id.king 37 } ... } board (includes 96 } edges) note: board and BLDSPEC chars. overlap without trouble</div>				SIZE <u>097</u>		TOT. REG. <u>241</u>		USER MODE			
				ENG		FIX <u>0</u>		SCI		ON OFF	
				DEG		RAD		GRAD			
				FLAGS							
				#	INIT S/C	SET INDICATES	CLEAR INDICATES				
				00		board always print.	prints after HP's				
				01		white king or knight tested/Q,R,B					
				02		id. pawn moves w.pawn not moving					
				03		id. king moves w.king not moving					
				04		id.pawn promotes w.pawn not prom.					
				05		b.king or knight tested /Q,R,B					
				06		b.pawn moves	b.pawn not moving				
				07		b.king moves	b.king not moving				
				08		b.pawn promotes	b.pawn not prom.				
				09		b.Q,R,B can follow	they can't				
				10		w.Q,R,B can follow	they can't				
				11		automatic start					
				12		double wide print	normal width				
				25		non-edge location	edge location				
				55		print board	do not print board				
				17		used					
				18		used					
				19		used					
				ASSIGNMENTS							
				FUNCTION		KEY					
				FUNCTION		KEY					

PROGRAM REGISTERS NEEDED: 132

ROW 1 (1 : 3)



ROW 2 (4 : 8)



ROW 3 (9 : 15)



ROW 4 (15 : 19)



ROW 5 (20 : 25)



ROW 6 (25 : 31)



ROW 7 (32 : 38)



ROW 8 (38 : 43)



ROW 9 (44 : 51)



ROW 10 (51 : 55)



ROW 11 (55 : 61)



ROW 12 (61 : 68)



ROW 13 (68 : 74)



ROW 14 (75 : 80)



ROW 15 (81 : 85)



ROW 16 (85 : 92)



ROW 17 (92 : 99)



ROW 18 (100 : 107)



ROW 19 (107 : 113)



ROW 20 (114 : 121)



ROW 21 (121 : 125)



ROW 22 (125 : 130)



ROW 23 (131 : 135)



ROW 24 (136 : 142)



ROW 25 (143 : 151)



ROW 26 (151 : 159)



ROW 27 (160 : 166)



ROW 28 (167 : 169)



ROW 29 (170 : 176)



ROW 30 (177 : 180)



ROW 31 (181 : 191)



ROW 32 (192 : 203)



ROW 33 (204 : 213)



ROW 34 (214 : 222)



ROW 35 (223 : 229)



ROW 36 (230 : 236)



ROW 37 (237 : 243)



ROW 38 (243 : 250)



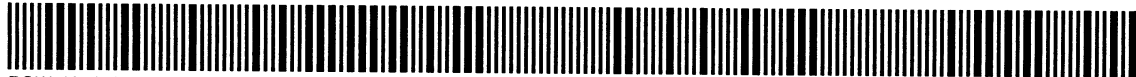
ROW 39 (250 : 258)



ROW 40 (258 : 265)



ROW 41 (265 : 272)



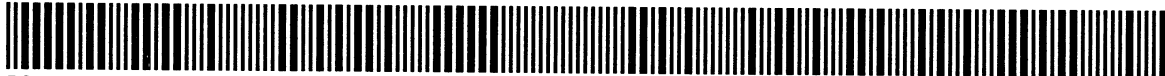
ROW 42 (273 : 282)



ROW 43 (283 : 289)



ROW 44 (289 : 297)



ROW 45 (298 : 302)



ROW 46 (302 : 309)



ROW 47 (310 : 316)



ROW 48 (317 : 326)



ROW 49 (327 : 335)



ROW 50 (335 : 344)



ROW 51 (345 : 354)



ROW 52 (355 : 361)



ROW 53 (361 : 369)



ROW 54 (370 : 377)



ROW 55 (378 : 387)



ROW 56 (388 : 395)



ROW 57 (395 : 403)



ROW 58 (404 : 412)



ROW 59 (412 : 419)



ROW 60 (419 : 425)



ROW 61 (426 : 432)



ROW 62 (432 : 439)



ROW 63 (440 : 447)



ROW 64 (447 : 453)



ROW 65 (454 : 462)



ROW 66 (463 : 471)



ROW 67 (471 : 480)



ROW 68 (481 : 491)



ROW 69 (492 : 499)



ROW 70 (500 : 509)



ROW 71 (509 : 515)

