Connect 4 Report

- Rowan Gamal Ahmed Elkhouly
  ID: **21010539**
- Rowan Mohamed Youssef Ali
  ID: **21010549**

1) Description:

It is required to implement (Connect four) game for our Final project for the first semester in the computer & systems department, using C programming language
The project is developed by: Rowan Gamal & Rowan Mohammed

GitHub Link: https://github.com/Rowan30/Final_project

2) Features:

- 3 Game Modes:
  - Human vs Human
  - Human vs computer (easy)
  - Human vs computer (medium)
- Playing a game
  - From scratch (new game)
  - An already saved game
- Displaying a rank for the top players of the game (sorted rank)
- Allowing the user to
  - Undo
  - Redo
  - Save the game
  - Change configurations of the game through xml file
- Displaying in each turn
  - The updated board
  - Number of moves of each player
  - Time passed since the start of the game
  - Scores of both parties during game

3) Design Overview:

- Menu display:
  - Ordered letter display of the menu options (a to e)

    - Sound is added whenever each choice pop in the console
- Game Display:
    - We used '|' to separate cells in columns and '-' to separate rows
    - Board is printed first then, the rest of the instructions are printed below

4) Assumptions:

- Assumptions in the **XML file**:
    - Any XML file having added tags other than (configuration, height, and width) is considered corrupted
- Assumptions in human vs **easy** computer mode
    - The computer input is random
- Assumptions in **winner determination**
    - There's a winner given that one's score is bigger than the other
    - If there's any draw, then there's no rematch, the user has the liberty to choose whether to go to the main menu or exit game
- Assumptions in **height and width** of the game
    - If the height or width taken from the user is less than allowing a player to score, then the game continues normally, but scores of both will be 0 till the end, meaning there'll be no winner
- Assumptions in **high scores**
    - If 2 players score the same high score, then the old player comes first in rank followed by the new player (since the old player's record wasn't beaten but equalized)
- Assumptions in **Undo & Redo**
    - Player can only undo till first move (only one which won't be undone) made since last undo process
    - Player won't be able to redo if he/she inputs a move after undo, because undo history then, is cleared, so redo will be considered wrong input

5) Data structure:

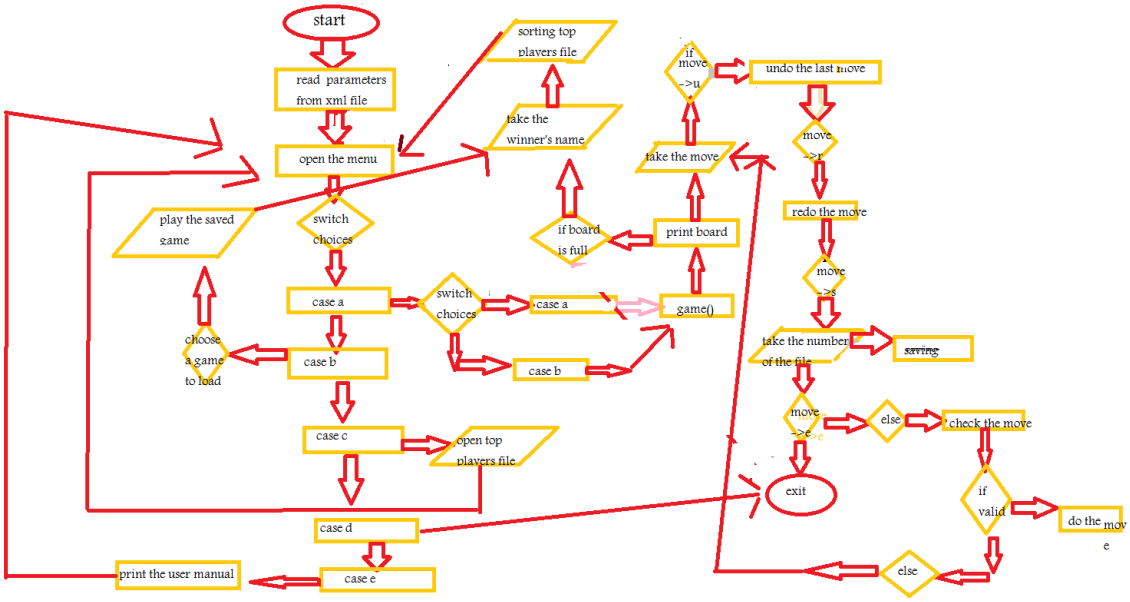
1. 2D Array (board) of characters
2. 1D Array character (move) of size 20
3. Structure for players containing the player name and score
4. 1D array (undo) for saving the moves in case the user want to undo or redo
5. Struct array (top) contains the top players and sorting them again whenever a new player is added

6. 1D Array of size = columns in both computer modes, to determine the vacancy of every column

6) Description of the Functions:

1. void menu() to print the main menu when you call it.
2. void game() function for human vs human mode.
3. void computerEasy()function for human vs computer easy mode
4. void computermed() function for human vs computer medium mode(bonus)
5. void game_end(char choice) at the end of the ask the user if he want to end the game or return to the main menu by taking his choice .
6. int Hcheck(int x,int y,char board[x][y],int row,int move)
7. int Vcheck(int x,int y,char board[x][y],int row,int move)
8. int D1check(int x,int y,char board[x][y],int row,int move)
9. int D2check(int x,int y,char board[x][y],int row,int move)
   Functions to check the connected four in each direction by taking the array and the value of the (correct move) index, do a specific algorithm and return the new score after each move
   (the new score is the sum of the scores returned from all four functions).
10. void show_scores();
11. void sort_top();
12. void add_score(); for adding a new score in the ranking sheet.
13. int check_number(char move[20]);
14. int scores_sum(int x,int y,char board[x][y],int row,int move);function to calculate the total score of each player.
15. void saving(int x,int y,char board[x][y],int turn,int n,int score1,int score2); function for saving a game.
16. void loadsaved(); for loading any of the saved games.
17. void playsaved(int x,int y,char board[x][y],int turn,int score1,int score2); for continue the saved game.
18. void readxml(char str[30],int trial);for reading parameters from the xml configurations file .
19. void print(int x,int y,char board[x][y]) function to print the updated board after each move.
20. Three files for saving different three games and a file for saving top scores (are created if they don't exist)
21. File for saving one easy computer mode
22. XML file for game parameters
23. Top scorers file (is created if it doesn't exist at the beginning)

7) FlowChart:



8) User Manual:

When you start the game,choose by using characters (a/b/c/d/e):

- if you want to start a new game
- or load a previous game
- or open the top players' file
- or look for help

If you start a new game you should choose between the game modes:

- human vs human
- human vs computer
    a. Easy
    b. medium

By choosing any one of them, we prepare an empty board for you and you could play by entering a number (1,2,3….) which indicates the column number
Scores, moves and timer are displayed during every turn.

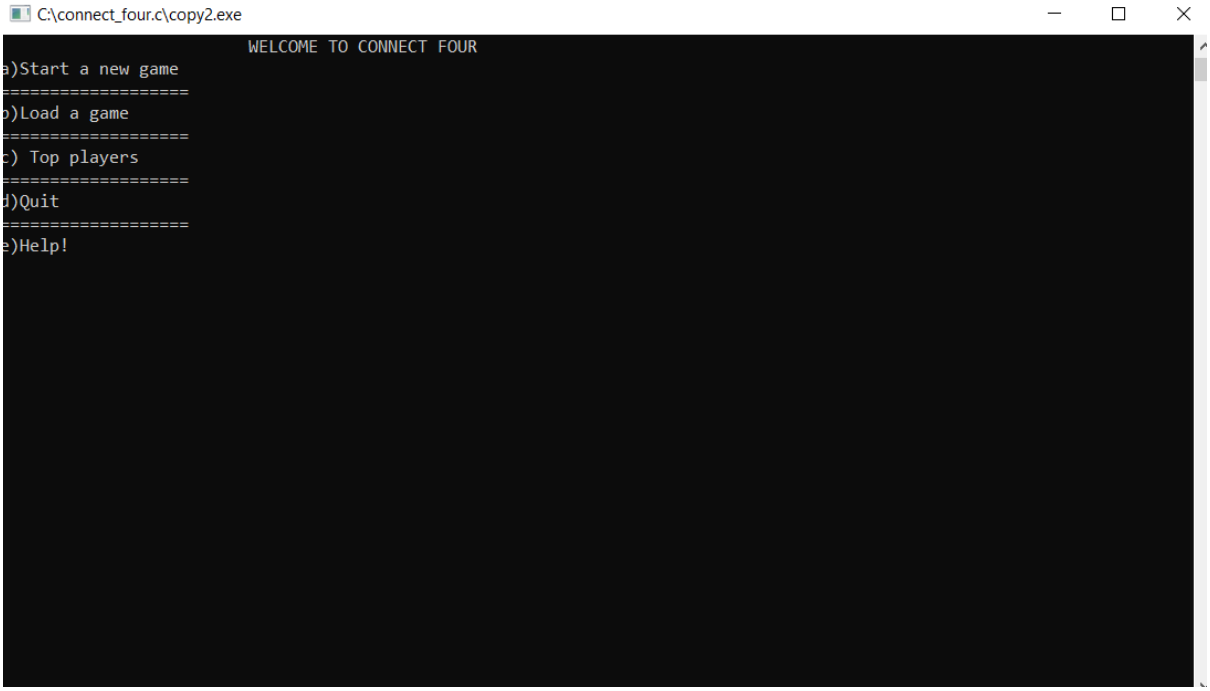
When you play the game you can press (u) for undo the last move or
(r) for the redo, or (s) for save, and (e) for exiting without saving.
You continue playing until the board is full

When board is full, we compute the score of each player and take the name of the user with the higher score and add him to the ranking sheet.

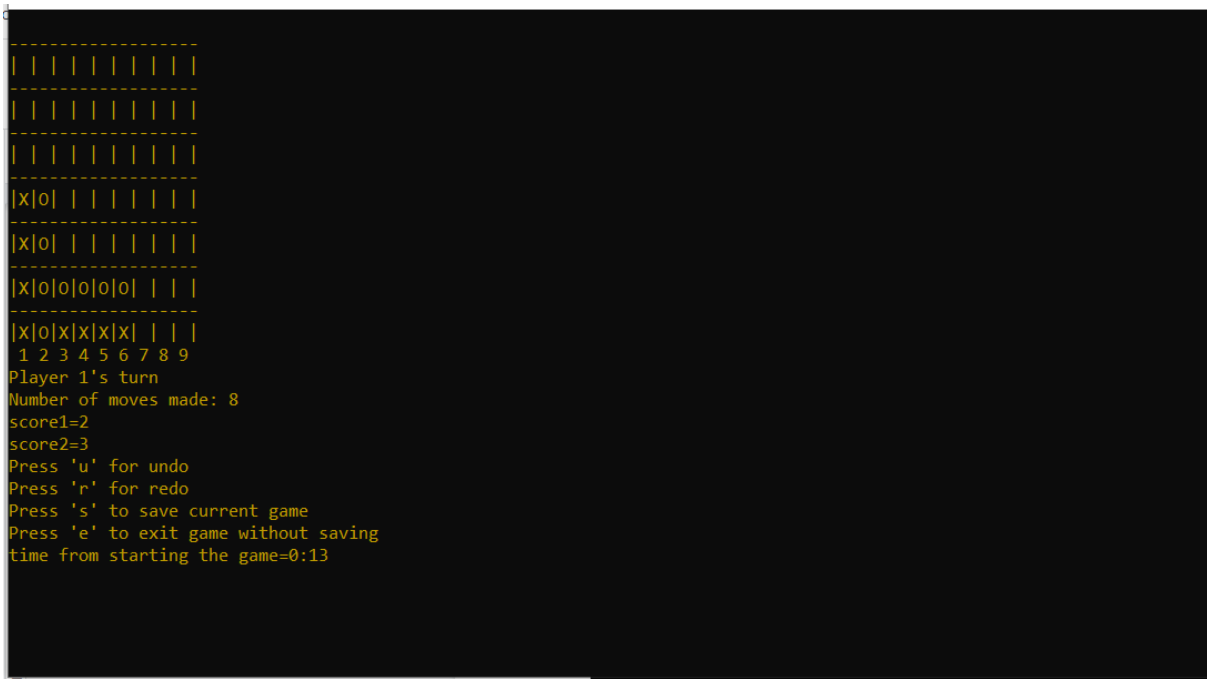
Then you are asked if you want to return to the main menu or exit game

9) Sample Runs:

Starting the game:



Human vs human mode:



If you press undo

If you press redo(the time is increasing)

Taking the winner's name:

Updating the highscores list

```
[1] rowan       18
[2] riri        16
[3] rmrm        7
[4] raaa        6
[5] rawan       6
[6] rooww       5
[7] roowaa      5
[8] rooo        4
[9] roww        4
[10] rawaan     4

Process returned 0 (0x0)   execution time : 64.671 s
Press any key to continue.
```

Loading a game:

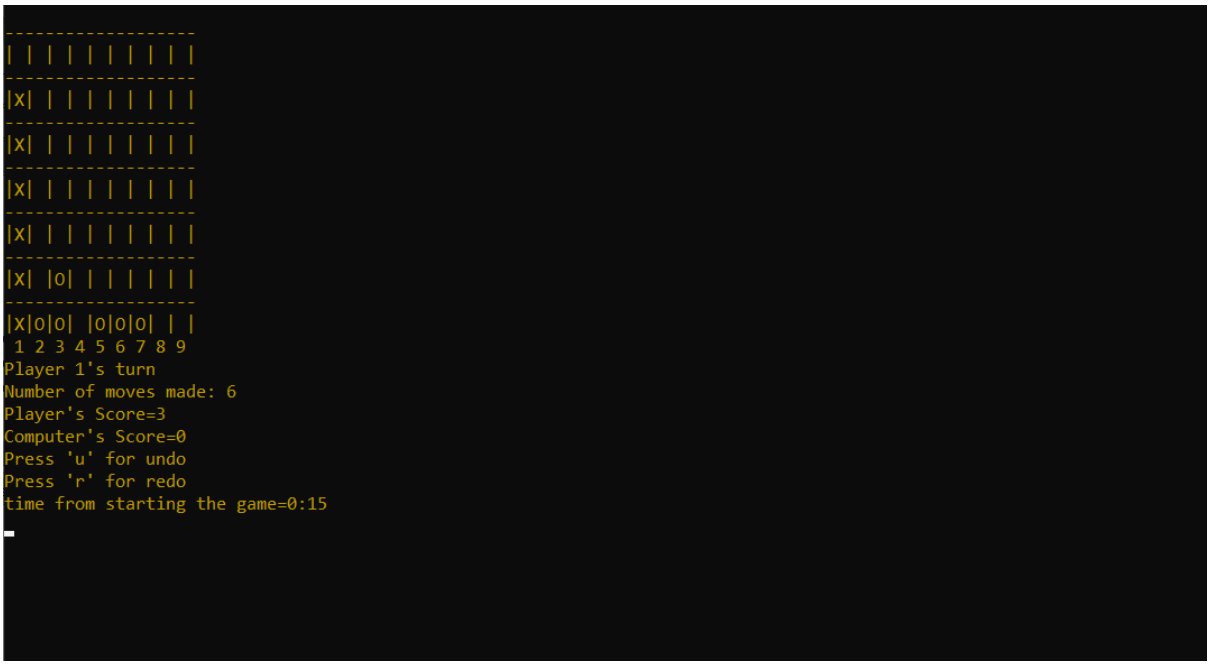
```
choose the saved game to load

Choose which game to load(1/2/3)?
```

```
choose the saved game to load

Choose which game to load(1/2/3)?1

------------------
| | | | | | | | | |
------------------
| | | | | | | | | |
------------------
| | | | | | | | | |
------------------
|X|O| | | | | | | |
------------------
|X|O| | | | | | | |
------------------
|X|O| | | | | | | |
------------------
|X|O| | | | | | | |
------------------
 1 2 3 4 5 6 7 8 9
Player 1's turn
Number of moves made: 4
score1=1
score2=1
Press 'u' for undo
Press 'r' for redo
Press 's' to save current game
Press 'e' to exit game without saving
time from starting the game=0:0
```

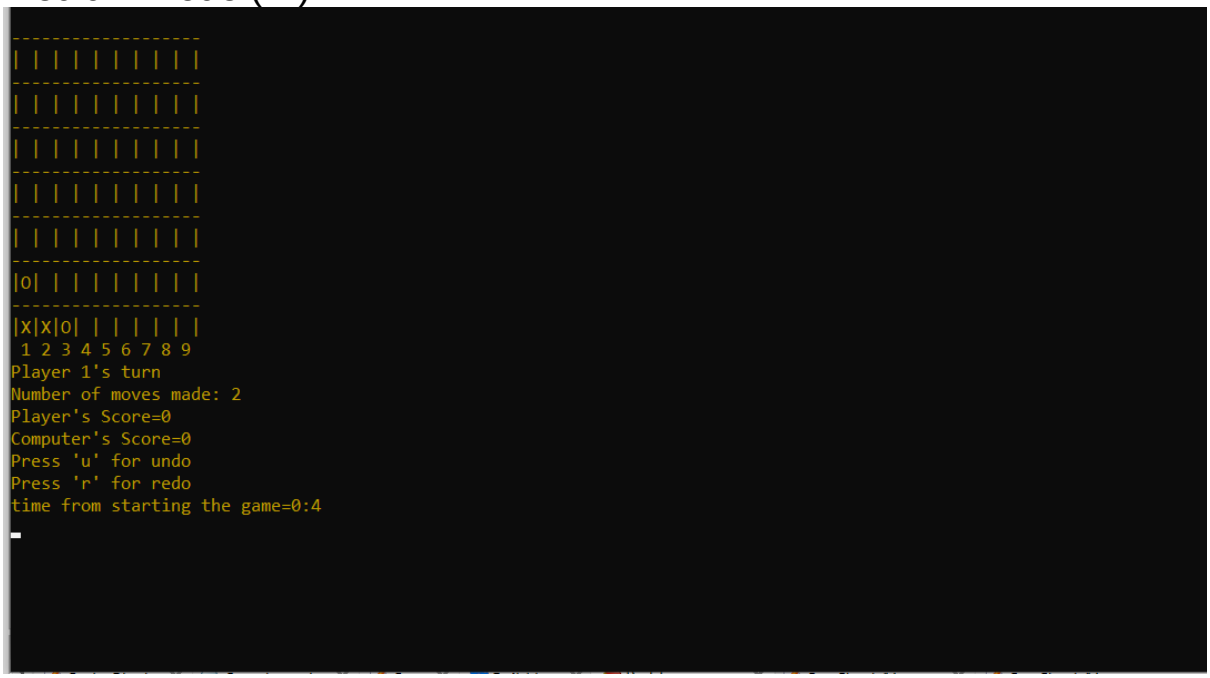
Two modes for computer:

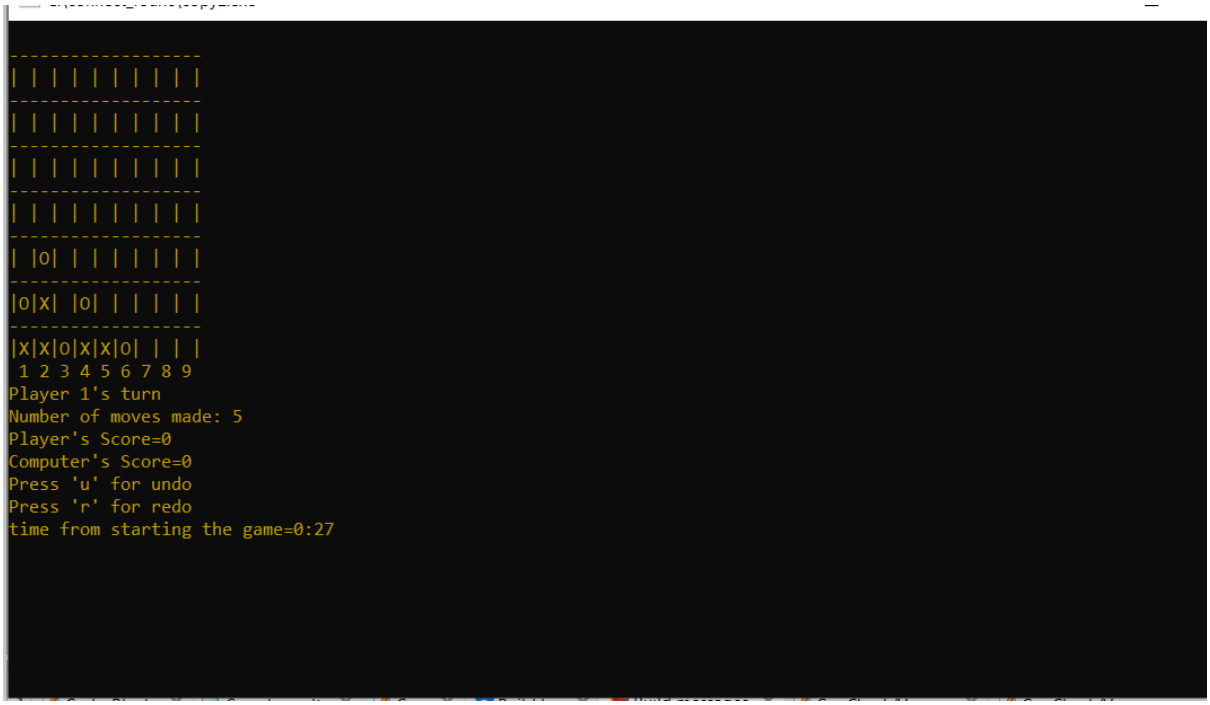
```
choose the mode of computer [a)easy OR b)medium]
```

Easy mode(random playing)

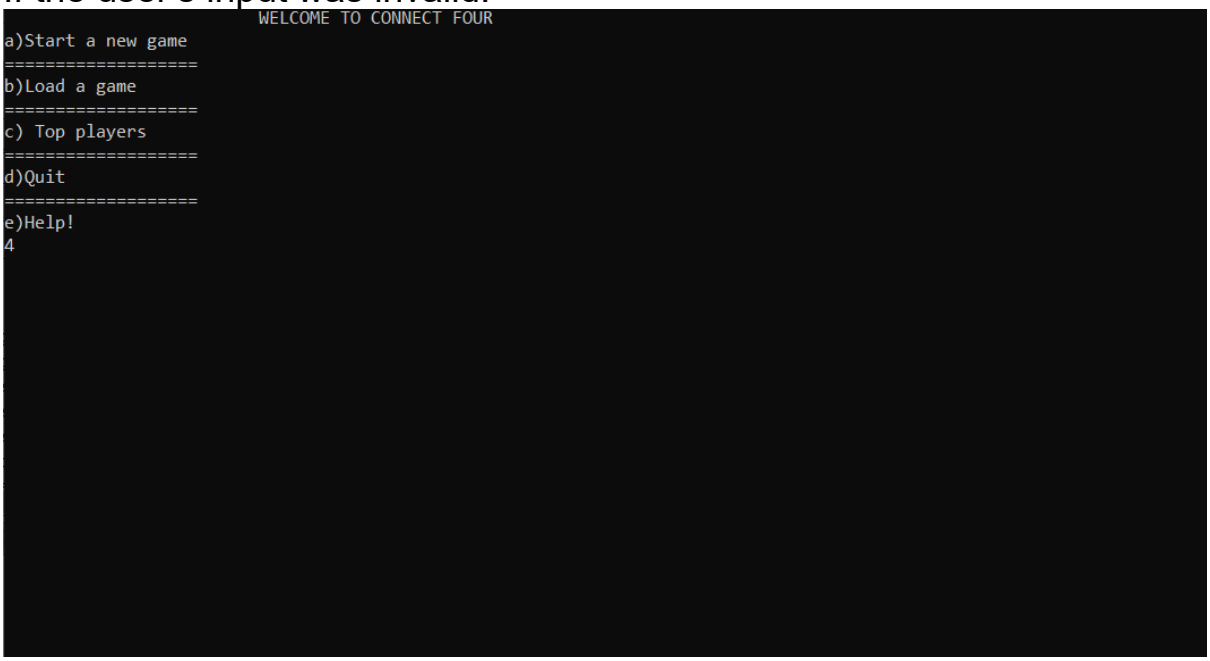


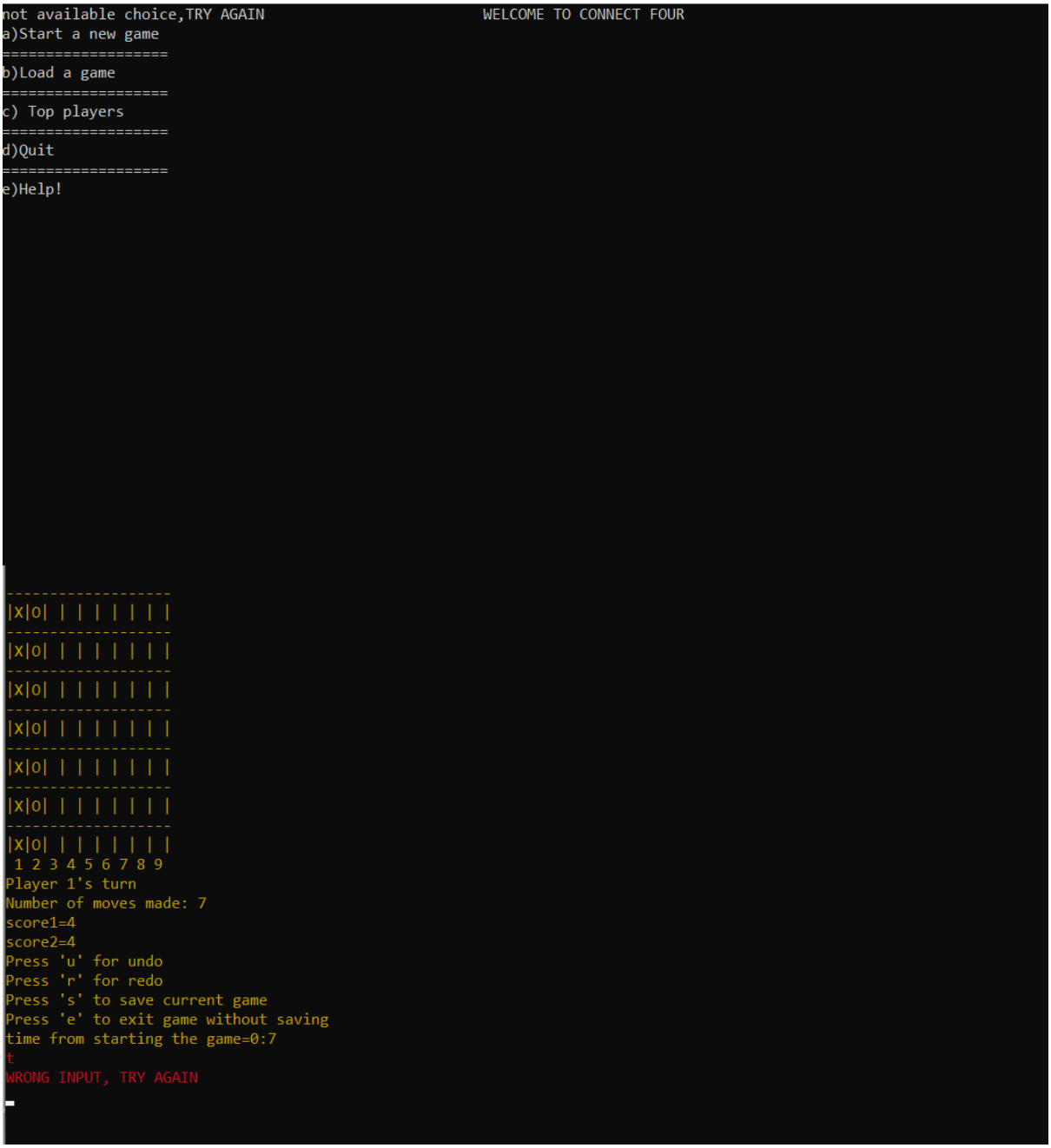
Medium mode (AI)

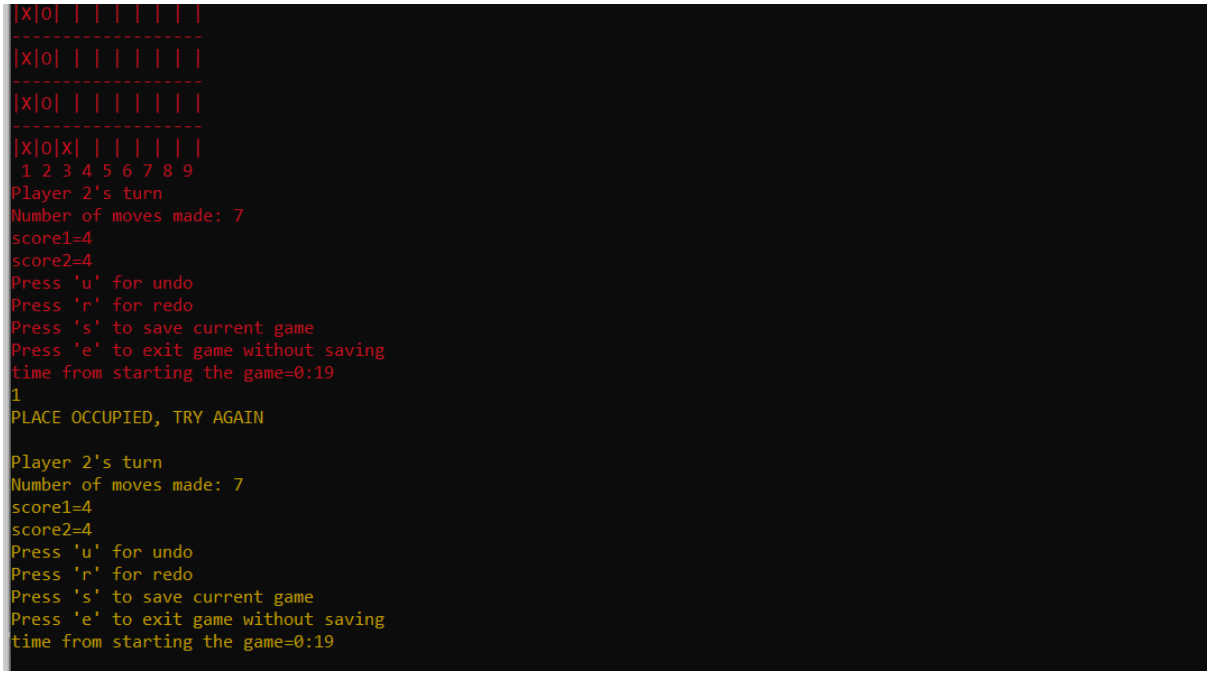




If the user's input was invalid:




not available choice,TRY AGAIN
WELCOME TO CONNECT FOUR
a)Start a new game
==========
b)Load a game
==========
c) Top players
==========
d)Quit
==========
e)Help!
|X|O| | | | | | |
1 2 3 4 5 6 7 8 9
Player 1's turn
Number of moves made: 7
score1=4
score2=4
Press 'u' for undo
Press 'r' for redo
Press 's' to save current game
Press 'e' to exit game without saving
time from starting the game=0:7
t
WRONG INPUT, TRY AGAIN



10) References:

1) We learnt how to set colours from:
https://c-for-dummies.com/blog/?p=5270

2) We understood isalpha() and atoi() functions from:
https://www.techonthenet.com/c_language/standard_library_functions/stdlib_h/atoi.php
https://www.geeksforgeeks.org/isalpha-isdigit-functions-c-example/

3) We learnt how to print the board in the same place from:
https://www.geeksforgeeks.org/clear-console-c-language/

4) We understood clock_t data type and clock() function from:
https://www.geeksforgeeks.org/how-to-measure-time-taken-by-a-program-in-c/?ref=lbp

5) How to add music for the game
https://www.geeksforgeeks.org/beep-function-in-c-with-examples/

6) Reading and writing using binary files
https://study.com/academy/lesson/writing-reading-binary-files-in-c-programming.html

11) Extra Features Added:

- **Medium Computer Mode**
  Made by an algorithm we developed, we call it the Rowan-Squared Algorithm,
  It focuses on beating player through blocking his way of connecting 4s either vertically or horizontally
  So, our priority in this algorithm is blocking 4s, not building our 4s
  We wanted to develop another one “Hard Computer Mode” that will cover blocking diagonally, and puts the connecting 4s into priority.

- **Saving One Game in Easy Computer Mode**
  The assignment required was allowing the user to save up to 3 “Human vs Human” games, we added a new feature to save one easy computer mode.
  With the allowance of user to apply changes and save the game multiple times in advance.

- **Handling spacing errors which corrupt XML file**
  (Covered in Corner Cases pdf)