REPORT OF PROJECT

1. Description :

The game called dots and boxes.

You can play as one player or two players . The game starts with an empty grid of dots .Two players take turns adding single horizontal or vertical line . The player who completes the forth of a 1x1 box earns one point and takes another turn . Any player can make undo or redo for a play and save the game or load it .

2. Features :

The game contains dots and the players connecting lines between the dots . Each player has his own color to play with it . Player one has the blue color and the other has the red color .

The game contains timer .

3. Design overview :

1. In 2x2 mode :

It contains 4 boxes and 9 dots .

2. In 5x5 mode :

It contains 25 boxes and 36 dots.

3. player win when takes more boxes than the other and marking the boxes with his color.
4. Remaining moves .
5. Score for each player .
6. Moves of each player .
7. Timer .
8. Player turn .

4. Assumption :

We assume that the horizontal lines consist of dashes and the vertical lines consist of pipes .

We assume that when player VS computer , Computer takes a name directly "Computer".

5. Data structures :

Structures :

1. Best : it is an array of structures contains the palyer name and his score .

Arrays :

1. A : Contains all players' moves and who played the move .
2. B : contains undo's moves and who made it .
3. Easy : contains the shape of the grid in mode 2x2 .
4. Hard : contains the shape of the grid in mode 5x5 .
5. H : character array to take name of player 1 .
6. P : character array to take name of player 2 .
7. Some of character arrays to take inputs and turn them into integers .

6. Functions :

1. Gready : it is for computer turn and it checks if the computer has a moves that will make him win a point and then play it .
2. Error_easy : check if the inputs are wrong and give the player another try in 2x2 mode .
3. Error_hard : check if the inputs are wrong and give the player another try in 5x5 mode .
4. Score : checks if the move will close a box in the grid and increase the score of player who played it .
5. Turn : it turns inputs to our logic to represent the move in the grid and calculate the score .
6. Gotoxy : it makes us go to the co-ordinate that we want to represent the move in the grid.
7. Color : to choose the color for each player .

7. User manual :

1. To gain point : you should close a square .

2. When you get a point you will have an extra move .

How to input :

1-you enter the number of the row where you start you move ,then press Enter.

2-you enter the number of the row where you end your move ,press Enter.

3-you enter the number of the column where you start you move ,then press Enter.

4-you enter the number of the column where you end your move ,then press Enter.

Some notes for playing :

1-To undo a play , you should enter '100' in the first input.

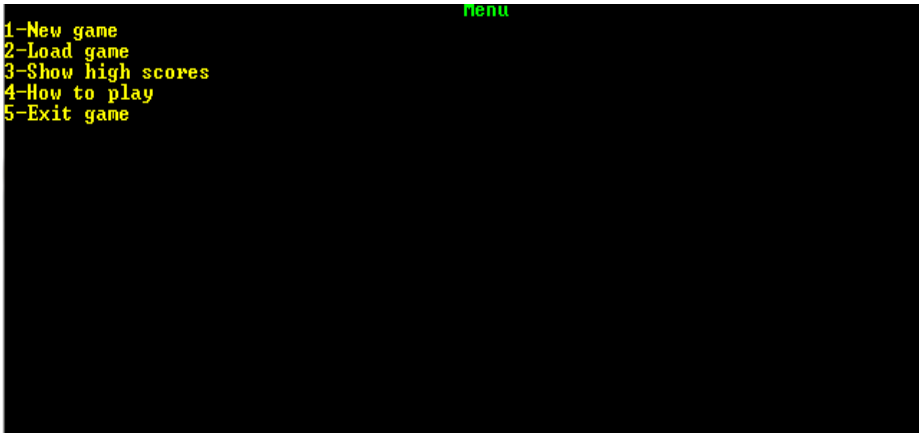
2-To redo a play , you should enter '200' in the first input.

3-To save the game , you should enter '300' in the first input.

4-To load the game , choose 'Load game' from the menu and choose the file.

8. Screenshots :

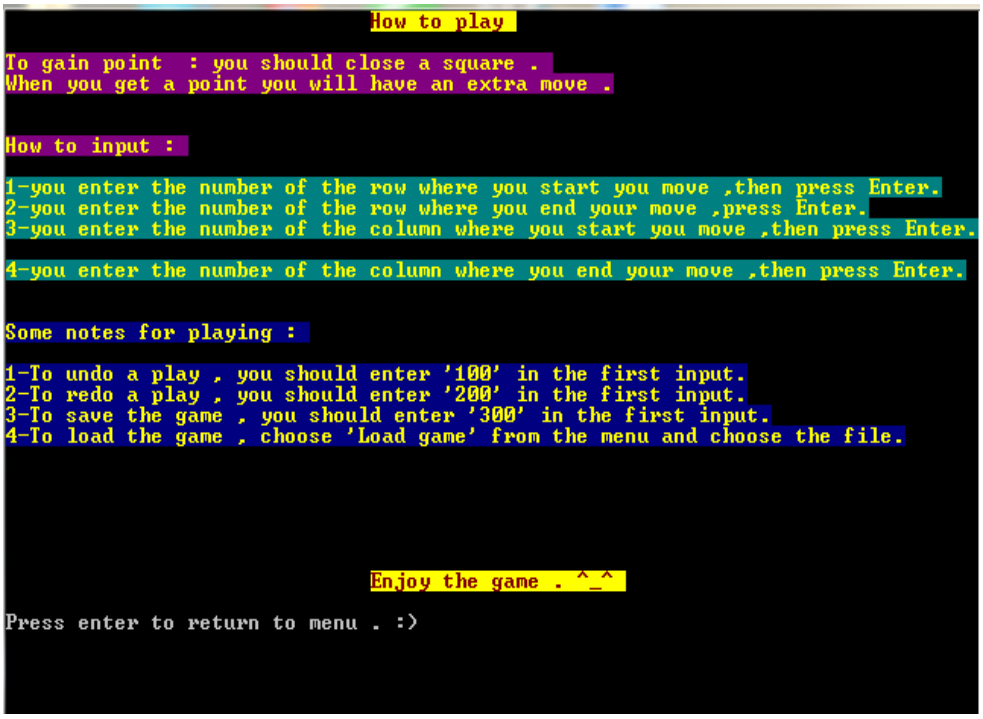
Menu :



Load game :

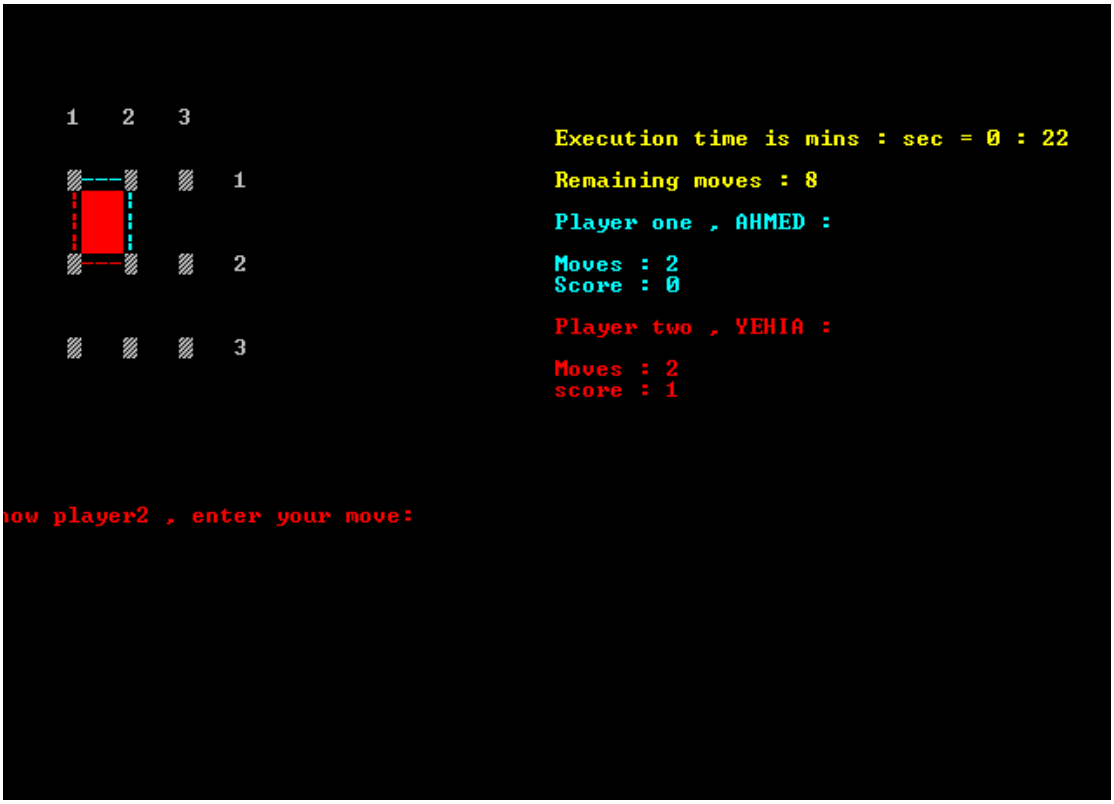
Top score :

How to play :



Playing 2x2 :

Winnig a point :



End game :

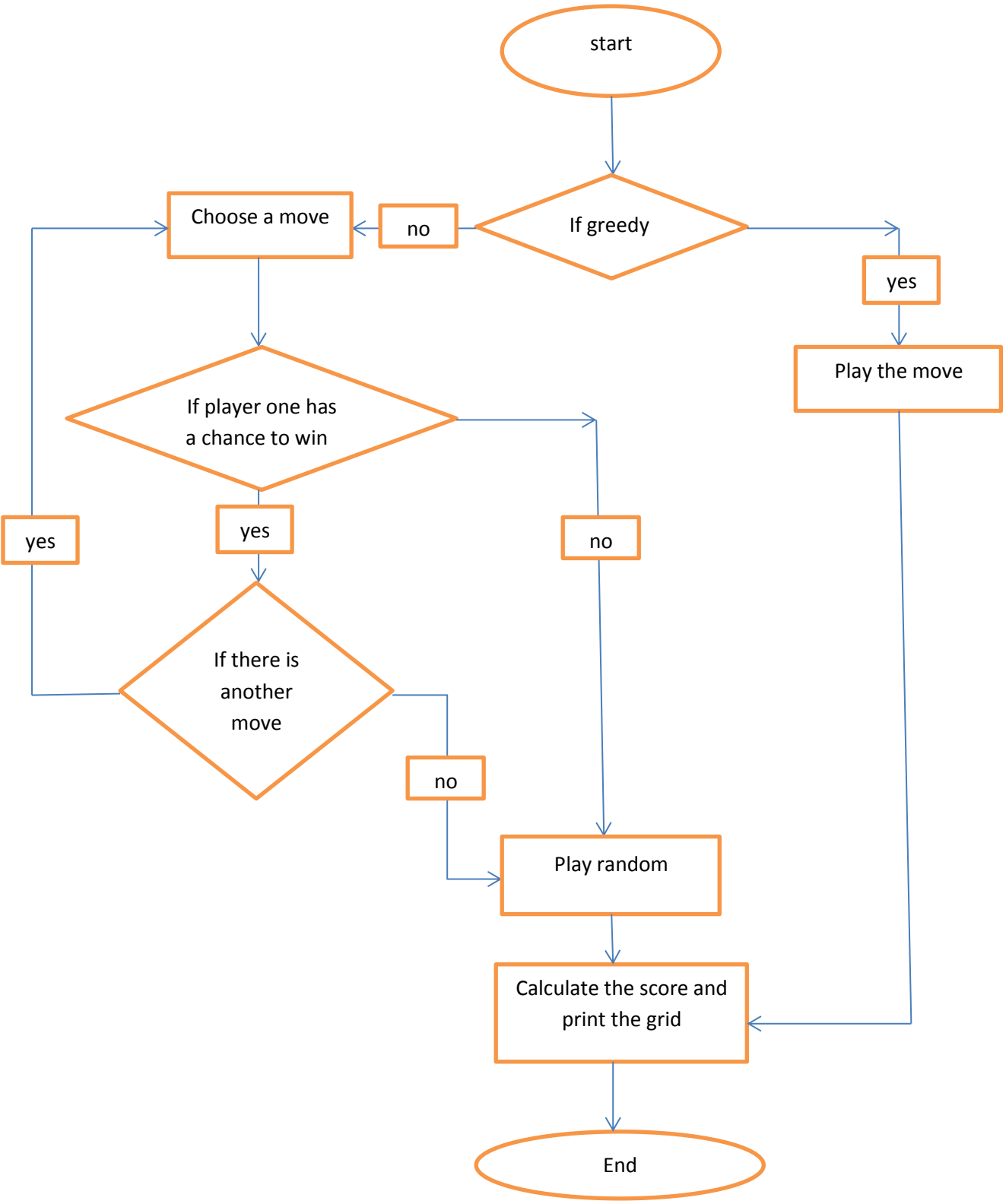
ahmed won the game
Press enter to return to menu :

5x5 :

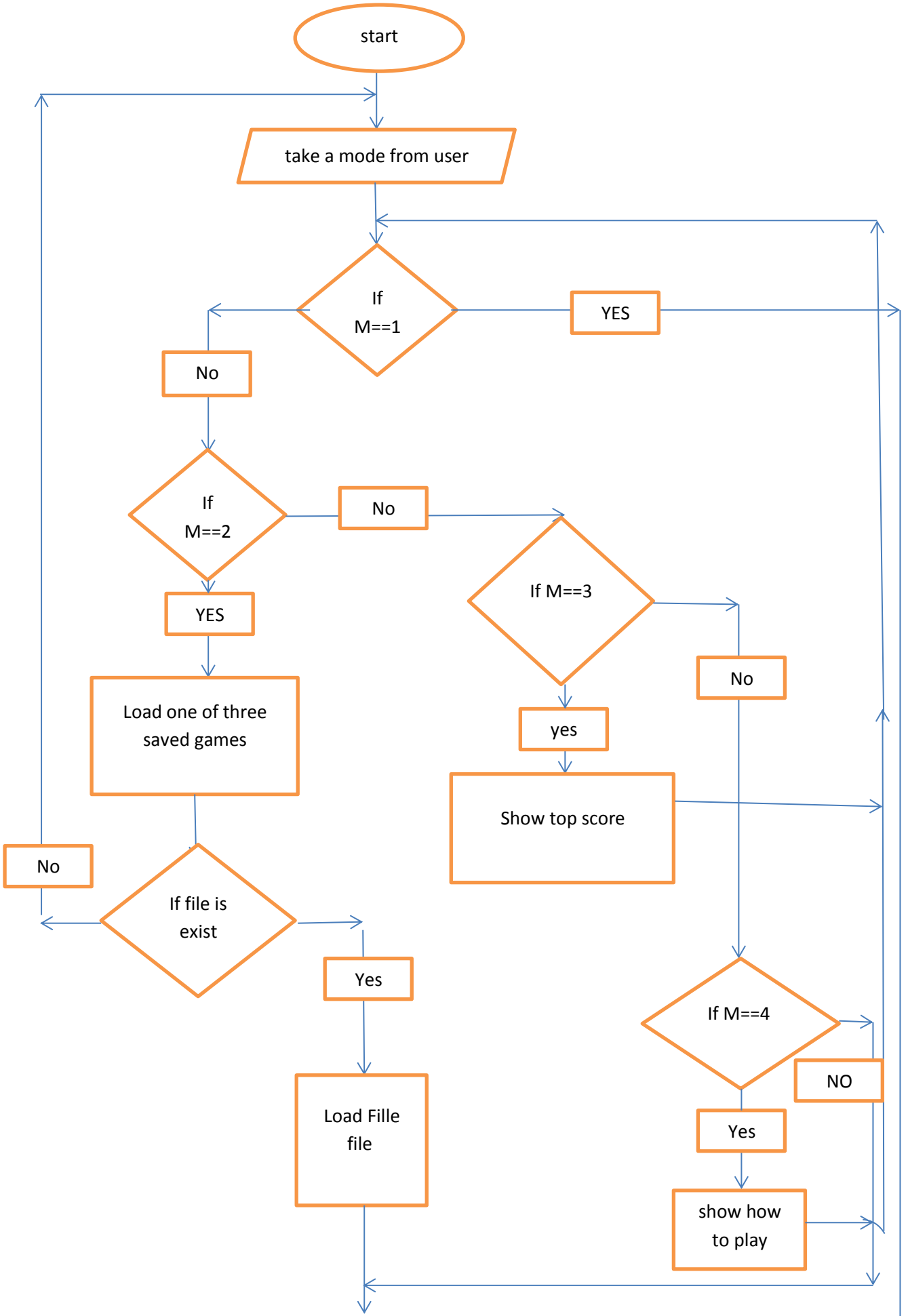
1 2 3 4 5 6
1
2
3
4
5
6
Execution time is mins : sec = 0 : 51
Remaining moves : 56
player one , yehya :
moves : 2
score : 0
player two , ahmed :
moves : 2
score : 1
now player2 , enter your move :
1
1
2
3

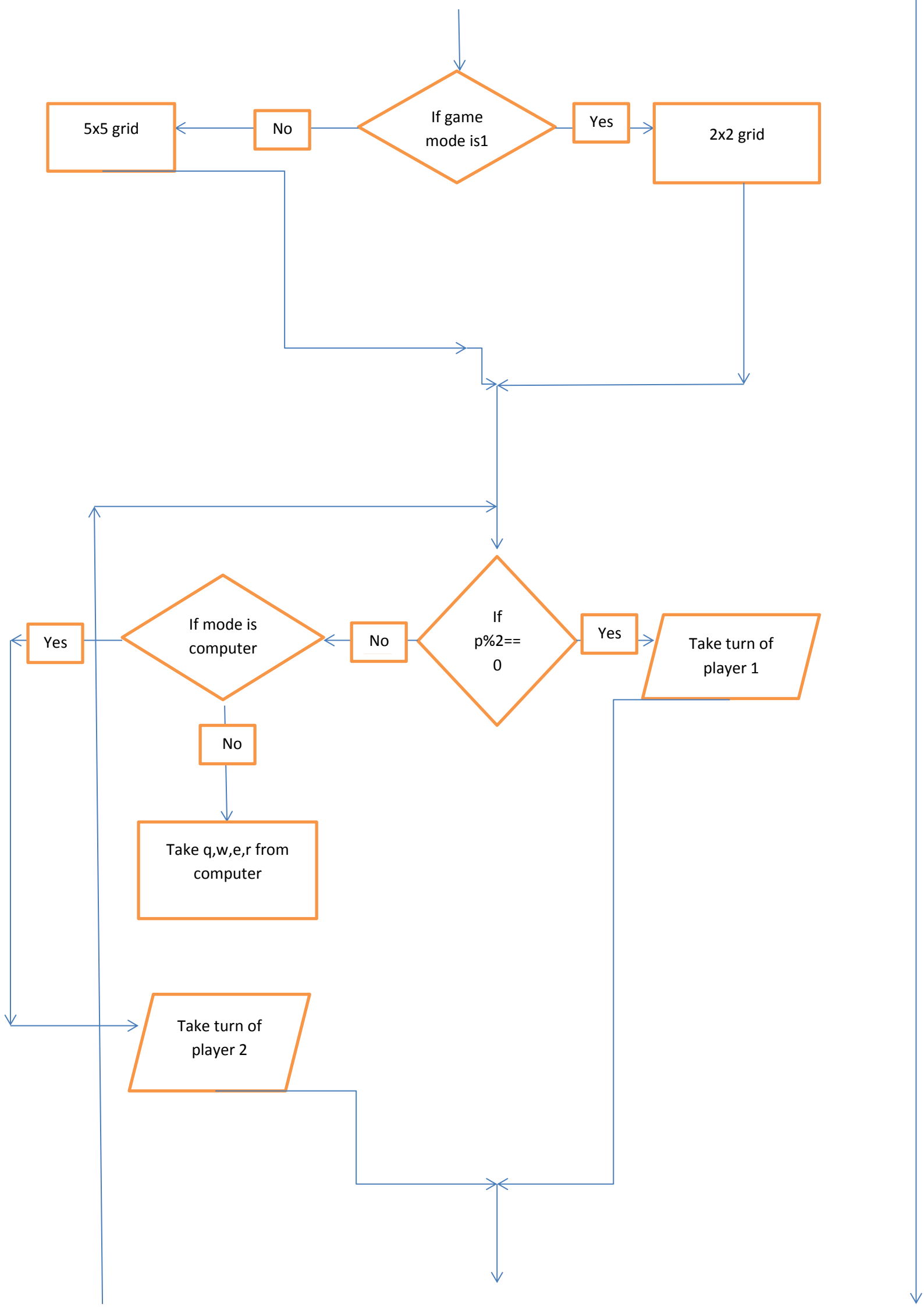
9. Flowcharts :

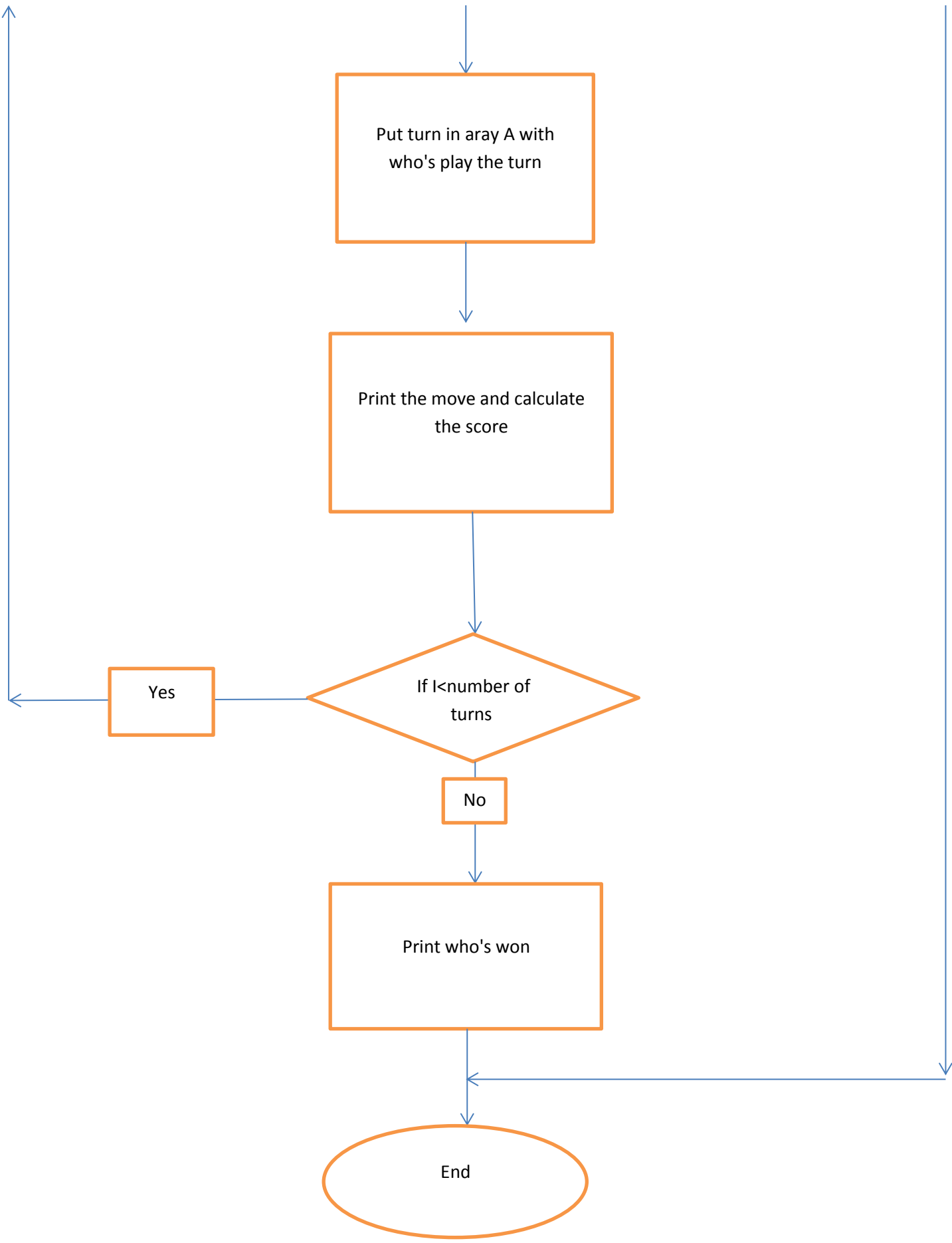
1. Computer algorithms :



2-Game loop


start
take a mode from user
If
M==1
YES
No
If
M==2
No
YES
Load one of three
saved games
If M==3
No
yes
Show top score
No
If file is
exist
Yes
Load Fille
file
If M==4
NO
Yes
show how
to play


If game mode is1
No
5x5 grid
Yes
2x2 grid
If p%2== 0
No
If mode is computer
Yes
No
Take q,w,e,r from computer
Yes
Take turn of player 1
Take turn of player 2


Put turn in aray A with who's play the turn
Print the move and calculate the score
If I<number of turns
Yes
No
Print who's won
End

10 . Pseuodo code :

1. Undo :
   If I =0 it will not make undo for the play in the first play .
   Else
   Put q,w,e,r in Array A by sequence
   And put them in Array B to be used in case of Redo .

   And then make all elements in last row array A = 0.

   B[J][0]=A[I][0]
   B[J][1]=A[I][1]
   B[J][2]=A[I][2]
   B[J][3]=A[I][3]
   B[J][4]=A[I][4]
   A[I][0]=0
   A[I][1]=0
   A[I][2]=0
   A[I][3]=0
   A[I][4]=0
   Calculate the new score after undo by calling the function 'score'
   s=score1(easy,s,q,w,e,r) in 2x2 mode or s=score1(hard,s,q,w,e,r)
   in 5x5 mode .
   print the grid by printing the moves that in array A.
   by calling the function that prints the grid .

2. Redo :

If J = 0 it will not make the redo game as the arrays that contain undo's plays is empty .

Putting the moves that in array B in array A to print it afterwards .

And then make all elements in array B = 0

A[I][0]=B[J][0]

A[I][1]=B[J][1]

A[I][2]=B[J][2]

A[I][3]=B[J][3]

A[I][4]=B[J][4]

B[J][0]=0

B[J][1]=0

B[J][2]=0

B[J][3]=0

B[J][4]=0

Calculate the new score after Redo by calling the function 'score' s=score1(easy,s,q,w,e,r) in 2x2 mode or s=score1(hard,s,q,w,e,r) in 5x5 mode .
print the grid by printing the moves that in array A.
by calling the function that prints the grid .

11. References :

Gotoxy : http://www.programmingsimplified.com/c/conio.h/gotoxy

Color : https://www.wikihow.com/Get-Color-in-C-Program

Time : https://www.tutorialspoint.com/c_standard_library/c_function_difftime.htm