

COMP 322

Winter Semester 2020

INSTRUCTOR: DR. CHAD ZAMMAR
chad.zammar@mcgill.ca

Assignment 1: Exploring Functions and arrays.

Due date: 14 February 2020, 11:59 PM.

Before you start:

- Collaboration and research for similar problems on the internet are recommended. However, your submission should reflect individual work and personal effort.
- Some of the topics may not be covered in class due to our limited time. You are encouraged to find answers online. You can also reach to your instructor or TAs for guidance.
- Please submit your assignment before the due date to avoid penalties or worse risking your assignment being rejected.
- Submit one zip file called **assignment1** containing 2 code files:
 - one file called **functions.cpp** containing all the functions together with their implementations
 - and another file called **main.cpp** containing only the main() function.

Make sure your code is clear and readable. **Readability of your code as well as the quality of your comments will be graded.**

- No submission by email. Submit your work to mycourse.
 - If your code does not compile on the lab machines (Trottier) it will not be graded.
 - Be happy when working on your assignment, because a happy software developer has more inspiration than a sad one :).
-

Tic-Tac-Toe game is probably one of the easiest board games to implement. In this assignment you will be implementing one by following the instructions given below. You can check the Wikipedia page for a memory refresh about the rules and how to play: <https://en.m.wikipedia.org/wiki/Tic-tac-toe>

In this assignment we will be implementing a 3D version of Tic-Tac-Toe game. It has 3 boards of 3x3 cells each. To win a player must have a series of 3 consecutive cells marked with the same symbol either within the same board or across the 3 boards.

For example, player X will win if he/she manages to get three Xs on one line in the same table (any table):

X	X	X

Or across three tables:

X		

	X	

		X

Same thing for the diagonal:

X		
	X	
		X

Or across the three tables:

X		

	X	

		X

A player will also win if he/she manages to put an X in the same cell across the three tables:

X		

X		

X		

And so on ...

In the following questions you will be asked to write multiple functions doing each a specific task. The signature of each function will be provided and should be respected. You need to implement a suitable `main()` function that calls all the functions one by one and print their results to the screen.

Question 1 (5 pts)

Write a function that greet the user and display instructions about the game.

Signature of the function: **void greetAndInstruct();**

When running the function, the output should be similar to the following example:

Hello and welcome to the Tic-Tac-Toe challenge: Player against Computer.

The board is numbered from 1 to 27 as per the following:

1 2 3	10 11 12	19 20 21
-----	-----	-----
4 5 6	13 14 15	22 23 24
-----	-----	-----
7 8 9	16 17 18	25 26 27

Player starts first. Simply input the number of the cell you want to occupy. Player's move is marked with X. Computer's move is marked with O.

Start? (y/n):

If you press "n" for no, the program exits. If you press "y" for yes, the game starts.

Question 2 (10 pts)

Write a function called **displayBoard** that prints out to the screen the tic-tac-toe board reflecting the current status of the game. For a new game, the board will be empty except for the cell numbers. After each move the displayBoard function should reflect the new status **by correctly displaying Xs and Os where appropriate.**

Signature of the function: **void displayBoard (char board[]);**

Question 3 (40 pts: 20 points for each function)

After every move we need to check if the move is legal and if so we have to check whether we have a winner. Write the functions **checkIfLegal** and **checkWinner**.

Signature of the functions:

bool checkIfLegal (int cellNb, char board[]);

bool checkWinner(char board[]);

If the user inputs an illegal cell number for the move (illegal because it is out of range or because it is the number of an already occupied cell) the checkIfLegal function should warn the user about the move and the program should ask the user to input a different cell number.

If checkWinner function finds a winner, a message should be displayed to the screen identifying who won the game and the program **should exit.**

Remember that sometimes the board is full and no winner, this is called a tie, in this case the game is over, the program should announce a tie then exits.

Question 4 (30 pts)

Write the function that decides the move for the computer. The move should not be random, you need to check whether there is a move that would make the computer a winner, if such a move exists, then this is the move to take. If such a move does not exist, then check whether there is a move that would make the player a winner, if so you need to block this cell in order to block the opponent from winning.

Signature of the function: **void computerMove(char board[]);**

Question 5 (15 pts)

Write the main function that would contain the main game loop and the calls for the previous functions along with any necessary logic that is suitable to achieve your goal.