CIS 121 Computer Science I

Assignment Four

Due: November 23, 2021

There are two problems descripted below. Select one problem to complete for your Assignment Four.

Option 1: Airline Reservations System

A small airline has just purchased a computer for its new automated reservations system. You have been asked to develop the new system. You are to write an application to assign seats on each flight of the airline's only plane (capacity: 7 rows and 4 columns = 28 seats).

You have been asked to develop the new airline reservation system. You are to write an application to assign seats in an airline. Assume a small airplane with 28 seats (7 rows 4 wide) as follows:

```
1  A B   C D
2  A B   C D
3  A B   C D
4  A B   C D
5  A B   C D
6  A B   C D
```

The program should display the seat pattern, with an X marking the seats already assigned.

```
1  X B   C D
2  A X   C D
3  A B   C D
4  A B   X D
5  A B   C D
6  A B   C D
```

The program then requests a choice of seat. If the seat if available, then the seating display is updated. Otherwise, the program requests another choice of seat. If the user types in a seat that is already assigned, the program should say that that seat is occupied and ask for another choice.

This continues until all seats are filled or until the user says that the program should end.

Sample Output:

```
ECC Airlines
Seat Reservation Program….
Reserved seats are marked 'X'. Others are available.

A B   C D
A B   C D
A B   C D
A B   C D
A B   C D
A B   C D

ECC Airlines
Seat Reservation Program:
Reserved seats are marked 'X'. Others are available.
Please enter your request in the form "3 C" for Row 3, Seat C
There are 7 rows. Seats are A, B, C, D: 3D

A B   C D
A B   C D
A B   C X
A B   C D
A B   C D
A B   C D
N or n quits, anything else continues: y

ECC Airlines
Seat Reservation Program.
Reserved seats are marked 'X'. Others are available.
Please enter your request in the form "3 C" for Row 3, Seat C
There are 7 rows. Seats are A, B, C, D: 3D

****That seat is taken****.
****No assignment made****.
****Please make another request****

A B   C D
A B   C D
A B   C X
A B   C D
A B   C D
A B   C D
N or n quits, anything else continues: n

Press any key to continue
```

Option 2: Tic-Tac-Toe Game

In a game of TicTacToe, two players take turns marking an available cell in a 3 × 3 grid with their respective tokens (either X or O). When one player has placed three tokens in a horizontal, vertical, or diagonal row on the grid, the game is over and that player has won. A draw (no winner) occurs when all the cells on the grid have been filled with tokens and neither player has achieved a win.

Write a program that allows two players to play a game of tic-tac-toe. Use a two-dimensional char array with 3 rows and 3 columns as the game board. Each element of the array should be initialized with an asterisk (*). The program should display the initial board configuration and then start a loop that does the following:

- The program prompts the first player to enter an X , and then prompts the second player to enter an O token by entering a row and column number. Whenever a token is entered, the program refreshes the board and determines the status of the game (win, draw, or unfinished).
- To place a token, prompt the user to enter the row and the column for the token.

Allow player 1 to select a location on the board for an X by entering a row and column number. Then redisplay the board with an X replacing the * in the chosen location.

If there is no winner yet and the board is not yet full, allow player 2 to select a location on the board for an O by entering a row and column number. Then redisplay the board with an O replacing the * in the chosen location. The loop should continue until a player has won or a tie has occurred, then display a message indicating who won, or reporting that a tie occurred.

- Player 1 wins when there are three Xs in a row, a column, or a diagonal on the game board.
- Player 2 wins when there are three Os in a row, a column, or a diagonal on the game board.
- A tie occurs when all of the locations on the board are full, but there is no winner.

Input Validation*: Only allow legal moves to be entered. The row must be 1, 2, or 3. The column must be 1, 2, or 3. The (row, column) position entered must currently be empty (i.e., still have an asterisk in it).*

Sample Output:

```
        1 2 3
Row 1:  * * *
Row 2:  * * *
Row 3:  * * *
```

```
Player X's turn.
Enter a row and column to place an X.
Row: 2
Column: 2

      Columns
        1 2 3
Row 1:  * * *
Row 2:  * X *
Row 3:  * * *

Player O's turn.
Enter a row and column to place an O.
Row: 1
Column: 2

       Columns
        1 2 3
Row 1:  * O *
Row 2:  * X *
Row 3:  * * *

Player X's turn.
Enter a row and column to place an X.
Row: 1
Column: 1

        Columns
        1 2 3
Row 1:  X O *
Row 2:  * X *
Row 3:  * * *
```

Requirements:

- Use the “Steps for creating a program” template to develop your solution. When complete, save as a .txt, .doc or .rtf file and upload this with the other two required files.
- Use meaningful variable names and reasonable data types.
- Use the header format template. Include a comment section at the top of you file, with your name, the assignment number and a brief summary (in your own words) of what the program does.
- Include comments to describe the major sections of the program
- Prompt for each input value you need and annotate the output to describe what it represents.

Submit:

- You should submit the three files: the source code file, the output screen shot and a written report (filled “Steps for creating a program”)
- Submit the source codes to the DropBox on D2L before the deadline.