COP-4338 System Programming

Programming Assignment 3:

FIU School of Computing & Info. Sciences

In this assignment, you are asked to write a program that stores a chessboard in a 2D array of integers. Your program must make necessary changes on the array after each move or capture of the two players of the game.

1 Array Format

A chessboard can be represented by an 8 × 8 table of integers in which every cell is corresponding to one of the 64 squares of the board and determines the occupant of the square. As seen in Figure 1, the table row with index 0 is labeled with "8" and the table row with index 7 is labeled with "1". Also, the table column with index 0 is labeled with "a", while the table column with index 7 is labeled with "h".

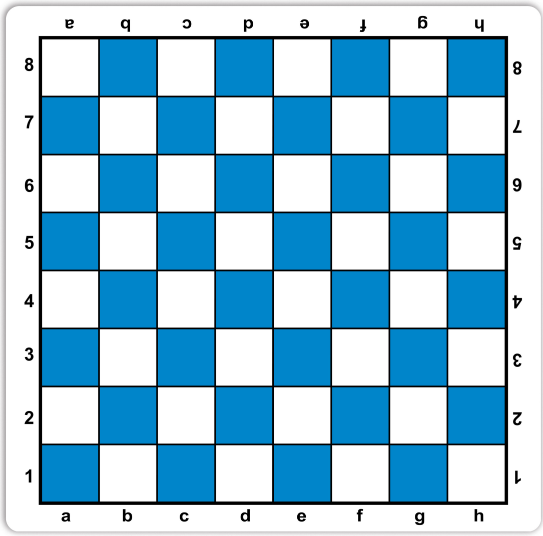


Figure 1: Row and column labels in a chessboard.

If the value of a table cell is zero, it means that the square corresponding to that cell is empty. If the value is positive, it means that a white piece occupies the corresponding square; while a negative value means that the occupying piece color is black. The absolute value of the cell specifies type of piece occupying the corresponding square. Here is the general rule:

$$|table[i][j]| = \begin{cases} 0 & \text{cell is empty} \\ 1 & \text{cell is occupied by a pawn} \\ 2 & \text{cell is occupied by a knight} \\ 3 & \text{cell is occupied by a bishop} \\ 4 & \text{cell is occupied by a rook} \\ 5 & \text{cell is occupied by a queen} \\ 6 & \text{cell is occupied by a king} \end{cases} \tag{1}$$

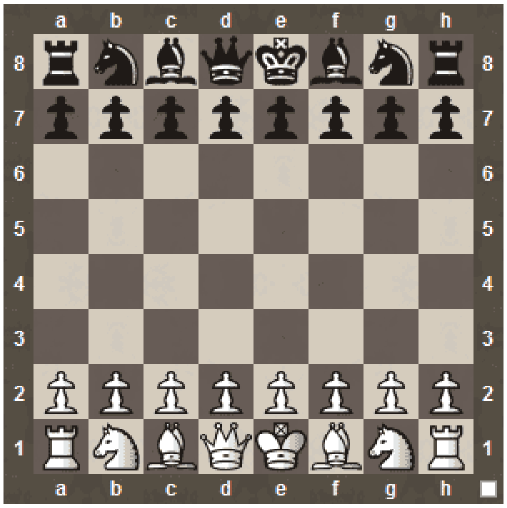
For the sake of simplicity, you can store this table in a static or external 2D array of integers in your program.

2 Initial Setting of Pieces on the Board

The program starts by initializing the table in the following fashion: which corresponds to

-4	-2	-3	-5	-6	-3	-2	-4
-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
4	2	3	5	6	3	2	4

the following chessboard setup:



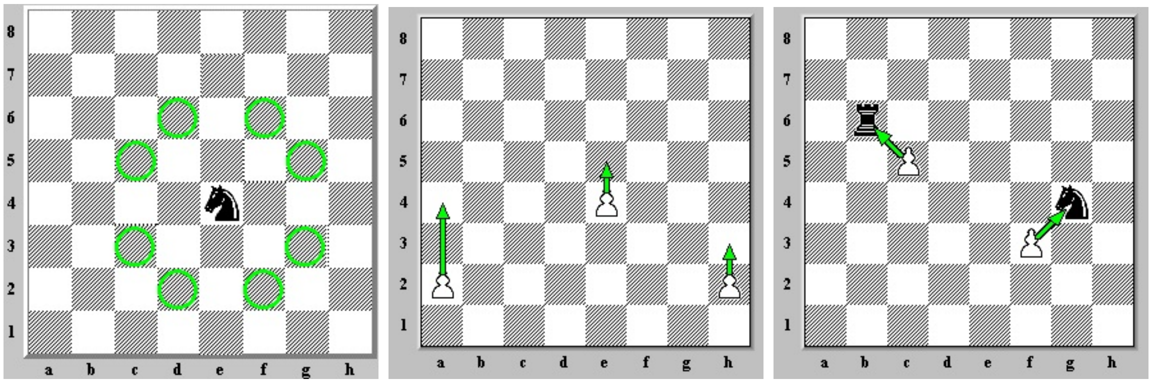


Figure 2: Movement and capture of knight (left), movement (middle) and capture of pawn (right) on chessboard.

3 Program Commands

Your program must perform the following commands entered by user via standard input stream:

- mv $\alpha_0\beta_0$ $\alpha_1\beta_1$: moves the piece in cell with label $\alpha_0\beta_0$ to cell with label $\alpha_1\beta_1$ where α_0 and α_1 can be letters from a to h; while β_0 and β_1 can be integers from 1 to 8.
- cp $\alpha_0\beta_0$ $\alpha_1\beta_1$: captures the piece in cell with label $\alpha_1\beta_1$ with the piece in cell $\alpha_0\beta_0$ where α_0 and α_1 can be letters from a to h; while β_0 and β_1 can be integers from 1 to 8.
- show: prints out the content of the 2D array on the screen in the form of a table like this (the ranks 1,2,..., 8 and files a, b, ..., h must be printed as well):

board	a	b	c	d	e	f	g	h
8	-4	-2	-3	-5	-6	-3	-2	-4
7	-1	-1	-1	-1	0	-1	-1	-1
6	0	0	0	0	0	0	0	0
5	0	0	0	0	-1	0	0	0
4	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0
2	1	1	1	1	0	1	1	1
1	4	2	3	5	6	3	2	4

Figure 2 shows how knight can move on the board and capture pieces with opposite color on the board. Also, it shows the direction in which white pawn moves and captures. The black pawn moves and captures in the opposite direction. In summary, a pawn can move only one cell forward at a time; however, if it is white and is in the second rank (or is black and is in the seventh rank), it has the option of moving two cells forward as well if there is no piece on its way. Bishop moves diagonally and rook moves vertically or horizontally. Queen can move like a bishop or rook. King moves one square at a time to one of the four adjacent squares.

If a ‘cp’ or ‘mv’ command tries to make illegal move, your program prints out an error message.

4 30% Bonus Parts

1. As the first bonus point, your program must support promotion of white/black pawn to a queen, knight, rook or bishop once it reaches to the eighth/first rank.

2. As the second bonus part, your program must support en passant move (see https://en.wikipedia.org/wiki/En_passant for details)

3. As the third bonus part, your program must print a “check” message once the king is in check.

5 Submissions

You need to submit a *.zip* file compressing the following files:

- the C source file(s) related to the assignment (*.c* files).

- the header files (.h files)

- **A readme file clearly explaining what parts have/haven’t been implemented.**