Instructions

Answer to question 1 in file a4q1.py, and question 2 in file a4q2Lib.py zip them in HW4-X#-StudNumber.zip and email it to the usual email address. (X#: D3; C4)

Do not forget to add comments in each program to describe it, the functionality of its functions and the type of parameters as well as the result.

Question 1 (8 points) Derive a function in Python called *add*, that takes an array and modifies it by adding 1 to all the elements. Create another function Python called *add_V2*, which takes an array and returns a new array containing the values of the given array provided as a parameter incremented by 1, without modifying it.

In the main part of the program, ask the user to input an array, call the function *add* and display the modified array. Then call the function *add_V2*, with the array as a parameter, and display its resulting array. Display the original array to check that it has not changed
.

Both functions must support any type of arrays (dimensions), even for lists that are not arrays (the inside lists are not necessarily the same size).

Examples :

```
Input the array elements with spaces between columns.
One row per line, and an empty line at the end.
1 2 3
4 5 6

The array is:
[[1, 2, 3], [4, 5, 6]]
After executing the function add, the array is:
[[2, 3, 4], [5, 6, 7]]
A new array created with add_V2:
[[3, 4, 5], [6, 7, 8]]
After executing the function add_V2, the initial array is:
[[2, 3, 4], [5, 6, 7]]
```

Question 2 (12 points)
It is time to play X-O. You should complete the program for this game.

Part of this program is provided to you in two files: d4q2.py (it is complete, no need to be modified) and d4q2Lib.py (you have to complete). The annex has examples of messages displayed during the game.

1) The main program controls the game.
a. It asks the user to start a game : if the response is not o or O, the program ends; If the response is o or O, it leads the game by using the following operations :
   i. erase the table (with function eraseTable(tab)).
   ii. display the table (with function displayble(tab)).
   iii. play a step (with function play(tab, player) – including the request made to the player for the new position).
   iv. verify if the player has won or if it is a draw (with function verifyWin(tab)).
   v. if the game is not completed, play again a step with the other player (from iii).
Note: the array of the game table is created in the main part of the program in a4q2.py and is passes as reference to other functions.
b. after each game, you are asked to start another game (restart from a.).
2) the function verifyWin call the following functions:
a. testRows(tab) – to check if a row has won.
b. testCols(tab) – to check if a column has won.
c. testDiags(tab) – to check if a diagonal has won.
d. testDraw(tab) – to check for a draw.

Your have to complete the following functions in the file a4q2Lib.py:
- eraseTable (tab)
- verifyWin(tab)
- testRows(tab)
- testCols(tab)
- testDiags(tab)
- testDraw(tab)

Notes:

1) The function verifyWin display the message “Draw” instead of “Player X has won!” or “Player O has won!” when there is a draw.
2) The new row and new column are given with input() and are stored in a list with two elements (the first is the row and the second the column).
3) The functions testRows, testCols, and testDiags, return one of the characters ‘-‘, ‘X’ or ‘O’.
   if ‘-‘ is returned, no one has won, otherwise the character for the winning player is returned.
4) The documentation of each function is available in the file a4q2Lib.py.

Annex – example of games

```
Start a game (O or N): O
  0 1 2
0 - - -
1 - - -
2 - - -
Player X, Please provide the row and column from 0 to 2:
Row: 1
Colomn: 1
  0 1 2
0 - - -
1 - X -
2 - - -
Player O, Please provide the row and column from 0 to 2:
Row: 0
Column: 0
  0 1 2
0 O - -
1 - X -
2 - - -
Player X, Please provide the row and column from 0 to 2:
Row: 1
Column: 5
Player X, Please provide the row and column from 0 to 2:
Row: 1
Column: 3
Player X, Please provide the row and column from 0 to 2:
Row: 2
Column: 2
  0 1 2
0 O - -
1 - X -
2 - - X
Player O, Please provide the row and column from 0 to 2:
Row: 0
Column: 0
The position 0 0 is occupied
Player O, Please provide the row and column from 0 to 2:
Row: 0
Column: 2
  0 1 2
0 O - O
1 - X -
2 - - X
Player X, Please provide the row and column from 0 to 2:
Row: 1
Column: 0
  0 1 2
0 O - O
1 X X -
2 - - X
Player O, Please provide the row and column from 0 to 2:
Row: 0
Column: 1
```

```
Player O has won!
  0 1 2
0 O O O
1 X X -
2 - - X
Start a game (O or N): N
```