Assignment 6 – Lists

Learning Objective

Write a Python program that uses lists, string processing, and loops.

Assignment Description

Write a program that displays a jumbled word to the user. Have the user guess the word and continue guessing until they get it correct. The user will have a choice to display a hint. After the user correctly enters the word, display the number of guesses it took for them to enter the correct word.

Steps

1. In PyCharm (Community Edition), open an existing project (such as ITP115) or create a new project.
   - If you open an existing project, then create a new directory (probably under the Assignments directory) named ***a6_last_first*** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.
   - If you create a new project, then name it ***a6_last_first*** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.
2. In the project or directory, create a new Python file called **assignment6.py**. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

```
# Name, USC email
# ITP 115, Summer 2022
# Section: number or nickname
# Assignment 6
# Description:
# Describe what this program does.
```

3. Import the random module.
4. Create two lists that contain strings. The first list contains words, and the second list contains hints for the words. Put the hints in the same order as the words such that the first hint goes with the first word. Put at least three strings in each list.
5. Get a random word and the corresponding hint from the lists and save them into variables. Here are two ways to do this. Pick one.
   - Get a random index depending on the size of the word list using the random.randrange() or random.randint() function. Use that index to get the random word and the corresponding hint.

- Get a random word using the random.choice() function. Use the list.index() method to get the index of the random word from the word list. Use that index to get the hint from the hints list.

6. Create a jumbled word from the random word by using lists.
   - Turn the random word into a list using the list(str) function.
   - Create an empty string to hold the jumbled word.
   - Loop through the list while it has elements. You can use the len() function.
   - In the loop, get a random letter from the list using the random.choice() function. Add the letter to the end of the jumbled word and remove it from the list.
   - Do not use random.shuffle(), random.sample(), or similar methods.
7. Print the jumbled word to the user. Here is an example:

```
The jumbled word is "vreeatlr"
```

8. Have the user guess the word until they get enter the correct word. Use a loop. Count the number of guesses it take for the user to enter the correct word. If they do not enter the correct word, ask them if they want a hint. If they enter "y" or "Y", give them a hint.

```
Enter your guess: vetreatlr
That is not correct
Do you want a hint (y or n)? q
The jumbled word is "vreeatlr"
Enter your guess: treatlrve
That is not correct
Do you want a hint (y or n)? Y
The hint is "horse"
The jumbled word is "vreeatlr"
Enter your guess: traveler
```

9. After the user enters the correct word (i.e., after the loop), print out a message to the user with the number of guesses it took them.

```
You got it!
It took you 3 guesses
```

10. Be sure to comment your code. This means that there should be comments throughout your code. Generally, a comment for every section of code explaining what it does. Points will be deducted for not having comments.
11. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
12. Test the program. Look at the Sample Output below. Assignment that do not run are subject to 20% penalty.
13. Prepare your submission:

- Find the **a6_*last_first*** folder on your computer and compress it. This cannot be done within PyCharm.
- On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
- On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.

14. Upload the zip file to your Blackboard section:
    - On Blackboard, navigate to the appropriate item.
    - Click on the specific item for this assignment.
    - Click on the **Browse Local Files** button and select the zip file.
    - Click the **Submit** button.

Grading

- This assignment is worth 30 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

Item	Points
Two lists	*5*
Get random word and corresponding hint	*5*
Jumble the word	*5*
Loop to have user guess	*5*
Count number of guesses and print after loop	*5*
Print hint if user wants	*5*
Total	**30**

Sample Output

```
The jumbled word is "thnyop"
Enter your guess: nopthy
That is not correct
Do you want a hint (y or n)? N
The jumbled word is "thnyop"
Enter your guess: thypon
That is not correct
Do you want a hint (y or n)? y
The hint is "code"
The jumbled word is "thnyop"
Enter your guess: Python
You got it!
It took you 3 guesses.
```