CS201 – Fall 2022-2023
Homework 3
– CS201 Course – Tic-Tac-Toe –
Due November 25th, Friday, 23:55 (Sharp Deadline)

Introduction

The aim of this homework is to practice on functions, nested if-else statements, for-while loops, arrays/vectors and matrices.

The task in this homework will be to write code for a Tic-Tac-Toe game to play with your family and friends. For more information about Tic-Tac-Toe:
https://en.wikipedia.org/wiki/Tic-tac-toe and https://playtictactoe.org/

Game Interface & Rules

First, the program is going to take the names of two players and then the players are going to play tic-tac-toe twice: **the first player starts the first game and the second player starts the second game.**

Players are going to enter the moves in the following format: example: 1-1, 2-3, 1-2, … etc. The first number indicates rows from 1 to 3 (from up to down) and the second number indicates columns from 1 to 3 (from left to right). The program must check if the user enters a valid format or not. The numbers can only be 1, 2 or 3 and there must be a “-” (minus) between the numbers.

A game can be won by player1 or player2 or ends in a draw. At the end of the game, the program shows the result of the game: either player1 or player2 wins (1-0 or 0-1) OR the game ends in a draw (0-0), where none of the players wins.

The Program is going to keep scores of who won how many games and then at the end the program is going to show the **final scores of players** and determine an overall winner of the two games. If scores are equal (0-0 or 1-1) then the games will end in a **draw (0-0)**.

The format of the inputs and how to process them are described below in detail.

You **should** write user-defined functions to implement your program, you can **not** use only the *main* function. The user-defined functions that you are *suggested* to implement are also described below in detail.

In this homework, you have to work with a matrix to keep track of the tic-tac-toe board. A matrix can be created as a 2-dimensional array as follows:

```
string tictactoeBoard[3][3];
```

Then you need to initialize all elements of this array with an empty string.

Given a board with `tictactoeBoard[2][1]="O"` as shown below:

```
   |   |
-----------
   |   |
-----------
   | O |
```

The following code

```
tictactoeBoard[0][2] = "X";
```

will update the above board to the following board shown below:

```
   |   | X
-----------
   |   |
-----------
   | O |
```

Inputs, Flow of the Program, Functions and Outputs

The program to be implemented should have five(5) functions, and one(1) main function:

1. ***Main*****:** Main program will first greet the user with a welcome message and ask their names. Then it will call the ***playGame*** function twice: first for player1 and then for player2. Main program also keeps track of which player won the games and decides on the overall winner. *Please see sample runs.*

2. ***playGame*****:** This void function plays a Tic-Tac-Toe game starting with any player (player1 or player2) and continues until the game ends (one of the players wins or it is a draw) and returns the scores (0-0, 0-1 or 1-0). This

function should take 6 parameters: the name of the first player, the sign of the first player ("X" or "O"), the name of the second player, the sign of the second player ("X" or "O"), the score of the first player and the score of the second player. From the main program you will call this function twice by interchanging the arguments as follows:

```
// player1 with symbol "X" starts the game
playGame(p1Name, "X", p2Name, "O", p1Score, p2Score);
// player2 with symbol "O" starts the game
playGame(p2Name, "O", p1Name, "X", p2Score, p1Score);
```

Then in a loop until the game is over, you will call the ***playerMove*** function for each player in turns. At the end of this function, you will call the ***checkWinner*** function twice for each player symbol ("X" or "O") to see if there is a winner or not.

3. ***playerMove:*** A void function that takes 3 parameters: the name, the symbol ("X" or "O") of the player and the current TicTacToe matrix and then makes a valid move for the given player. This function reads the move string from the user and calls the ***checkFormat*** function to validate the string input. If the move input is valid, it then checks whether the board has that location empty or not. It displays an error message if the location is full. Otherwise, it modifies the TicTacToe matrix with the current move of the given player.
4. ***checkFormat*:** A **bool** returning function that takes a string parameter for the move and checks if it is in the correct format or not.
5. ***checkWinner*:** A **bool** returning function that takes the current TicTacToe matrix and the symbol used by the player ("X" or "O") and checks if the game has a winner or not.
6. ***print*:** A void function that takes the current TicTacToe matrix as the only parameter and prints it in a visual way as shown in sample runs.

Additionally, you will decide how to integrate these functions and trace the points earned by the user to complete your solution.

Please refer to the "Sample Runs" section for some examples and further details.

IMPORTANT!

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_HWnumber.cpp" (all lowercase letters). Please check the submission procedures of the homework, which are listed at the end of this document.

You should submit all of your files to SUCourse **without zipping** them.

VERY IMPORTANT!

Your homework will be automatically graded using **GradeChecker**, so it is very important to satisfy the exact same output given in the sample runs. You use (http://learnt.sabanciuniv.edu/GradeChecker/) GradeChecker to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all the files in your homework project. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

If your code does not compile, then you will get **zero**. Please be careful about this and double check your code before submission.

Your programs will be compiled, executed and evaluated **automatically**; therefore you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the mentioned format.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some points in the best scenario.

See the General Guidelines below, about homework submissions and how to get help etc.

Sample Runs

Below, we provide some sample runs of the program you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

```
Welcome to Tic-Tac-Toe!
Player 1, please enter your name: Beyza
Hello, Beyza. You are going to play with X
Player 2, please enter your name: Ekin
Hello, Ekin. You are going to play with O

Starting a new game...
Current board:
   |   |
-----------
   |   |
-----------
   |   |
Beyza, please enter your move: 2*1
Please enter correct move format!
Beyza, please enter your move: 2-1
Current board:
   |   |
-----------
 X |   |
-----------
   |   |
Ekin, please enter your move: 2-2
Current board:
   |   |
-----------
 X | O |
-----------
   |   |
Beyza, please enter your move: 2-2
That location is already full!
Beyza, please enter your move: 2-3
```

```
Current board:
   |   |
-----------
 X | O | X
-----------
   |   |
Ekin, please enter your move: 1-3
Current board:
   |   | O
-----------
 X | O | X
-----------
   |   |
Beyza, please enter your move: 3-1
Current board:
   |   | O
-----------
 X | O | X
-----------
 X |   |
Ekin, please enter your move: 1-1
Current board:
 O |   | O
-----------
 X | O | X
-----------
 X |   |
Beyza, please enter your move: 1-2
Current board:
 O | X | O
-----------
 X | O | X
-----------
 X |   |
Ekin, please enter your move: 3-3
Ekin is the winner!
Current Scores:
Beyza: 0   Ekin: 1
```

```
Starting a new game...
Current board:
   |   |
-----------
   |   |
-----------
   |   |
Ekin, please enter your move: 1-1
Current board:
 O |   |
-----------
   |   |
-----------
   |   |
Beyza, please enter your move: 2-3
Current board:
 O |   |
-----------
   |   | X
-----------
   |   |
Ekin, please enter your move: 2-1
Current board:
 O |   |
-----------
 O |   | X
-----------
   |   |
Beyza, please enter your move: 1-3
Current board:
 O |   | X
-----------
 O |   | X
-----------
   |   |
Ekin, please enter your move: 2-2
Current board:
 O |   | X
-----------
 O | O | X
-----------
   |   |
Beyza, please enter your move: 3-3
Beyza is the winner!
Current Scores:
Beyza: 1   Ekin: 1

Final Scores:
Beyza: 1   Ekin: 1
The overall game ended in a draw!
```

Sample Run 2

```
Welcome to Tic-Tac-Toe!
Player 1, please enter your name: Stark
Hello, Stark. You are going to play with X
Player 2, please enter your name: Rogers
Hello, Rogers. You are going to play with O

Starting a new game...
Current board:
   |   |
-----------
   |   |
-----------
   |   |
Stark, please enter your move: 1-1
Current board:
 X |   |
-----------
   |   |
-----------
   |   |
Rogers, please enter your move: 3-3
Current board:
 X |   |
-----------
   |   |
-----------
   |   | O
Stark, please enter your move: 2-2
Current board:
 X |   |
-----------
   | X |
-----------
   |   | O
Rogers, please enter your move: 1-3
Current board:
 X |   | O
-----------
   | X |
-----------
   |   | O
Stark, please enter your move: 4-1
Please enter correct move format!
Stark, please enter your move: 2-2
That location is already full!
Stark, please enter your move: 1-2
Current board:
 X | X | O
-----------
   | X |
-----------
   |   | O
Rogers, please enter your move: 3-2
Current board:
 X | X | O
-----------
   | X |
-----------
   | O | O
Stark, please enter your move: 1*3
Please enter correct move format!
Stark, please enter your move: 3-1
Current board:
 X | X | O
-----------
   | X |
-----------
 X | O | O
Rogers, please enter your move: 2-1
```

```
Current board:
 X | X | O
-----------
 O | X |
-----------
 X | O | O
Stark, please enter your move: 2-3
The game ended in a draw!
Current Scores:
Stark: 0   Rogers: 0

Starting a new game...
Current board:
   |   |
-----------
   |   |
-----------
   |   |
Rogers, please enter your move: 1-1
Current board:
 O |   |
-----------
   |   |
-----------
   |   |
Stark, please enter your move: 3-3
Current board:
 O |   |
-----------
   |   |
-----------
   |   | X
Rogers, please enter your move: 1-2
Current board:
 O | O |
-----------
   |   |
-----------
   |   | X
Stark, please enter your move: 1-3
```

```
Current board:
 O | O | X
-----------
   |   |
-----------
   |   | X
Rogers, please enter your move: 2-1
Current board:
 O | O | X
-----------
 O |   |
-----------
   |   | X
Stark, please enter your move: 2-3
Stark is the winner!
Current Scores:
Stark: 1   Rogers: 0

Final Scores:
Stark: 1   Rogers: 0
Stark is the overall winner!
```

Sample Run 3

```
Welcome to Tic-Tac-Toe!
Player 1, please enter your name: Larry
Hello, Larry. You are going to play with X
Player 2, please enter your name: Kramer
Hello, Kramer. You are going to play with O

Starting a new game...
Current board:
   |   |
-----------
   |   |
-----------
   |   |
Larry, please enter your move: 1-1
Current board:
 X |   |
-----------
   |   |
-----------
   |   |
Kramer, please enter your move: 3-3
Current board:
 X |   |
-----------
   |   |
-----------
   |   | O
Larry, please enter your move: 1-2
Current board:
 X | X |
-----------
   |   |
-----------
   |   | O
Kramer, please enter your move: 2-2
Current board:
 X | X |
-----------
   | O |
-----------
   |   | O
Larry, please enter your move: 2-1
Current board:
 X | X |
-----------
 X | O |
-----------
   |   | O
Kramer, please enter your move: 3-1
Current board:
 X | X |
-----------
 X | O |
-----------
 O |   | O
```

```
Larry, please enter your move: 3-2
Current board:
 X | X |   
-----------
 X | O |   
-----------
 O | X | O
Kramer, please enter your move: 1-3
Kramer is the winner!
Current Scores:
Larry: 0   Kramer: 1

Starting a new game...
Current board:
   |   |   
-----------
   |   |   
-----------
   |   |   
Kramer, please enter your move: 2-1
Current board:
   |   |   
-----------
 O |   |   
-----------
   |   |   
Larry, please enter your move: 3-3
Current board:
   |   |   
-----------
 O |   |   
-----------
   |   | X
Kramer, please enter your move: 2-3
Current board:
   |   |   
-----------
 O |   | O
-----------
   |   | X
```

```
Larry, please enter your move: 2-2
Current board:
   |   |
-----------
 O | X | O
-----------
   |   | X
Kramer, please enter your move: 1-1
Current board:
 O |   |
-----------
 O | X | O
-----------
   |   | X
Larry, please enter your move: 3-2
Current board:
 O |   |
-----------
 O | X | O
-----------
   | X | X
Kramer, please enter your move: 3-1
Kramer is the winner!
Current Scores:
Larry: 0   Kramer: 2

Final Scores:
Larry: 0   Kramer: 2
Kramer is the overall winner!
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homework unless otherwise noted.

How to get help?

You can use GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs can be found at SUCourse.

What and Where to Submit

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using XCode (macOS users).

You should also write your name and last name inside the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "Gülşen Demiröz", and if you want to write it as comment; then you must type it as follows:

// Gulsen Demiroz

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:
  - Use only English alphabet letters, digits, dot ('.') or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
  - Name your cpp file that contains your program as follows:
    "SUCourseUsername_hwnumber.cpp"
  - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**) except dot ('.'). For example, if your SUCourse username is "**gulsend**", then the file name should be: **gulsend_hw3.*cpp*** (please only use lowercase letters).
  - Do not add any other character or phrase to the file name.
- Please make sure that this file is the latest version of your homework program.
- Submit your work **through SUCourse only**! You can use GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).
- If you want to resubmit your work, you should first remove the existing file(s). This step is very important as if you don't delete the old files, we receive both files and the old one may be graded.

Grading, Review and Objections

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your homework, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- There is NO late submission. You need to submit your homework before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minute differences. You need to take this time difference into consideration.
- Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the Syllabus.

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Ekin Marlalı & E. Beyza Çandır & CS201 İnstructors