CS121: Computer Programming 1 – Fall 2022

Final Project Report (Due 31/12/2022)

Game Of Connect Four

Contributors:

1) **Name:** Karene Antoine Nassif Guirguis **ID:** 21010969
2) **Name:** Andrew Achraf Lamei Labib **ID:** 21010313

GitHub Link:

https://github.com/AndrewAchraf/Fall2022_CSE-121_Final-Project_Connect-Four.git

Game Description:

Our Connect 4 program is a two-players game in which the players alternately drop their different colored shapes into a grid by choosing the column's number. The shapes occupy the next available space within the column. The objective of the game is to connect-four of one's own same color shape next to each other vertically, horizontally, or diagonally. The two players keep playing until the board is full. The winner is the player having greater number of connected-fours.

Features:

- **Board User Interface:** The board may be of any size, the player can change the board's height and width by editing in the XML file choosing the number of rows and columns.
- **Drop shape in grid:** The player inputs his colored shape by choosing the column number desired and if there is an available space in that column, this shape appears in the bottom or above the last displayed one.
- **Game Mode:** Our Connect 4 program allows the player to choose the game mode: either vs another human player or vs computer.

- **Save and Load:** The game can be saved at any time to be loaded and continued whenever needed.
- **Undo and Redo:** In his turn, the player has the right to undo all played moves and make new ones OR redo them.
- **Top Players:** High Scores are updated at the end of each game and the player has access to view the number of top players he chooses.
- **Winning Announcement:** Once the board is full, the player having higher score is announced to be Winner and is asked if he wants to play again or to return to the main menu.
- **Draw Announcement:** Once the board is full and the two players' scores are equal, Draw is announced and the players are asked if they want to play again or to return to the main menu.
- **High Scores List:** The game stores the score of each game-winner in a *.bin* file and displays them in a sorted way when accessed. If the winner had played the game before, the game updates his score if it is greater than his high score.

Assumptions:

- To let the user be able to undo and redo, we assumed that he won't enter in the XML file more than 100 rows and 100 columns.

- If the player's new score is less than his high score, this new score will be neglected. But if it is higher than his high score, the high score will be updated to the new greater one.

- In the case of new player his score will not be considered as a new personal high score, so the new high score list won't be printed.

- When a player chooses the (exit) option, the game prints an exiting screen and then the program execution stops.

- When a player chooses the (save & exit), the game saves his progress into the file of his choosing, prints an exiting screen and then the program execution stops.

User Manual:

This game consists of a board with horizontal rows and vertical columns. You have access to choose the height and width of the board by editing in the XML file.

One player's pieces are 'x' shaped red colored, the other's are 'o' shaped yellow colored. You can input your colored shape by choosing the column's number desired and if there is an available space in that column, your shape appears in the bottom or above the last displayed one. In this way the columns start to fill up.

The objective of the game is to connect-four of one's own same color shape next to each other vertically, horizontally, or diagonally. The two players keep playing until the board is full. The winner is the player having greater number of connected-fours.

Press 1 to Start a new game then Press 1 to choose one player or Press 2 to choose two players or Press 3 to go back.

Press 2 to Continue a previous saved game

Press 3 to View the top players

Press 4 to Quit the game

Data Structure:

- **Structures used in the program:**

  - ***Player*** struct: to store the data of the players

  - ***Time*** struct: to store all data related to time

  - ***State*** struct: to store the updates after each move

  - ***Configuration*** struct: to store the game's configuration parameters: width, height, and max highscores

- **Arrays used in the program:**

  -***board*** array: Stores the board saved in the *state* struct to be reverted to in undo and redo

  -***array*** array: Stores the current board

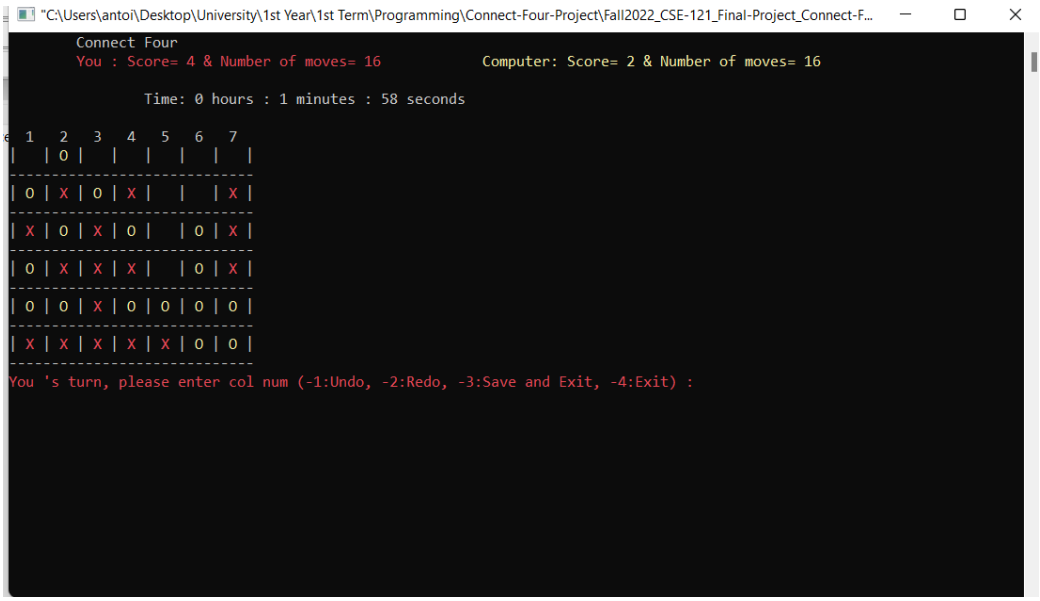
Used functions:

- void ***clearBoard***(): Empties the board array from all garbage values
- void ***drawBoard***(): Prints the board and other information on the screen
- void ***red***(): Makes the next outputs of the console colored in red
- void ***yellow***(): Makes the next outputs of the console colored in yellow
- void ***reset***(): Restores the text output color to the regular white
- void ***printNamesAndScores***(): Prints the name of the player along with his score
- void ***playerVsComputer***(): Launches a player vs computer game
- void ***playerVsPlayer***(): Launches a player vs player game
- void ***takePlayerTurn***(): Takes and processes the player's input
- void ***takeComputerTurn***(): Generates the next move for the computer
- int ***checkIfValidInput***(): Checks whether the user's input is valid or not
- void ***checkScores***(): Changes the score based on the last entry to the board
- bool ***checkForFreeSlots***(): Checks whether there are still empty slots in the board so that the game may continue
- void ***printWinnerPlayerVsComputer***(): Prints whether the player has won, lost or drawn against the computer
- void ***printWinnerPlayerVsPlayer***(): Prints the game winner (or draw) in case of the two player mode
- void ***saveState***(): Saves the current state of the game in the timeline/stack array
- void ***undo***(): Undo the last move
- void ***redo***(): Redo one move
- bool ***saveGame***(): Saves the current game to a user chosen file
- State ****loadGame***(): Loads a game from a user chosen file
- void ***xml***(): Reads the xml file and determine the width,height and highscores
- void ***convertToUppercase***(): Converts the player name to all upper case characters in order to save it in the highscores
- bool ***updateHighScores***(): Updates the high score list with the last game winner's name

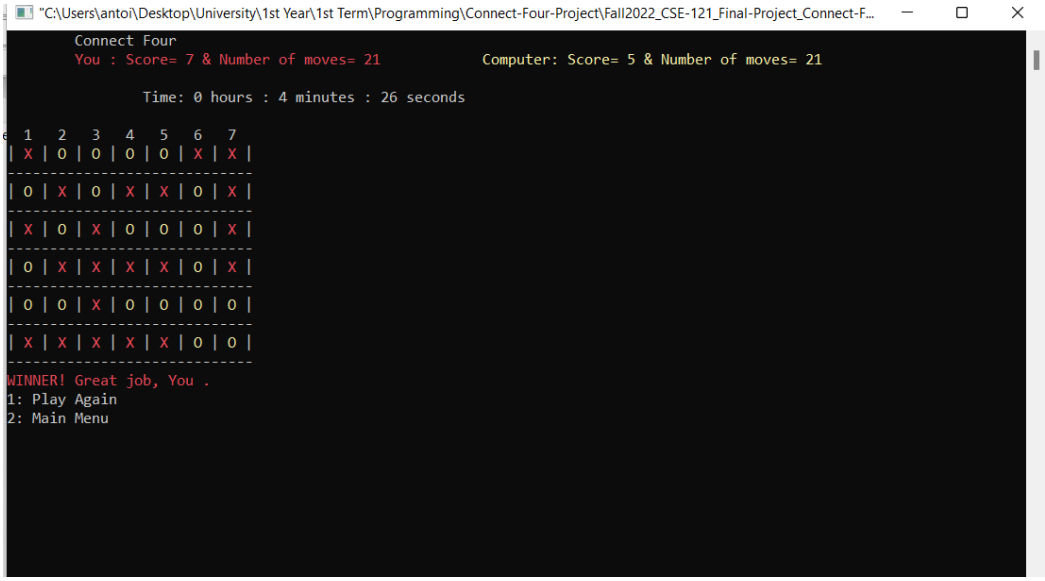
- int ***sortHighScores***(): Sorts the high score list and returns the current winner's rank
- void ***displayHighscores***(): Prints the high scores list

Sample runs:

- While Playing:



- Winning:



-

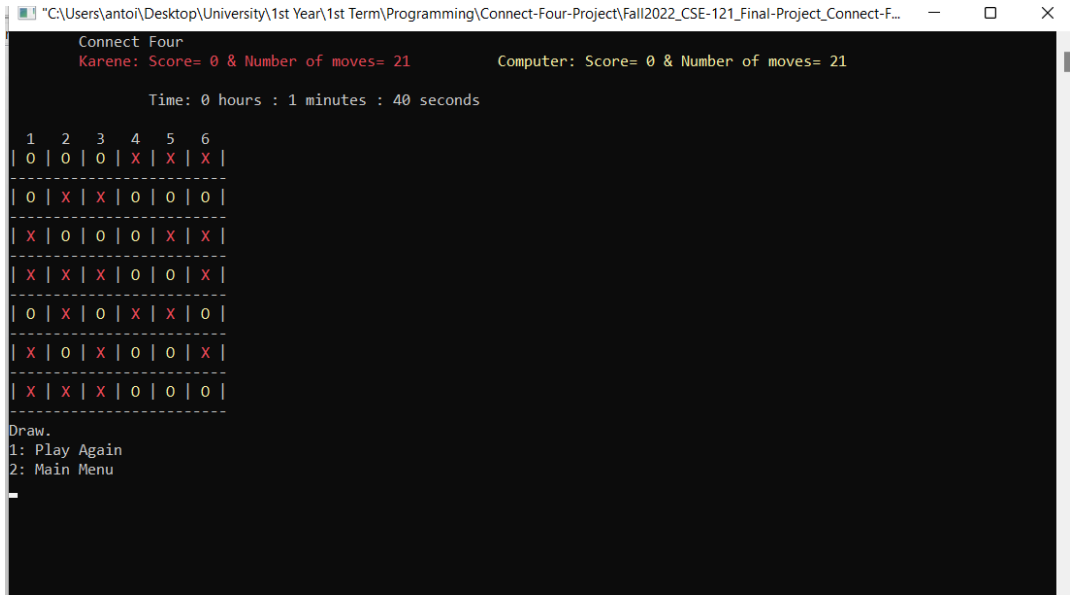
- Player vs Player:

```
C:\Users\antoi\Desktop\University\1st Year\1st Term\Programming\Connect-Four-Project\Fall2022_CSE-121_Final-Project_Connect-F...
        Connect Four
        Karene : Score= 4 & Number of moves= 21        Andrew: Score= 5 & Number of moves= 21

                Time: 0 hours : 5 minutes : 12 seconds

  1   2   3   4   5   6
| X | X | X | O | X | O |
------------------------
| O | X | O | O | O | X |
------------------------
| X | O | O | X | O | O |
------------------------
| O | O | O | X | X | X |
------------------------
| O | O | X | O | X | X |
------------------------
| X | X | X | X | X | X |
------------------------
| X | O | O | O | O | O |
------------------------
Andrew Wins! Congrats !
1: Play Again
2: Main Menu
```

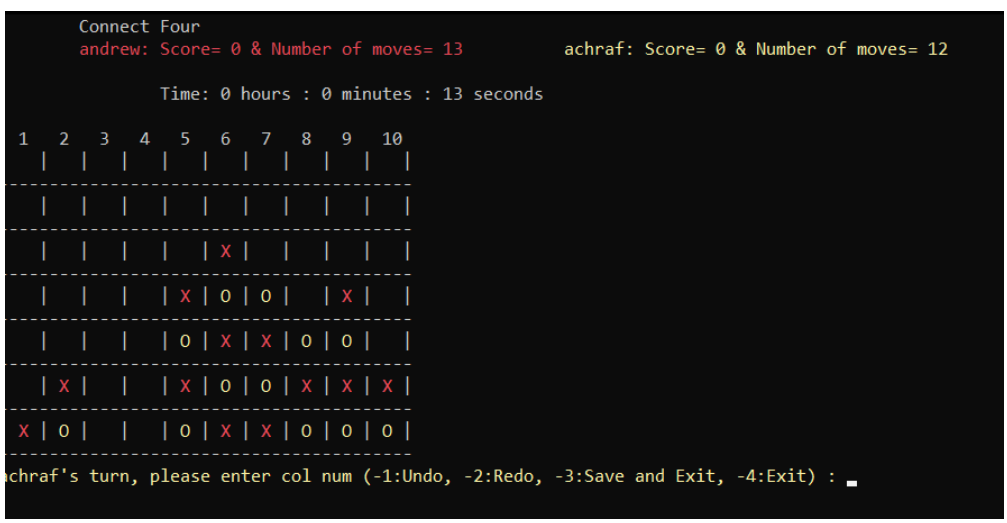
- Losing:

```
"C:\Users\antoi\Desktop\University\1st Year\1st Term\Programming\Connect-Four-Project\Fall2022_CSE-121_Final-Project_Connec
        Connect Four
        Karene: Score= 1 & Number of moves= 21          Computer: Score= 8 & Number of moves= 21

                Time: 0 hours : 2 minutes : 0 seconds

  1   2   3   4   5   6
| X | O | X | X | O | X |
------------------------
| X | X | O | O | X | O |
------------------------
| X | X | X | O | O | X |
------------------------
| X | O | O | O | O | O |
------------------------
| O | X | X | O | X | O |
------------------------
| X | O | X | O | X | O |
------------------------
| X | X | X | O | O | O |
------------------------
You Lost. Hard luck. Have another try.
1: Play Again
2: Main Menu
```

- <u>Draw:</u>



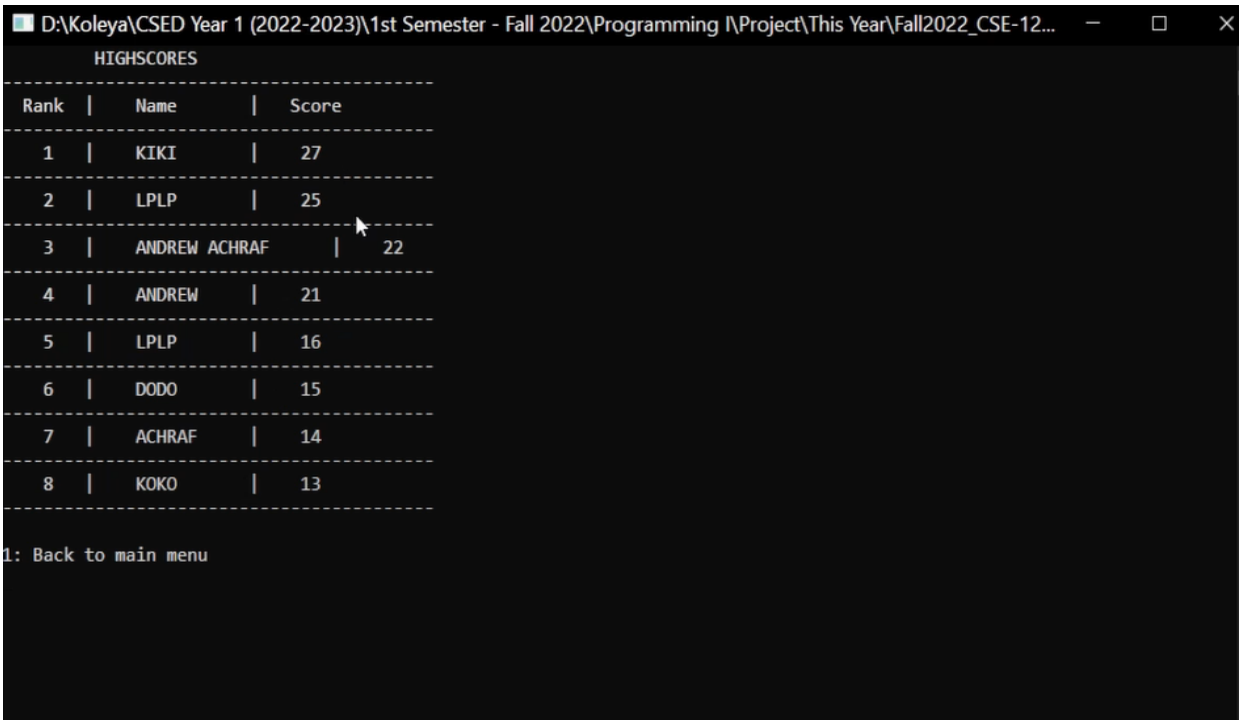
- <u>Undo and Redo:</u>



```
        Connect Four
        andrew: Score= 0 & Number of moves= 8          achraf: Score= 0 & Number of moves= 7

                Time: 0 hours : 0 minutes : 39 seconds

 1   2   3   4   5   6   7   8   9   10
|   |   |   |   |   |   |   |   |   |
-----------------------------------------
|   |   |   |   |   |   |   |   |   |
-----------------------------------------
|   |   |   |   |   |   |   |   |   |
-----------------------------------------
|   |   |   |   |   | O |   |   |   |
-----------------------------------------
|   |   |   |   | X | X | O |   |   |
-----------------------------------------
|   | X |   |   |   | O | O | X |   | X |
-----------------------------------------
| X | O |   |   |   | X | X | O |   | O |
-----------------------------------------
achraf's turn, please enter col num (-1:Undo, -2:Redo, -3:Save and Exit, -4:Exit) :
```

```
D:\Koleya\CSED Year 1 (2022-2023)\1st Semester - Fall 2022\Programming 1\Project\This Year\Fall2022_CSE-12...

        Connect Four
        andrew: Score= 0 & Number of moves= 13          achraf: Score= 0 & Number of moves= 12

                Time: 0 hours : 1 minutes : 4 seconds

 1   2   3   4   5   6   7   8   9   10
|   |   |   |   |   |   |   |   |   |
-----------------------------------------
|   |   |   |   |   |   |   |   |   |
-----------------------------------------
|   |   |   |   |   | X |   |   |   |
-----------------------------------------
|   |   |   |   | X | O | O |   | X |
-----------------------------------------
|   |   |   |   | O | X | X | O | O |
-----------------------------------------
|   | X |   |   | X | O | O | X | X | X |
-----------------------------------------
| X | O |   |   | O | X | X | O | O | O |
-----------------------------------------
achraf's turn, please enter col num (-1:Undo, -2:Redo, -3:Save and Exit, -4:Exit) : -2
Nothing To Redo !
achraf's turn, please enter col num (-1:Undo, -2:Redo, -3:Save and Exit, -4:Exit) :
```

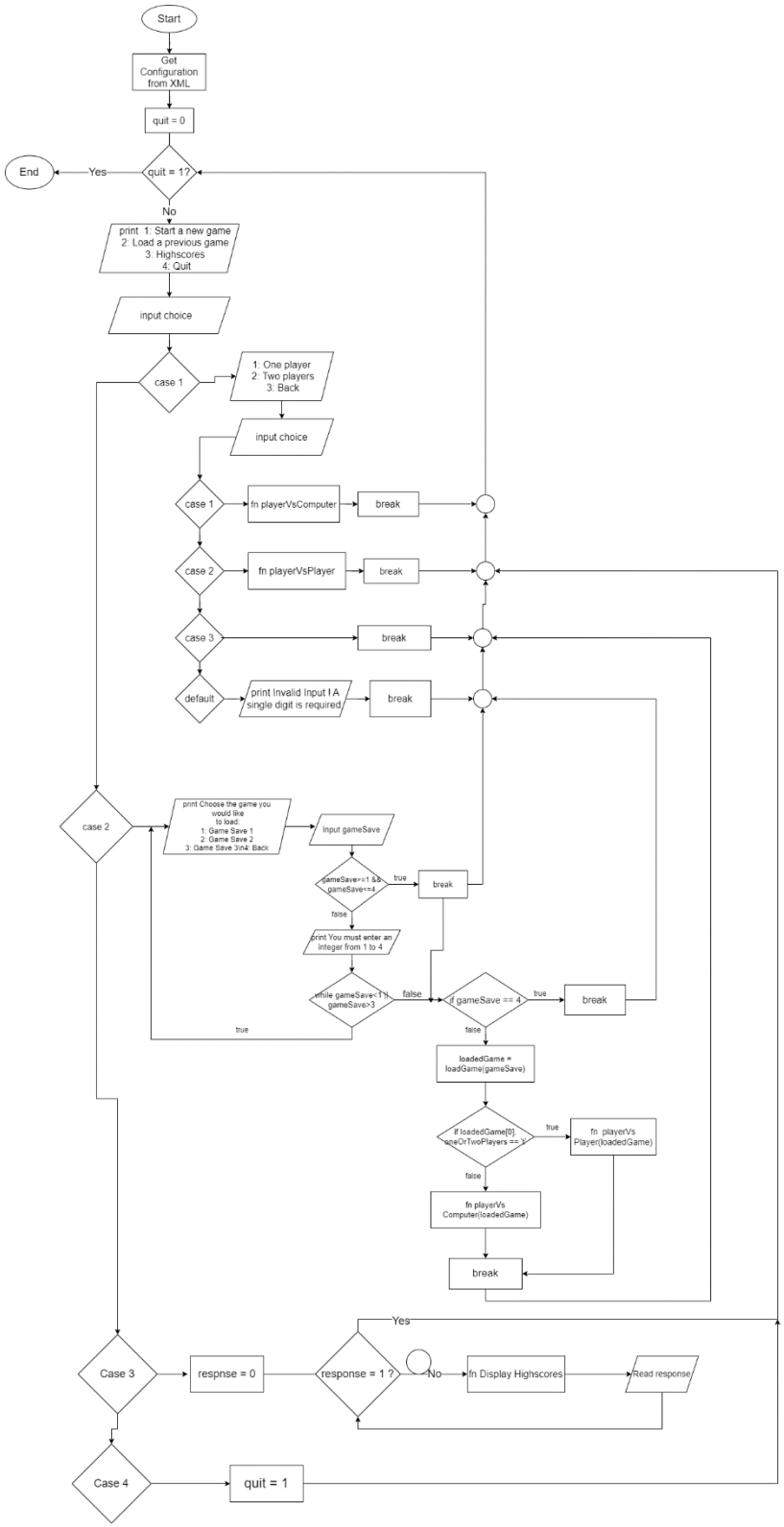
- <u>Achieving a new personal high score:</u>



```
D:\Koleya\CSED Year 1 (2022-2023)\1st Semester - Fall 2022\Programming I\Project\This Year\Fall2022_CSE-121_Final-Project_Connect-Four\Main Game.exe

        Connect Four
        andrew: Score= 24 & Number of moves= 35         Computer: Score= 16 & Number of moves= 35

                Time: 0 hours : 0 minutes : 24 seconds

 1   2   3   4   5   6   7   8   9   10
| O | O | O | O | X | X | O | X | X | X |
-----------------------------------------
| O | O | O | O | X | O | X | X | O | X |
-----------------------------------------
| O | X | O | O | X | X | O | X | X | X |
-----------------------------------------
| O | O | O | O | X | X | X | X | X | X |
-----------------------------------------
| O | X | O | O | O | X | O | X | O | X |
-----------------------------------------
| X | O | O | X | X | O | X | O | O | X |
-----------------------------------------
| X | X | O | X | X | O | O | O | O | X |
-----------------------------------------
INNER! Great job, andrew ,you won with score 24.
core Saved successfully

NDREW's rank : 3
ew personal highscore achieved !
ongratulations, ANDREW, take a look at your new rank !

        HIGHSCORES
-----------------------------------------
 Rank  |    Name       |   Score
-----------------------------------------
   1   |    KIKI       |    27
-----------------------------------------
   2   |    LPLP       |    25
-----------------------------------------
   3   |    ANDREW     |    24
-----------------------------------------
   4   |    ANDREW ACHRAF      |    22
-----------------------------------------
   5   |    LPLP       |    16
-----------------------------------------
   6   |    DODO       |    15
-----------------------------------------
   7   |    ACHRAF     |    14
-----------------------------------------
   8   |    KOKO       |    13
-----------------------------------------

: Play Again
```

Essential Flowcharts:

1) Main Menu:



References:

- **Writing, reading, and sorting binary files tutorials:**
  - https://youtu.be/P-fWNCF7Wx8
  - https://youtu.be/dDjfaA9Q3n8
  - https://www.youtube.com/watch?v=O6U9lZJqdeo
  - https://www.youtube.com/watch?v=8RkVs_B28mM
  - https://www.youtube.com/watch?v=QJKtAUWmYdE&t=2s
  - https://www.youtube.com/watch?v=VOpjAHCee7c&t=885s
  - https://www.youtube.com/watch?v=A4sRhuGkRb0&t=667s
  - https://youtu.be/1lNL31q4T5I

- **Useful functions we learned about:**
  - ***fopen():*** *https://www.geeksforgeeks.org/c-fopen-function-with-examples/*
  - ***fwrite():****https://www.tutorialspoint.com/c_standard_library/c_function_fwrite.htm*
  - ***toupper():*** *https://www.geeksforgeeks.org/toupper-function-in-c/*
  - ***strcmp():*** *https://www.scaler.com/topics/c/string-comparison-in-c/*
  - ***strcpy():*** *https://www.programiz.com/c-programming/library-function/string.h/strcpy*
  - ***fputs():*** *https://www.tutorialspoint.com/c_standard_library/c_function_fputs.htm*
  - ***fputc():*** *https://www.tutorialspoint.com/c_standard_library/c_function_fputc.htm*