



Inspiring Excellence

Course Code:	CSE111
Course Title:	Programming Language II
Homework No:	02
Topic:	Functions
Submission Type:	Hard Copy (Only write down the functions that you are instructed to design. No need to write down the given codes.)
Resources:	Multidimensional list (Nested list): a. <a href="https://www.youtube.com/watch?v=UyA6g4hKpTU">https://www.youtube.com/watch?v=UyA6g4hKpTU</a> b. <a href="https://www.learnbyexample.org/python-nested-list/">https://www.learnbyexample.org/python-nested-list/</a>

# TIC TAC TOE

It's a classic paper-and-pencil game played on a 3x3 grid. The objective is to match three identical symbols (traditionally "X" or "O") in a row, either horizontally, vertically, or diagonally, while preventing your opponent from doing the same. Players take turns placing their symbols on the grid until one player wins or the game ends in a draw. A demo of the game is given at the end of the file.

Please find the google colab file containing the given code here:

<https://colab.research.google.com/drive/1WWvONfTPN604jrg8Er1ZB9WdD0mCrIFa?usp=sharing>

The file contains two functions named `runGame()` and `printCurrentBoard()`. **Understand the workflow of the `runGame()` function first.** Then you need to design the -

- `checkHorizontal()`
- `checkVertical()`
- `checkDiagonal()`
- `checkBoard()`, and
- `placeCharacterOnBoard(position)`

functions that are used in the given `runGame()` function based on your understanding of it, and **DO NOT CHANGE any given code except for the functions you are instructed to design.** Just run the program after completing all the above-mentioned functions. **The file also contains 5 global variables at the top of the program. You cannot declare any other global variables to develop the game.** Read the comments carefully to understand the purpose of a particular statement in the given code.

Each function has a defined workflow that it acts upon. Follow the game rules that are explained in the next part. On this note, the logical building of the functions completely depends on your own design sense.

## 1. `checkHorizontal()`

X	O	X
O	X	O
X	X	X

To determine if a player has won horizontally, you need to check if they have placed their symbol (X or O) in three consecutive cells on the same row. In the above grid, the player with the symbol "X" has won horizontally because he/she has three "X" symbols in a row on the bottom row. This function returns **True** if any of the three rows contain the same symbol (X or O) three times consecutively; otherwise, it returns **False**.

## 2. `checkVertical()`

X	O	X
X	O	O
X	X	O

To check the vertical condition in Tic Tac Toe, you need to verify if a player has placed the symbol (X or O) in three consecutive cells in the same column. The three cells must be aligned vertically. In this grid, the player with the symbol "X" has won vertically because the grid has three "X" symbols in a column,

specifically the leftmost column. This function returns **True** if any of the three columns contain the same symbol (X or O) three times consecutively; otherwise, it returns **False**.

### 3. `checkDiagonal()`

X	O	X
O	X	O
X	O	X

To check the diagonal condition in Tic Tac Toe, you must verify if a player has placed their symbol (X or O) in three consecutive cells along a diagonal line. To summarize, if you find a line of three matching symbols on either diagonal, with the symbols being adjacent and aligned diagonally, the player who placed those symbols is declared the winner. This function returns **True** if any of the diagonals contain the same symbol (X or O) three times consecutively; otherwise, it returns **False**.

### 4. `checkBoard()`

In Tic Tac Toe, each player takes turns placing their symbol (X or O) on the board until there is a winner or the game ends in a draw.

Here's a step-by-step process to check the board:

- Check for a horizontal win
- Check for a vertical win
- Check for a diagonal win

If any of the above conditions are met, then you have a winner, and the player with the matching symbols in the winning line is declared the winner. If there are no winners and the entire board is filled with symbols (X or O), then the game is a draw. If there are no winners and there are still empty cells on the board, the game is ongoing, and players can continue taking turns. This function returns either **True** or **False**.

### 5. `placeCharacterOnBoard(position)`

This function takes a particular position on the grid where the symbol 'X' or 'O' will be placed. For a successful move, this function returns 1; otherwise, the return value will be 0. This return value will be added to the `inputCount` counter mentioned in the `runGame()` function. Furthermore, if a player inputs a position that is already filled out of the range from 1 to 9, an “**invalid position**” message will be printed, and the player will be asked to enter a valid position again. You will be using a nested list to represent the board.

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

or

```
[[1,2,3],  
 [4,5,6],  
 [7,8,9]]
```

So, when a player wants to place an 'X' or 'O' symbol in the bottom-left corner, he/she needs to put the symbol in place of 7 in the nested list. The symbols 'X' and 'O' will be placed alternately as the players are taking turns, which is already written in the `runGame()` function.

The given `printCurrentBoard()` function will print the current status of the board in the particular format shown in the demo. You do not need to worry about it.

## DEMO TIC-TAC-TOE GAME:

You can download the demo game from here: [tic\\_tac\\_toe\\_demo.exe](#). It will help you understand what you are expected to do for the homework.

Follow the steps given below to successfully run the file:

1. Download the file.
2. Copy and paste it on the Desktop.
3. In the taskbar of your PC, you will find a **Search** bar.



4. Search for "**Command Prompt**" and select it. A black window will appear.
5. In the command prompt, write **cd Desktop** and press Enter.
6. Finally, write **tic\_tac\_toe\_demo.exe** and boom! Your game will start.