Set of Descriptions and features:

At first of our report , we should show a description of our application . our application is for a chess game , usage of our application is completely out of complexity for users .
as soon as you open the application , you enter your name , and the name of the other player .. then immediately the game begins..

player name appears below of the board to show the player turn , and player one is who begin the game with the small pieces , then player two with the capital pieces ..

The pieces of the first player in down and the pieces of the other is up ..

The movements the players enter is on the same line of the player name ..

Every player should enter a movement consists of 4 letters or digits , the first 2 is the position of the piece the player wants to move , the second 2 is the position the player wants to move the specified piece to ..
the first of 2 represents the column from A to H , and the second of 2 represents the row from 1 to 8 .

If there is an incompatible movement with respect to the specified piece , the same player reenter the movement , and so on until a correct movement to be implemented then and take turn to the other player …

If there are dead pieces , it appears on right of the board with the player name these pieces belongs to ..

If there is any action like "check , checkmate , stalemate ,promotion .." , there is a message appears to the player below of the player turn name ..

Promotion need only a choice of the illustrated pieces and the player enter it individually after making the movements ..

Check forces you to enter the required movement to cancel this threat ..

Provided stalemate is for surrounding the king ..

Checkmate end the game with a message of the winner player ..

Additional Features:

There's no big additional features , we seek only to improve the efficiency as we can..

for example :saving and loading at the quickest possible way , using data file not a text , includes the data after making analysis for it and being in the final form…….

Redo , and undo can be used professionally as any other game , that’s because you can make redo and undo at any time you load your game , so making redo and undo of the saved game is included ..

Overview of the design:

Our design consist of a 32 functions excluding the main one ..

There are a while loop in which all the game inside it , and the game ends when a the flag of this loop take a specified value of the checkmate function..

Inside this loop there are two other loops each one for each player ..

Every while loop of them have a flag take a value to end the player turn after achieving sets of functions and validations…

Every turn there's a check for the entry of "Redo , Undo , Save , Load " and use their function if true ..

Every movement the player enter , have been analyzed to the opponent number of table of the game..

If the output of analyzing is compatible to the validation terms .. we turn to achieve this movement ..

After achieving every movement we save the analyzed entry in a list of two dimension .

We save the dead pieces in a list in order as they died .. this list is connected with another list of last position of the dead piece ..

When the player makes undo .. there are two pointers one go back for a one step for every entry of "undo" .. when "redo" have been entered the pointer take a step forward ..there's no effect when the first pointer reach the second or the position "0".. when there's a new movement after redoing the second pointer move to the position of the first and the rest of list have been clarified..

When the player make "save" the lists saved in data file .. "load" make reading of these files ..

The ordinary entries of moving pieces have been checked , and depending on the position of the piece , the suitable function be used for example "pawn , knight , queen ……." , if the function returned true we turn to the "Makingmoving" function … and so on ….

And finally , after every making moving for the pawn we check for promotion and achieve it using "pawnpromotion"…

Function (pseudocode & flowcharts) and describing if the important functions in commenct in pseudocodes:

1)drawboard function:

؛draw board

؛define i,j,c1,c2

}(while(loop j for the board size for position 0 & 9 for row & column

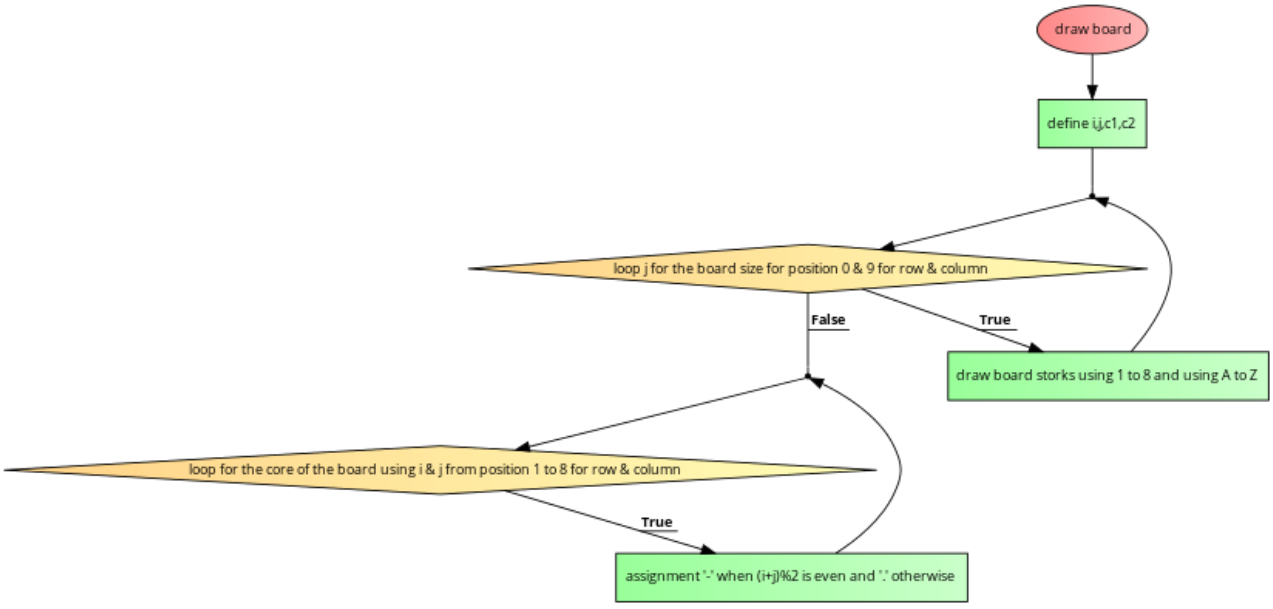
{؛draw board storks using 1 to 8 and using A to Z

؛reinitialize j

while(loop for the core of the board using i & j from position 1 to 8 for row & (column

؛assignment '-' when (i+j)%2 is even and '.' otherwise

return;



2)pieces function:

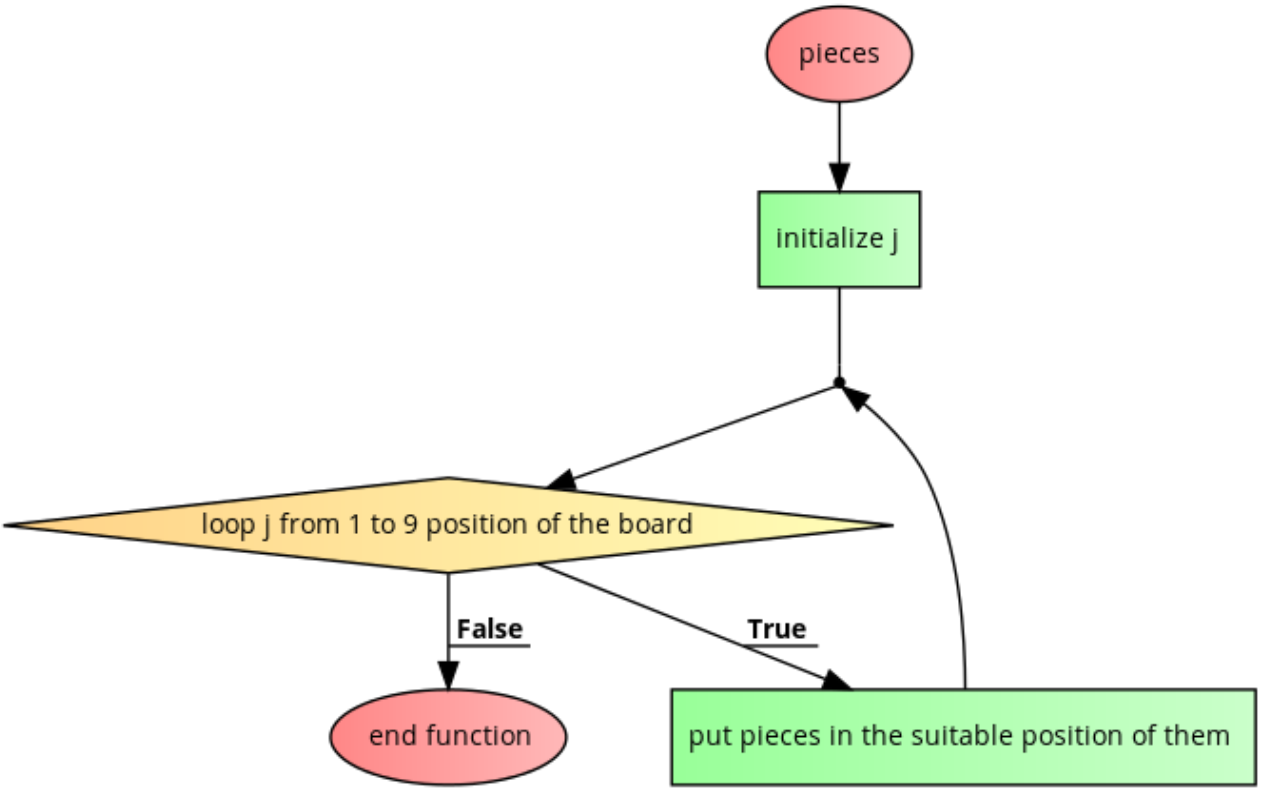
؛pieces

؛initialize j

(while (loop j from 1 to 9 position of the board

؛ put pieces in the suitable position of them

؛end function



3)analyzing moving function:

؛analizing moving

؛initialize i

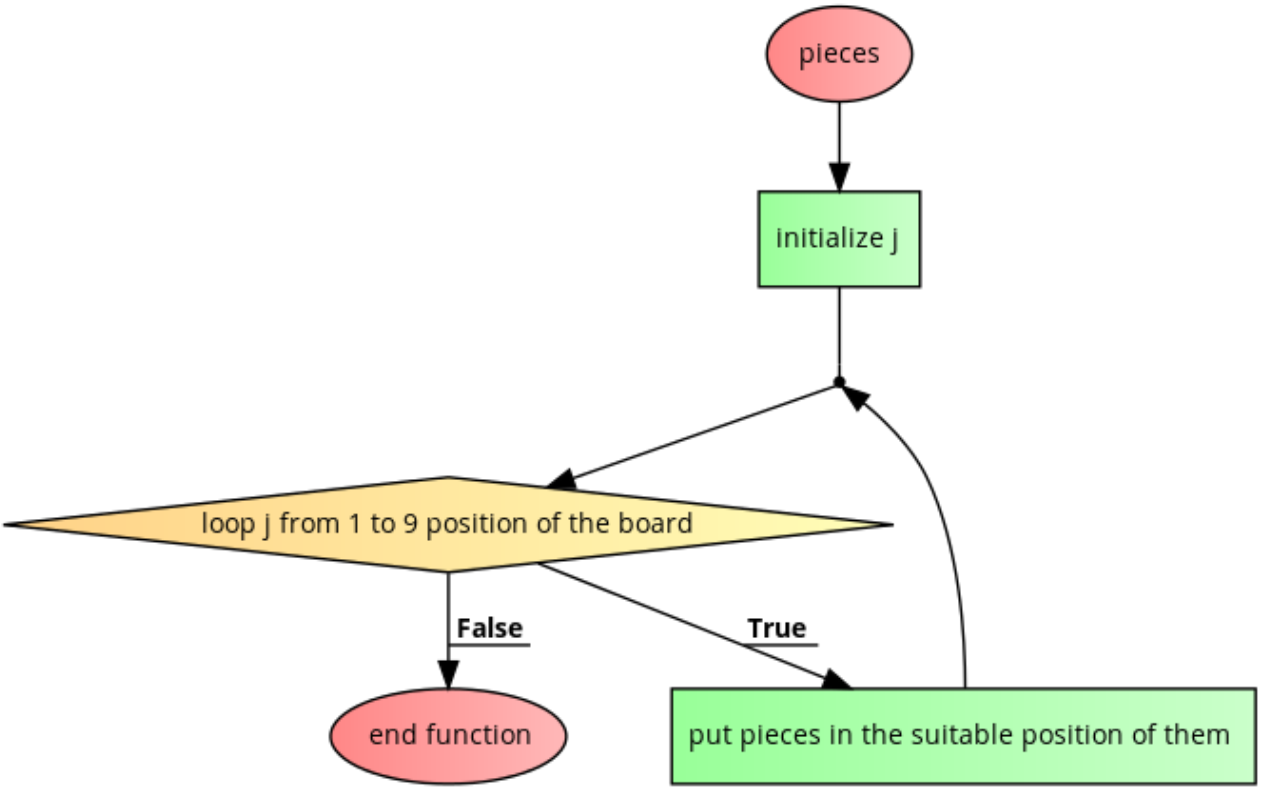
}(while(loop four times

؛take every input char and determine it's position in the board by the storks

؛in one of 4 pointer assignment the number

{

End function;



4)making moving function:

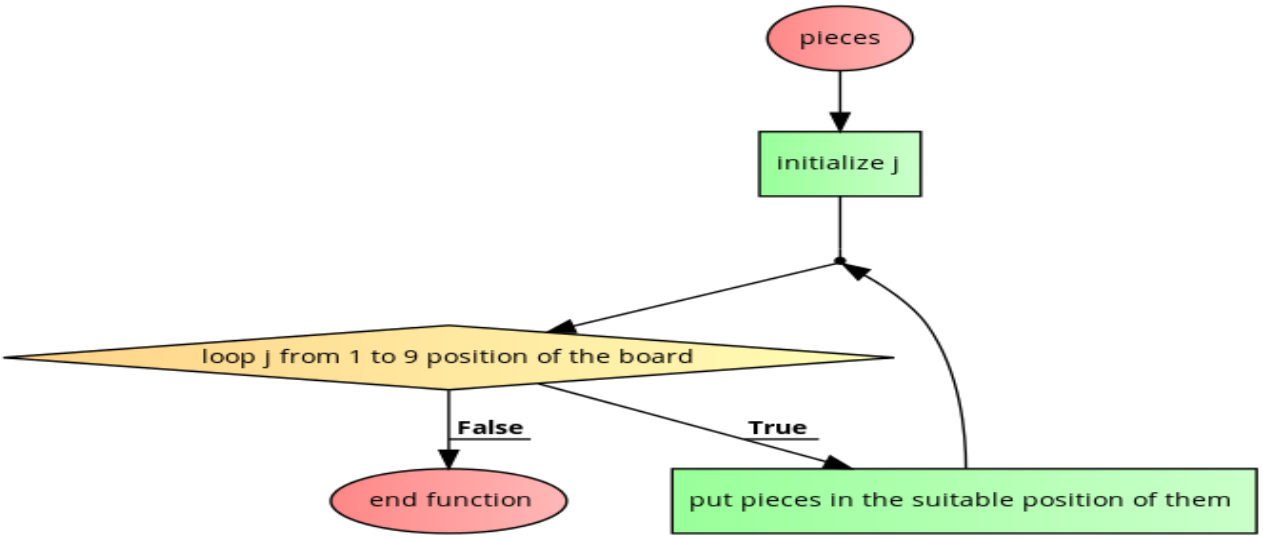
making moving;

import the 4 pointers of "analyzing moving function";

assigne position 3,4 of board with 1,2

assign position 1,2 with '-' or '.';

end function;



5)undo function:

؛ undo

، import board, lists of correctmoves

؛losesposition and the two pointers of undo

؛decrement pointer1

using the 4 numbers stored at pointer1 in the list correctmoves//

؛||making moving||

if(there's compatability between numbers

(in correctmoves and lossesposition

assigne the last piece in lostpieces

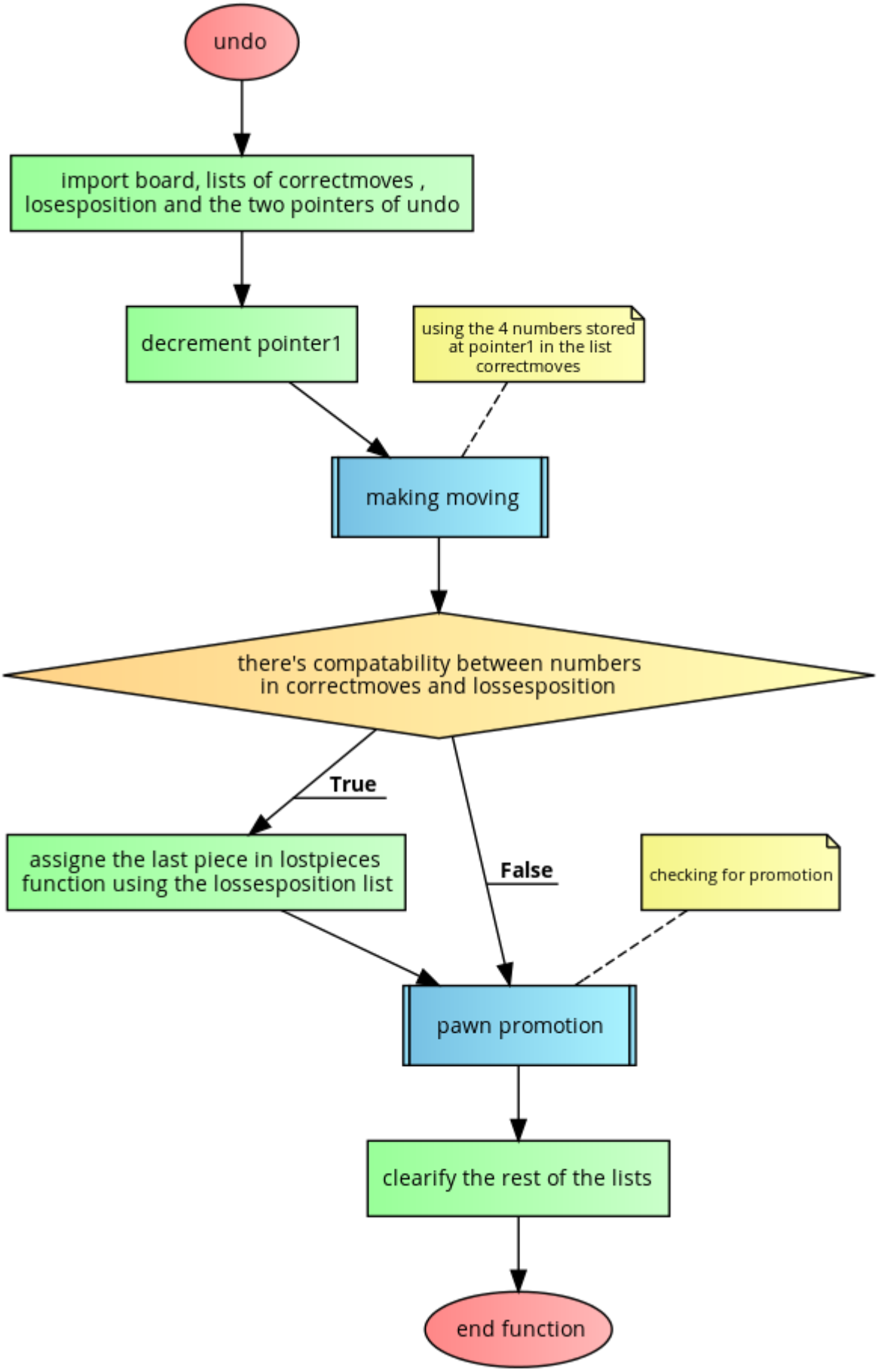
؛function using the lossesposition list

checking for promotion//

؛||pawn promotion||

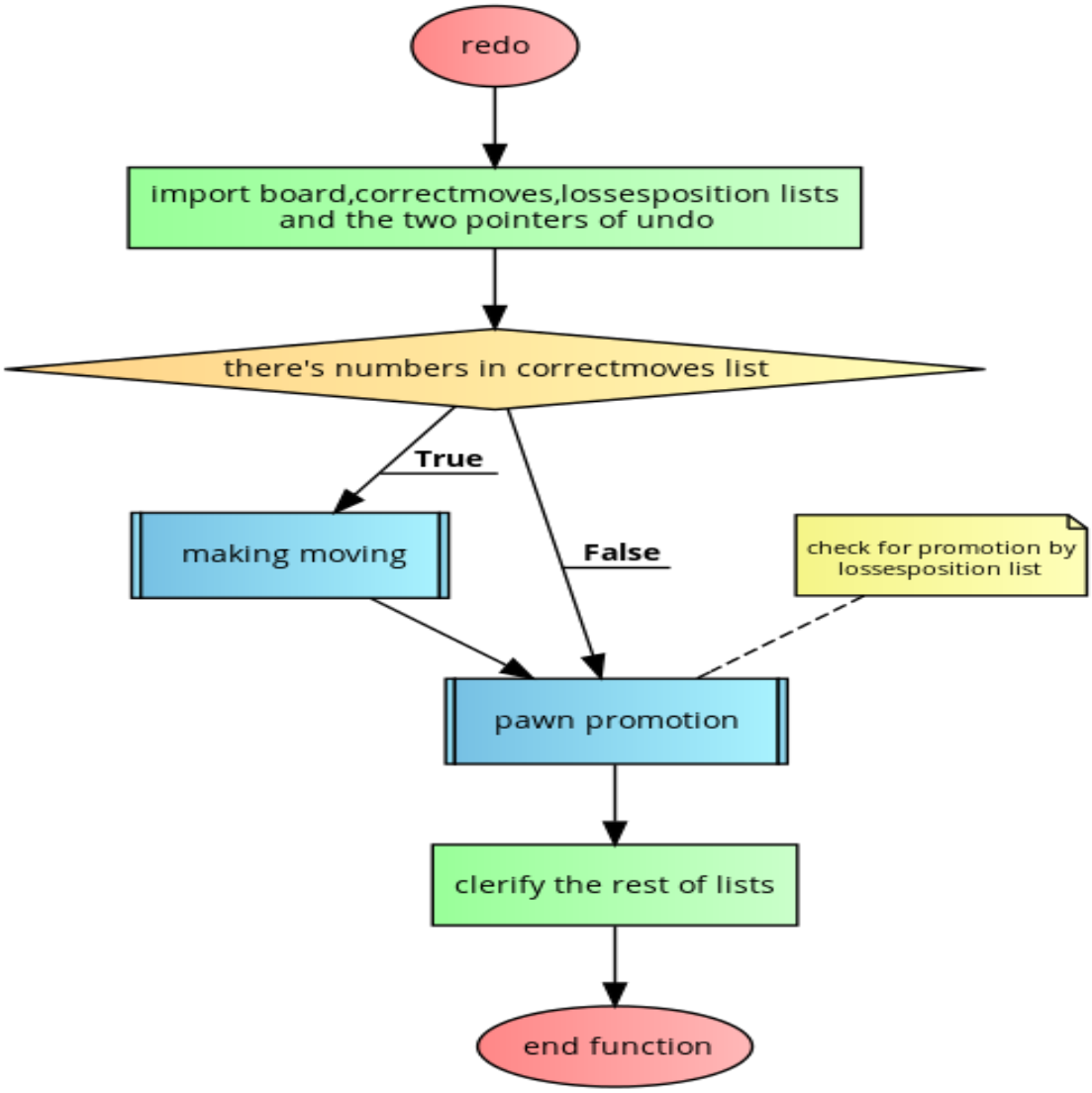
؛clearify the rest of the lists

end function;


undo
import board, lists of correctmoves , losesposition and the two pointers of undo
decrement pointer1
using the 4 numbers stored at pointer1 in the list correctmoves
making moving
there's compatability between numbers in correctmoves and lossesposition
True
False
assigne the last piece in lostpieces function using the lossesposition list
checking for promotion
pawn promotion
clearify the rest of the lists
end function

6)redo function:

```
redo؛
import board,correctmoves,lossesposition lists
and the two pointers of undo؛
(if (there's numbers in correctmoves list
||making moving||؛
check for promotion by lossesposition list//
||pawn promotion||؛
clerify the rest of lists؛
end function;
```



7) **important note:** logical functions is the function which check the possibility of moving of every pieces , they are just logical relations , very simple … so they needn't any illustration .. they very clear in the source code , we won't considered it important algorithms as required…

8)check function:

//check function is to check if the king threatened or not

import board;

//using all logical pieces functions we determine if the king is threatend or not

while (making loop throughout all position of the board tied it with the position

of the king of this player)

check for availability moving

pieces to the king position;

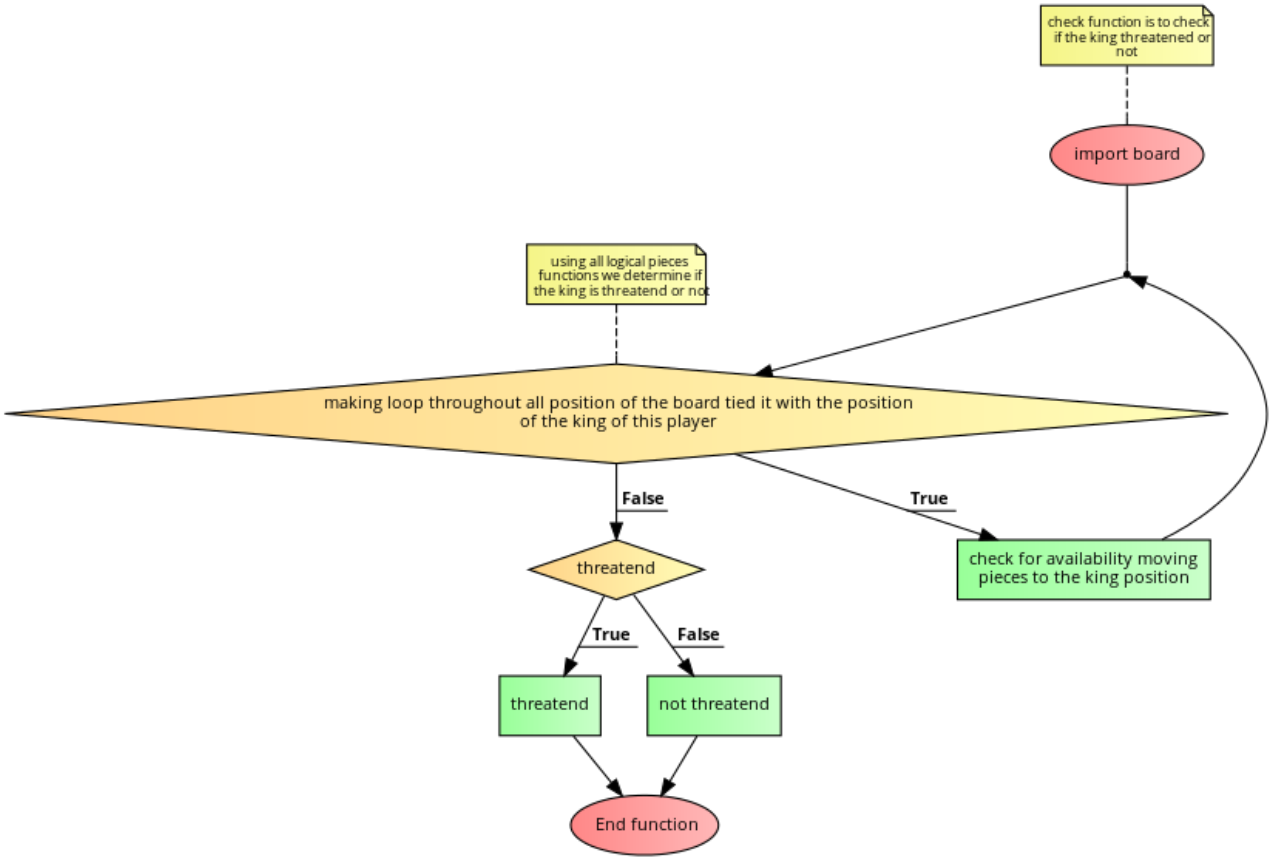
if (threatend){

threatend}

else

not threatend;

End function;



9)remove check function:

؛removecheck

(while (search looping for the position of the king

store the position in k1,k2؛

}(while(make a loop around the king's position

check for avialbility by this function to return not threatend if the king moved to an // around square

||check||؛

}(if (not threatened

{flag=not threatend؛

{

}(while(make a loop for the king's pieces

}(while(make a loop on all the board

check every time if ckeck function return not thretened after a virtural moving//

||check||؛

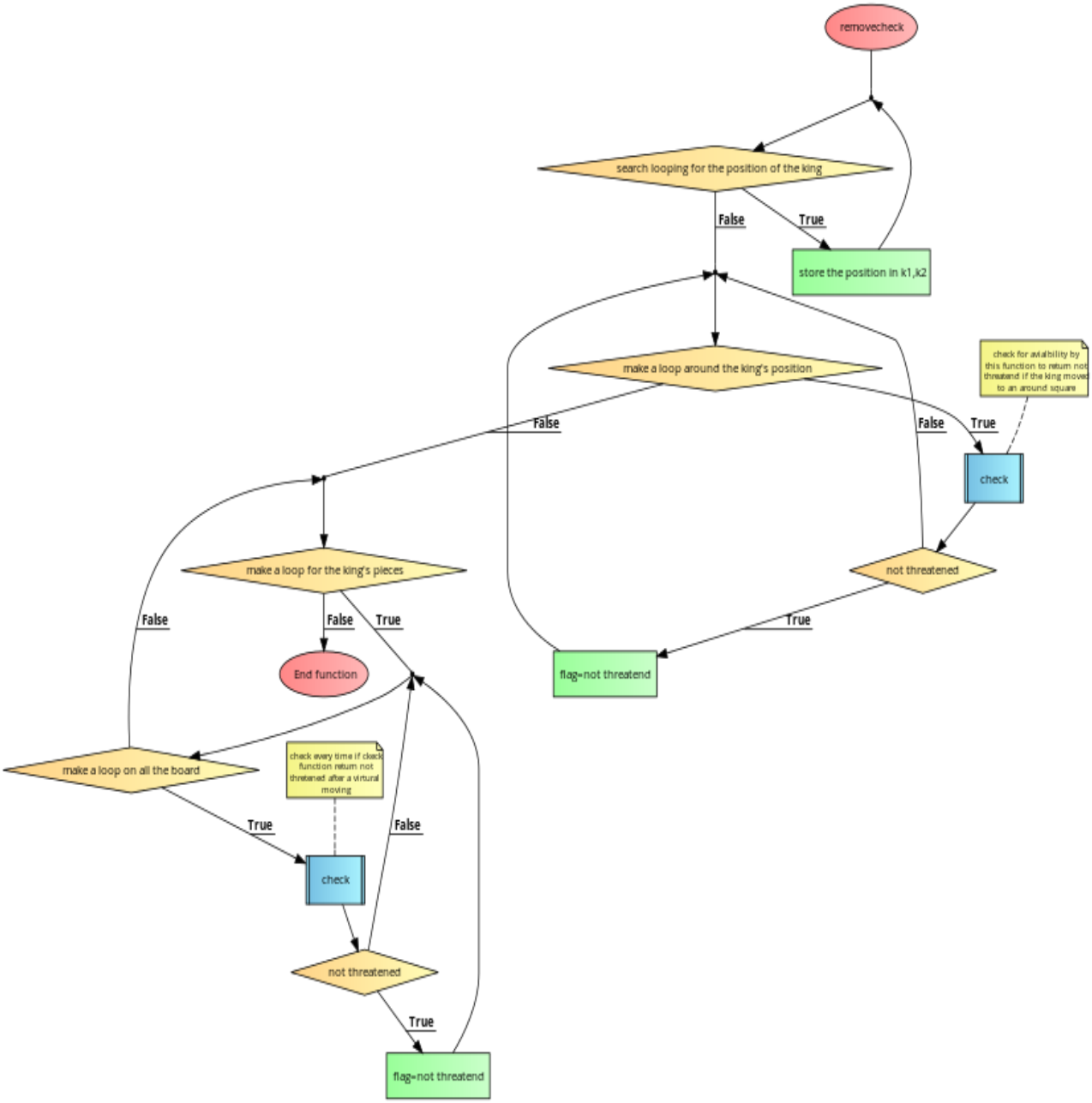
}(if (not threatened

{flag=not threatend؛

{

{

End function;



10)checkmate function:

checkmate;

//check if the king threatened and store the returned value in a flag

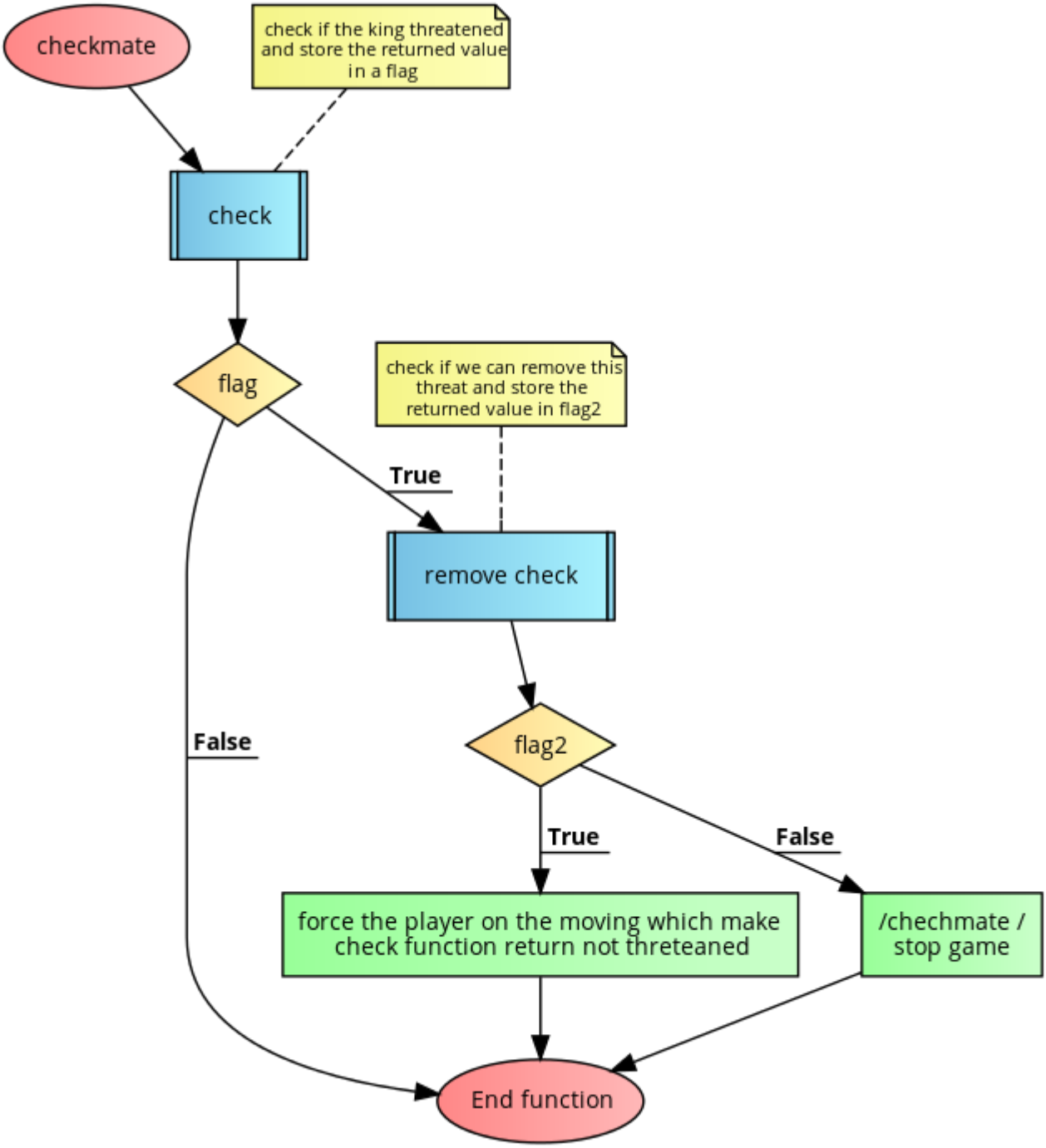
||check||;

if (flag){//check if we can remove this threat and store the returned value in flag2

||remove check||;

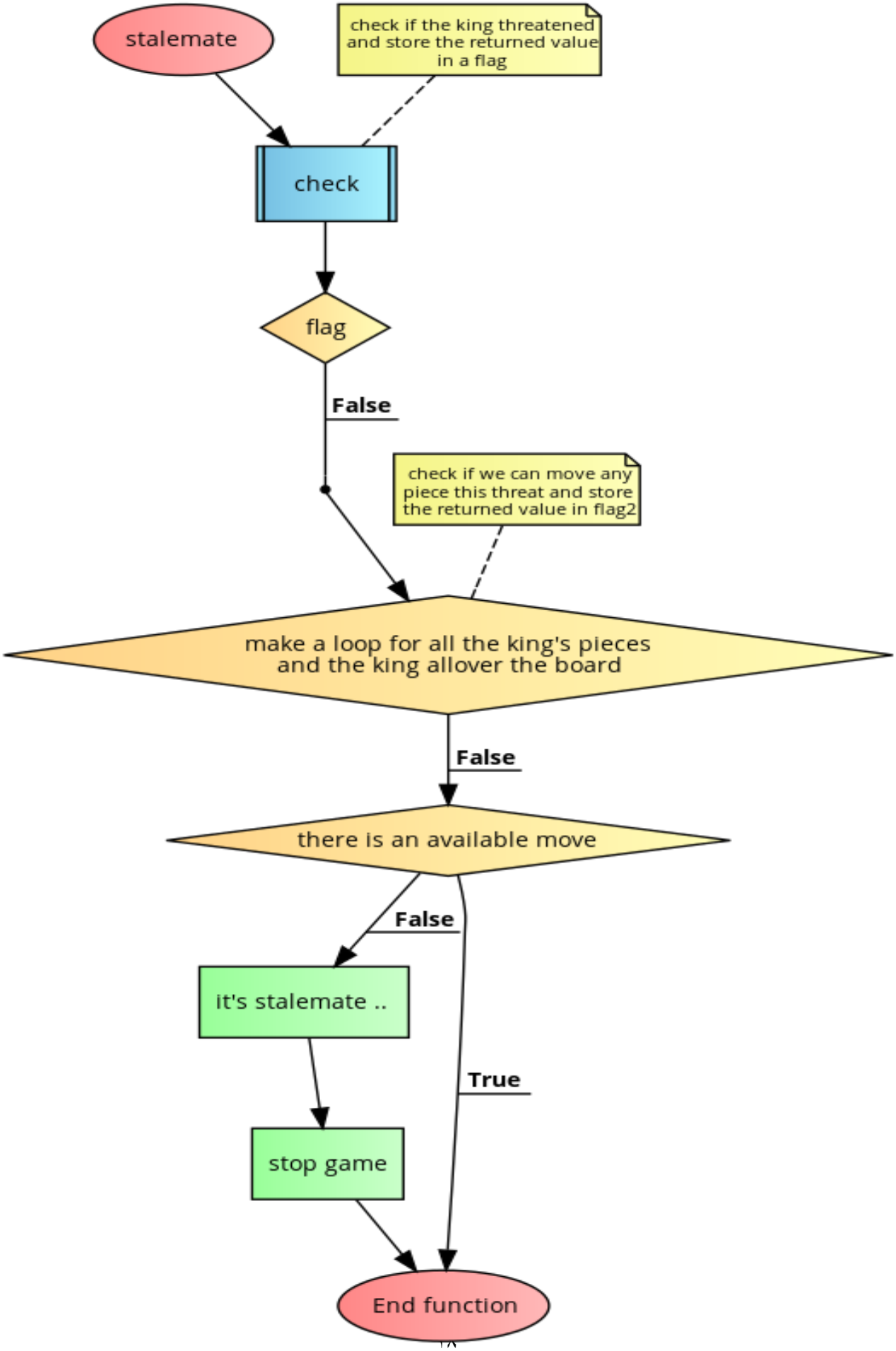
if (!flag2){

```
 /chechmate /
 stop game;
}
else
force the player on the moving which make
check function return not threteaned;
}
End function;
```


checkmate
check if the king threatened and store the returned value in a flag
check
flag
check if we can remove this threat and store the returned value in flag2
True
remove check
False
flag2
True
False
force the player on the moving which make check function return not threteaned
/chechmate / stop game
End function

11)stalemate function:

```
؛stalemate

check if the king threatened and store the returned value in a flag//

؛||check||

if (!flag){//check if we can move any piece this threat and store the returned value in
                                                                                flag2

while (make a loop for all the king's pieces

}(and the king allover the board

{

}(if (!there is an available move

؛.. it's stalemate

؛stop game

{

؛End function
```


stalemate
check if the king threatened and store the returned value in a flag
check
flag
False
check if we can move any piece this threat and store the returned value in flag2
make a loop for all the king's pieces and the king allover the board
False
there is an available move
False
it's stalemate ..
True
stop game
End function

12)main function:

```
؛main function
؛flag=1
loop of the game have a flag//
}(while(flag!=0
first player loop with a flag1//
؛flag1=1,flag2=1
}(while(flag1&&flag
check for checkmate//
؛||checkmate||
check for stalemate//
؛||stalemate||
؛take the moving form player
؛||analyzing moving||
}(if (output of analayzing is suitable
(! if (there's checkmate or stalemate
{؛End the game, flag=0}
(else if (there's a check
{؛force the player on the appropriate moving}
{{؛making moving||; flag1=0||}
؛change the important pointer's numbers and lists
check for promotion//
؛||pawn promotion||
{
```

؛flag1=1

second player loop with flag2//

}(while(flag2&&flag

check for checkmate//

؛||checkmate||

check for stalemate//

؛||stalemate||

؛take the moving form player

؛||analyzing moving||

}(if (output of analayzing is suitable

(! if (there's checkmate or stalemate

{؛End the game, flag=0}

(else if (there's a check

{؛force the player on the appropriate moving}

{{؛making moving||; flag2=0||}

؛change the important pointer's numbers and lists

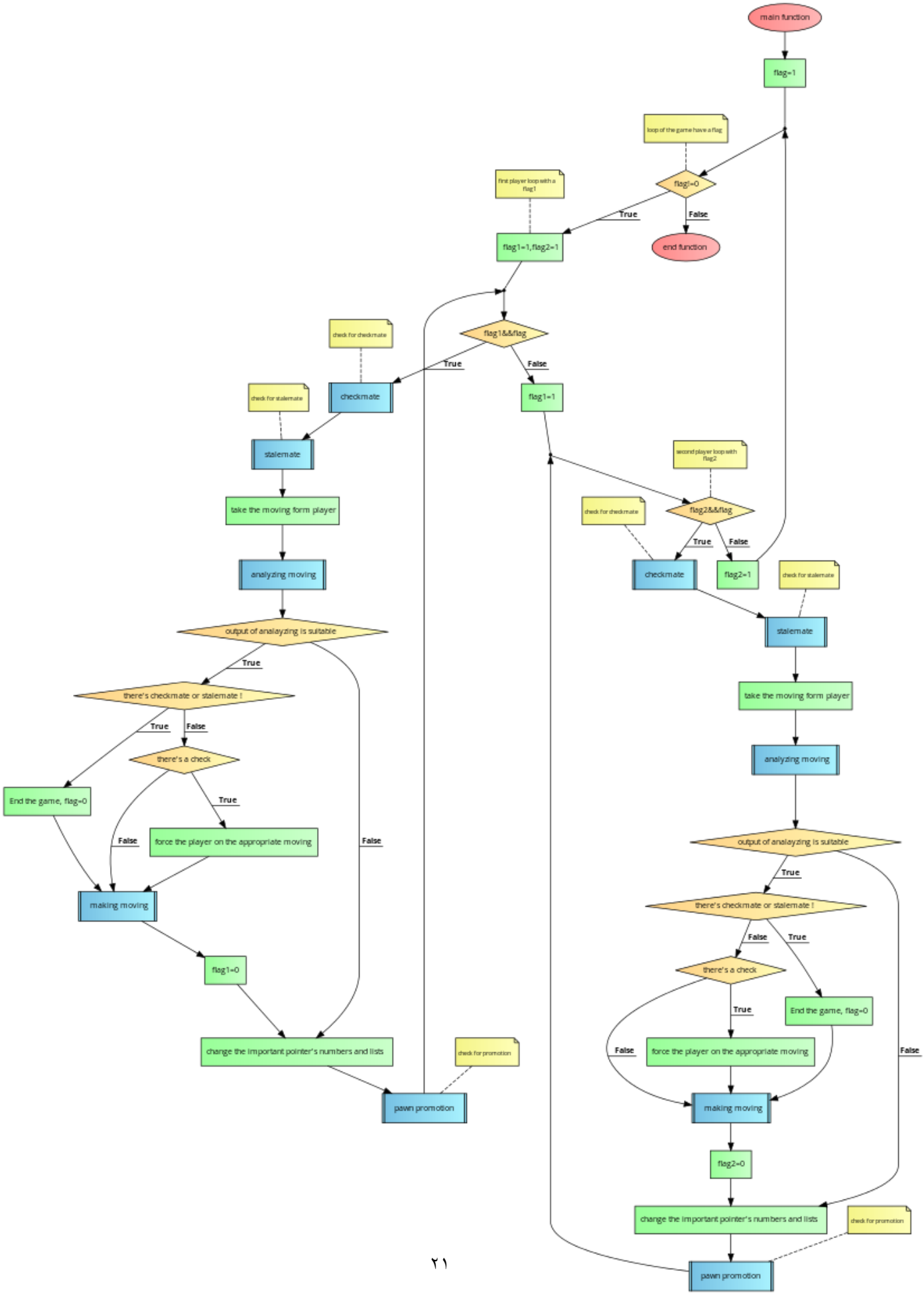
check for promotion//

؛||pawn promotion||

{

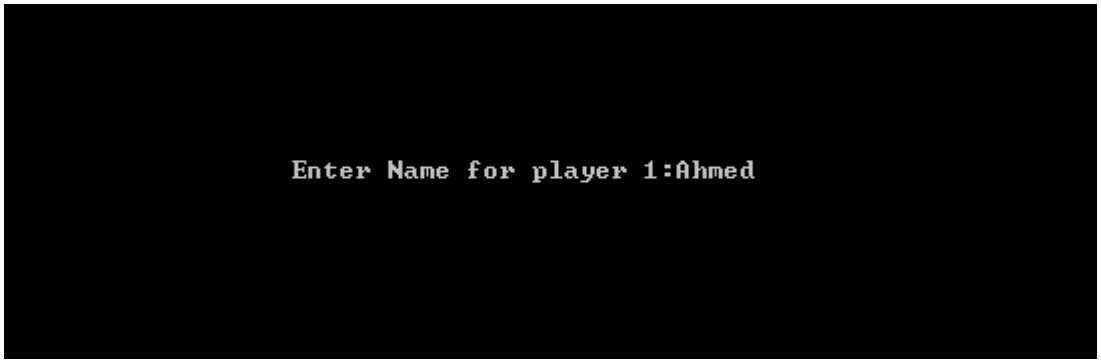
{؛flag2=1

end function;


main function
flag=1
loop of the game have a flag
flag!=0
True
False
end function
first player loop with a flag1
flag1=1,flag2=1
flag1&&flag
True
False
check for checkmate
checkmate
check for stalemate
stalemate
flag1=1
take the moving form player
analyzing moving
output of analayzing is suitable
True
there's checkmate or stalemate !
True
False
there's a check
True
End the game, flag=0
False
force the player on the appropriate moving
False
making moving
flag1=0
change the important pointer's numbers and lists
check for promotion
pawn promotion
second player loop with flag2
flag2&&flag
check for checkmate
True
False
checkmate
flag2=1
check for stalemate
stalemate
take the moving form player
analyzing moving
output of analayzing is suitable
True
there's checkmate or stalemate !
False
True
there's a check
True
End the game, flag=0
False
force the player on the appropriate moving
False
making moving
flag2=0
change the important pointer's numbers and lists
check for promotion
pawn promotion

Sample runs:

Enter player's name , for Example: Ahmed & Mohamed



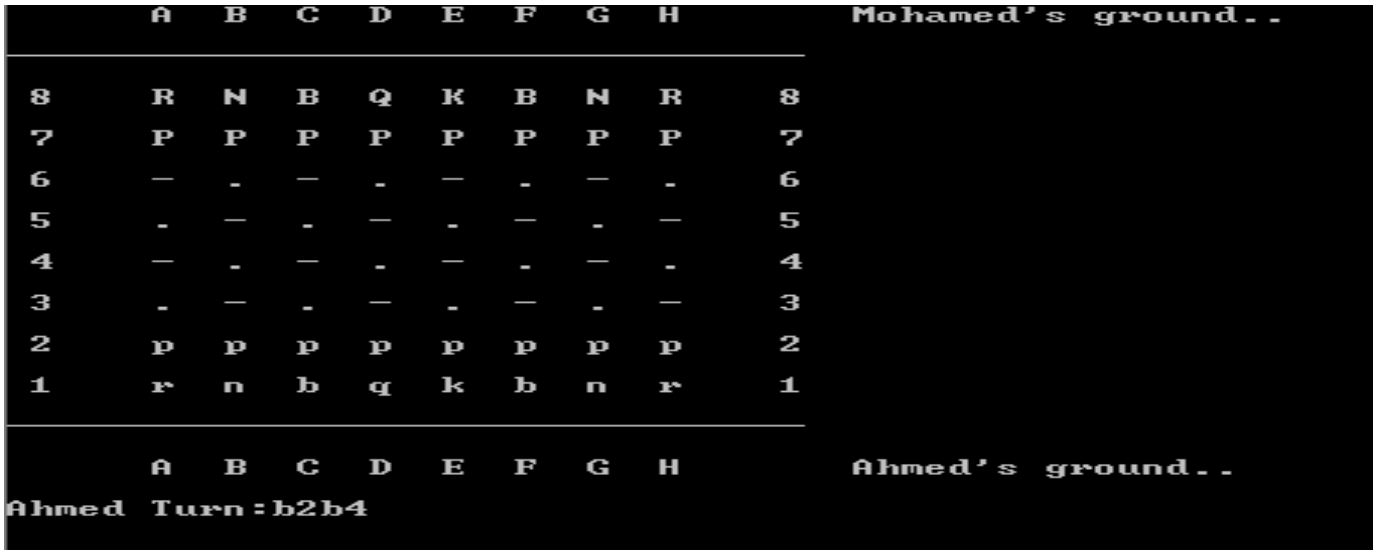
Now we have this window:

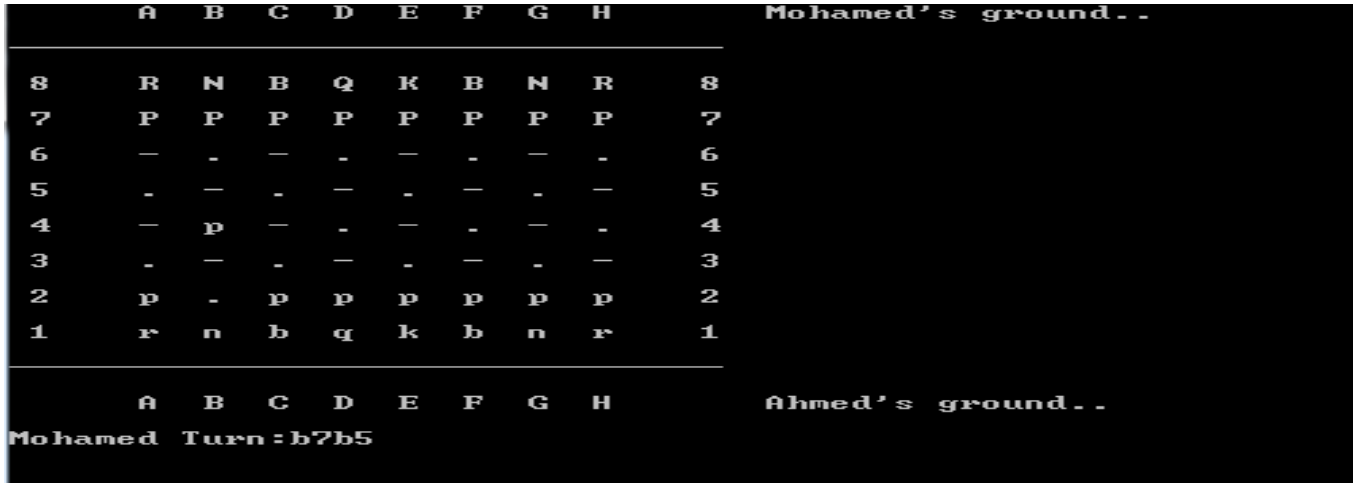
It's ahmed turn's with the below pieces as illustrated before and on his right there's where his dead pieces are..

The same things for the second player "Mohamed" above..

Let's make 2 tests for promotion and checkmate to be our sample example runs..

1)promotion:





```
     A  B  C  D  E  F  G  H        Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  P  .  -  .  -  .  -   5
 4   -  p  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   p  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1
 ------------------------------
     A  B  C  D  E  F  G  H        Ahmed's ground..
Ahmed Turn:a2a4
```

```
     A  B  C  D  E  F  G  H        Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  P  .  -  .  -  .  -   5
 4   p  p  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1
 ------------------------------
     A  B  C  D  E  F  G  H        Ahmed's ground..
Mohamed Turn:b5a4
```

```
     A  B  C  D  E  F  G  H        Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   P  p  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1  p
 ------------------------------
     A  B  C  D  E  F  G  H        Ahmed's ground..
Ahmed Turn:b4b5
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  p  .  -  .  -  .  -   5
 4   P  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1  p
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Mohamed Turn:a4a3
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  p  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   P  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1  p
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Ahmed Turn:b5b6
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  p  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   P  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1  p
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Mohamed Turn:a3a2
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  -  P  P  P  P  P  P   7
 6   -  p  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   P  .  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1  p
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Ahmed Turn:b6b7
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
-----------------------------
8    R  N  B  Q  K  B  N  R    8
7    P  p  P  P  P  P  P  P    7
6    -  .  -  .  -  .  -  .    6
5    .  -  .  -  .  -  .  -    5
4    -  .  -  .  -  .  -  .    4
3    .  -  .  -  .  -  .  -    3
2    P  .  p  p  p  p  p  p    2
1    r  n  b  q  k  b  n  r    1  p
-----------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Mohamed Turn:a2b1
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
-----------------------------
8    R  N  B  Q  K  B  N  R    8
7    P  p  P  P  P  P  P  P    7
6    -  .  -  .  -  .  -  .    6
5    .  -  .  -  .  -  .  -    5
4    -  .  -  .  -  .  -  .    4
3    .  -  .  -  .  -  .  -    3
2    -  .  p  p  p  p  p  p    2
1    r  P  b  q  k  b  n  r    1  p n
-----------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
You have a promotion to  Q/N/B/R : Q
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  p  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   -  .  p  p  p  p  p  p   2
 1   r  Q  b  q  k  b  n  r   1  p n
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Ahmed Turn:b8a6
```

2)checkmate:

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  P  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  -  .  -   3
 2   p  p  p  p  p  p  p  p   2
 1   r  n  b  q  k  b  n  r   1
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Ahmed Turn:f2f3
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  P  P  P  P  P  P  P   7
 6   -  .  -  .  -  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  p  .  -   3
 2   p  p  p  p  p  .  p  p   2
 1   r  n  b  q  k  b  n  r   1
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Mohamed Turn:e7e6
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  P  P  P  .  P  P  P   7
 6   -  .  -  .  P  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  -  .   4
 3   .  -  .  -  .  p  .  -   3
 2   p  p  p  p  p  .  p  p   2
 1   r  n  b  q  k  b  n  r   1
 ------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Ahmed Turn:g2g4
```

```
     A  B  C  D  E  F  G  H       Mohamed's ground..
 ---------------------------------
 8   R  N  B  Q  K  B  N  R   8
 7   P  P  P  P  .  P  P  P   7
 6   -  .  -  .  P  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  p  .   4
 3   .  -  .  -  .  p  .  -   3
 2   p  p  p  p  p  .  -  p   2
 1   r  n  b  q  k  b  n  r   1
 ---------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
Mohamed Turn:d8h4
```

```
 ---------------------------------
 8   R  N  B  .  K  B  N  R   8
 7   P  P  P  P  .  P  P  P   7
 6   -  .  -  .  P  .  -  .   6
 5   .  -  .  -  .  -  .  -   5
 4   -  .  -  .  -  .  p  Q   4
 3   .  -  .  -  .  p  .  -   3
 2   p  p  p  p  p  .  -  p   2
 1   r  n  b  q  k  b  n  r   1
 ---------------------------------
     A  B  C  D  E  F  G  H       Ahmed's ground..
It's a checkmate , player1 lose !
Process returned 0 (0x0)   execution time : 300.870 s
Press any key to continue.
```

Indexes: