PYTHON WEEK 3 ASSIGNMENT

1. Command Line Interface challenge

Using the CLI, create a plain text file (.txt) that contains the commands necessary to answer the following questions:

1. How do I create a hidden file or folder? How do I display it in the CLI?
2. How do I create multiple nested directories, like /c/Users/myusername/these/folders/are/just/for/fun?
3. How do I append a line to a file, without a newline character, so the output would be:

```
first line
second line
```

2. GIT exercises

The first exercise is to create a local repository using GIT and link it to a remote GitHub repository. Go through the following instructions:

- Getting Started with Git and GitHub

https://www.codecademy.com/articles/f1-u3-git-setup

This exercise is a mini-course of Git and GitHub. Go through it and try to code along:

- How to Use GIT and GitHub

https://eu.udacity.com/course/how-to-use-git-and-github--ud775

3. GIT exercise: animals repository

Tip: make use of the CLI to practice your GIT skills. This not only teaches you how GIT works, but also how to work like a real software developer!
In this homework you'll be working with GIT and GitHub. Follow the steps to learn how to create a remote repository and work with it from your local machine:

1. Create a repository on GitHub, called animals
2. Clone the repository to your local machine, using SSH
3. Locally, create a file called "zoo.txt". Include 3 animals found in a zoo
4. Add and commit the file to the local repository. Make sure the commit message is meaningful (ex. "created .txt file with animal names")
5. Push your commit to the remote repository, verify that it has worked on

GitHub

6. Go back to your local repository and create a branch called new-feature

Tip: in software, a "feature" is a technical term that points to any functionality that a user can derive benefit from. For example, Facebook has many features: the ability to make a profile, like a post, place comments, etc.

7. Inside the new branch, create a file called "pets.txt". Include 3 animals that could be a pet
8. Also, add 2 more animals to the "zoo.txt" file
9. Add and commit the file to the local repository. Again, make sure the commit message is meaningful
10. Push your commit to the remote repository, verify that it has worked on GitHub
11. On GitHub, find out how to merge branch new-feature into master
12. Merge the branches
13. Switch back to branch master
14. Pull the changes from your remote repository to your local repository, verify that everything worked

Ps: Don't be sad if you can't figure out steps 11 to 14. We didn't cover these steps during the class. We just want you to try and find solutions on your own. So if you fail, it is alright!

4. Rock, Paper, Scissors

You are to develop an "Rock, Paper, Scissors" game that is intended to be played between user and computer itself.

Winning Rules as follows :

Rock vs paper-> paper wins

Rock vs scissor-> Rock wins

paper vs scissor-> scissor wins.

Example Output :

```
Winning rules of the rock paper and scissors game as follows:

rock vs paper->paper wins

rock vs scissors->rock wins

paper vs scissors->scissors wins
```

```
Enter choice
 1. Rock
 2. paper
 3. scissor

User turn: 1
User choice is: Rock

Now its computer turn.......

computer choice is: paper
Rock V/s paper
paper wins =>computer wins
do you want to play again?
N
```

5. tic_tac_toe.py

As per our request, you are to develop an “XOX” game that is intended to be played between user and computer itself. Also, with the algorithm you create, you’re expected to make the computer play the game with reasonable moves rather than random moves.
For example, assume that the computer is using “O” for its moves.

```
X O _
_ X _
_ _ _
```

In this case, now it’s the computer’s turn and it must put an “O” to the bottom-right corner in order not to lose the game.

Different assumption:

O X X
O _ X
_ _ _

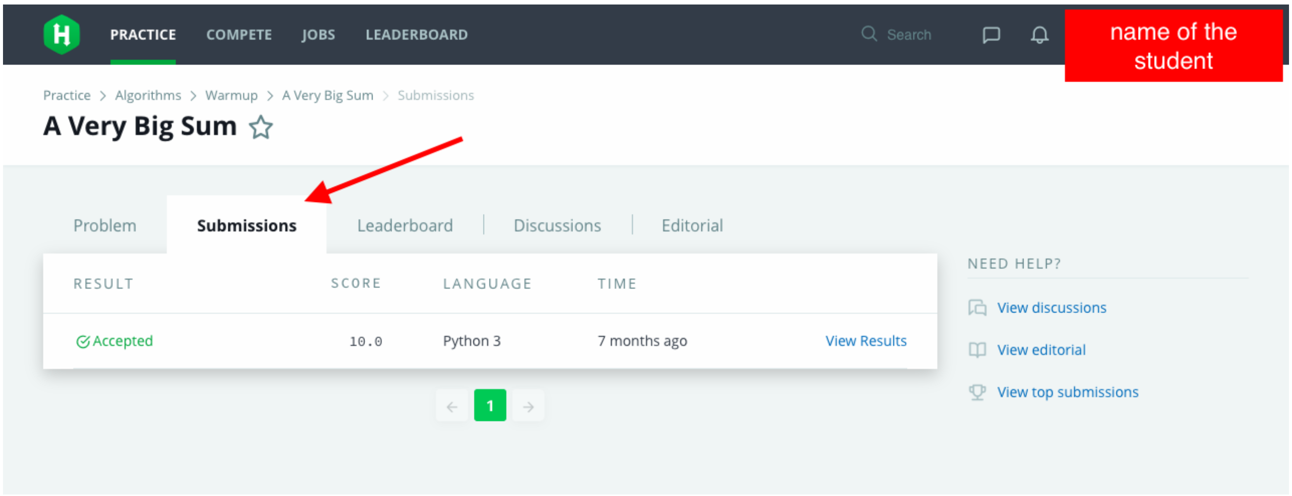
In a situation like this, computer must, in its turn, put an “O” to the bottom-left corner as the winning move.

HackerRank 01:

This is just an explanation. You can reach the HackerRack exercise from the other assignment of this week.

If your code passes all cases, you finish the exercise and you will be sure that your code properly works. The student needs to show that he/she did the assignment by taking a screenshot of the **submissions** sub-title like below and this screenshot needs to be attached to the google classroom assignment as an answer to prove that the student did the hackerrank exercises. It is important to see the name of the student in the screenshot below.

You have to submit a screenshot of your submission to the Google Classroom, like:



RESULT	SCORE	LANGUAGE	TIME	
✓Accepted	10.0	Python 3	7 months ago	View Results

Do not submit your screenshot of your code. We just want to see like above picture. Also you can copy your code to the py file and save with the name of the exercise in the HackerRank. You can hold your whole exercises, which you are going to do, in a GitHub repository. This is just an advice, not mandatory. We do not look for your github whether you uploaded hackerrank exercises codes.

RUBRIC

TASK	CRITERIA	POINTS
1		15
2		20
3		20
4		15
5		30
	TOTAL POINTS	100