CSC 110 – Lab 4
Algorithms using Functions

Names: ____

Tic-Tac-Toe Game

Write a program that will allow two players to play Tic-Tac-Toe. You will be given some code as a starting point. Fill in the parts of the main algorithm and the functions to make the program work.

How it works

The game board is represented by a list where each position in the list represents a spot on the board:

```
   0  |  1  |  2
----------------
   3  |  4  |  5
----------------
   6  |  7  |  8
```

So we have the list:

```
bd = [0,1,2,3,4,5,6,7,8]
```

where the first three elements of the list (`bd[0], bd[1], bd[2]`) represent the first row of the board, the second three elements (`bd[3], bd[4], bd[5]`) represent the second row of the board. And the last three elements (`bd[0], bd[1], bd[2]`) represent the bottom row.

If you want to fill in a spot with an X or an O, you can replace the value in the list. So to put an X in the center square, you will use the following command:

```
bd[4] = ‘X’
```

Tic-Tac-Toe python code

All of the code you will need for this lab can be found in this file:

http://www.cs.uri.edu/~cingiser/csc110/labs/python/tictactoe_skeleton.py

For the exercises below, you can open this file in a text editor or in IDLE and copy and paste the parts you need for each exercise.

Get Started – Draw the board

To get started, create a Python code file with the `drawBoard` function code in it:

```
def drawBoard(board):
    # Draws the board using the list of numbers
    print(" ")
    print(" ",board[0]," | ",board[1]," | ", board[2])
    print("----------------")
    print(" ",board[3]," | ",board[4]," | ", board[5])
    print("----------------")
    print(" ", board[6]," | ",board[7]," | ", board[8])
    print(" ")
    return
```

Save the file as `draw.py` and then run the code.

Create the board with the following command in the Python Shell:

```
bd = [0,1,2,3,4,5,6,7,8]
```

Now draw the board with the command, also in the Python Shell:

```
drawBoard(bd)
```

In our game, this is considered an empty board because it has no X’s or O’s in it. It only has place holder numbers.

To place a symbol in the spot on the board labeled 3, type the following command:

```
bd[3] = ‘X’
```

Now draw the board again to see the new board.

Instructor Initials: ______________________________

Try a function:

The `nextMove` function asks a player to choose a spot on the board, replaces the spot on the board with the player’s symbol, and then returns the board as a result. Write this function by filling in this code:

```
def nextMove(turn,bd):
    # Ask the player to choose a spot on the board
    # Replace the spot on the board with the player’s symbol
    # The turn parameter represents the player’s symbol
    # Return the board
    return bd
```

After you write the code for this function, save it in the same file where you saved the `drawBoard` function and run the code.

In the Python Shell, create the board again:

```
bd = [0,1,2,3,4,5,6,7,8]
```

Now draw the board:

```
drawBoard(bd)
```

and then call for the next move:

```
bd = nextMove(‘X’, bd)
```

and draw the board again. This will allow you to test the function before you use it in the full program.

Instructor Initials: ______________________

Fill in the code

Now you can take the rest of the code from the tictactoe_empty.py file, along with the functions you have written and fill it in to create the program to play the game. Save your program as `tictactoe.py` and test it out.

The main code is given in the file, along with comments to describe how it works. You are to fill in the code for the functions.

Here are some hints about how to complete the rest of the functions.

`moreMoves`

This function is used to determine if there are any more spots on the board to be filled in. Think about what the board would look like if it was FULL. What kinds of values would it be full of?

Instructor Initials: ______________________________

`nextTurn`

This function switches whose turn it is. It is given the symbol for the player whose turn it is currently, and it should return the symbol for the other player.

Instructor Initials: ______________________________

`isWinner`

This function is used to determine if there is a winner. In tic tac toe, there is a winner when one of the players has taken three spots in a row, horizontally, vertically or diagonally. One way to implement this is to consider all of the possible ways that a player can have three in a row with this particular board representation. For example, if the first three values in the board are equal to each other:

```
if bd[0] == bd[1] and bd[1] == bd[2]
```

Then whichever symbol is in those positions is the winner, and that symbol should be returned by the function.

Instructor Initials: ______________________________

Challenge Problem

Modify the tic-tac-toe program above so that the computer plays as one of the players. If you can get it to just play, without any type of intelligence, just choosing a position on the board that is not already taken, you will get one extra point. If you can get it to play so that the computer player blocks a win by the human player, and tries to win, then you can get two extra points.

Instructor Initials: ______________________________