Assignment 4 – Character Counter & D20 Dice Game

Goals

- Part 1: Practice for each loops
- Part 2: Practice range-based for loops

Note: You may **not** use the **`count()`** or **`find()`** functions in this assignment.

Requirements

- Create a new Python file for this assignment. You may put both parts into one file or create another Python file for the second part.
- Your new file(s) must begin with comments in the following format (replace the name and email with your actual information and write text for the description):

```
# Name, USC email
# ITP 115, Spring 2020
# Assignment 4
# Description:
# Describe what this program does.
```

Part 1 – Character Counter (For Each Loops)

- **Requirements**
  - Ask the user for a sentence.
  - Using a for each loop, look at every letter in the sentence in order to count how many times each letter of the alphabet appears, and how many times special characters occur.
    - Special characters are any characters that aren't letters or spaces.
    - Numbers count as special characters.
  - Print out each letter of the alphabet followed by the number of asterisks representing how many times that letter appeared in the sentence.
    - If the letter never appeared, print "NONE".
    - Remember to also print out asterisks for the number of special characters appearing in the sentence.

Part 2 – D20 Dice Game (Range-Based For Loops)

- **Background**
  - 20-sided dice were used as early as 300 BC in Ancient Egyptian, but only became popular in the modern world around 1969 in board games.

- The naming convention of dice is a D followed by the number of sides. The most common dice are 6-sided: D6. In this program, you will be rolling a virtual 20-sided die, called a D20!

- **Requirements**
  - Overview: This program plays a game where the user wins points by rolling one of the winning numbers. Before the user's roll, the program prints out which numbers qualify for points. The game is played 10 times and the user's score is printed at the very end.
    - "Rolling" the D20 will be virtually done by randomly generating a number, therefore this program takes **no user input**
  - There are multiple winning numbers for each of the 5 different cases:
    - Case 1: Roll a number **any even number**
      - Valid winning numbers for Case 1 are **2, 4, 6, (…) 18, 20**
    - Case 2: Roll **any odd number**
      - Valid winning numbers for Case 2 are **1, 3, 5, (…) 15, 17, 19**
    - Case 3: Roll **any number 5 through 10 inclusive**
      - Valid winning numbers for Case 3 are **5, 6, 7, 8, 9, 10**
    - Case 4: Roll an **even number 10 or greater**
      - Valid winning numbers for Case 4 are **10, 12, 14, 16, 18, 20**
    - Case 5: Roll **any multiple of 3**
      - Valid winning numbers for Case 5 are **3, 6, 9, 12, 15, 18**
  - Use a for loop to run the game **10 ten times**
  - Each time the game is played, randomly assign the game one of the five cases by using the **random.randrange()** function
  - Tell the user which case they are playing for and print out the valid winning numbers so they know whether their roll qualifies as a win.
    This is the tricky part of the program:
    - **You must use range-based for loops to print the valid winning numbers for each of the 5 cases**
    - **Do NOT hardcode printing the winning numbers**
    - Hint: Review how to write 2-input and 3-input range-based for loops
  - "Roll" the D20 by generating a random number, and check whether their roll produced a winning number. If so, award **50 points** to their score
    - When checking whether the user's roll was a winning number, use if/else statements. You may hardcode this check, but using math operations is easier and much shorter (>, >=, <, <=, %, etc.)
    - Keep adding the scores for each of the 10 times the game is played

- After the game has been played 10 times, print out the user's final score and end the program

- **Hints**
  - Begin programming by thinking about how you would play the game just 1 time.
  - Start by figuring out how to write the print statements for each of the winning cases. This way, once the program randomly determines which case the user is playing for, you can use if/else statements to print your work.
    - Remember that dice start at 1, not zero!
  - The for loop you write to run the game 10 times must encompass almost all your code except for declaring and printing out the user's final score.
    - When checking whether your program works, it may be easier to simply run the game once, so that your output is shorter and easier to read.
    - The most important thing is to make sure your winning numbers are printing correctly! Secondly, make sure your math is correct in checking whether the roll won points.

Sample Output for Part 1

```
PART 1 - Character Counter
Please enter a sentence: All along the watchtower princes kept the
view 1967

Here is the character distribution:

special characters: ****
a: ***
b: NONE
c: **
d: NONE
e: ******
f: NONE
g: *
h: ***
i: **
j: NONE
k: *
l: ***
m: NONE
n: **
o: **
```

```
p: **
q: NONE
r: **
s: *
t: *****
u: NONE
v: *
w: ***
x: NONE
y: NONE
z: NONE
```

Sample Output for Part 2

```
PART 2 - D20 Dice Game


You are playing for CASE 4

You will win for the following numbers:

10  12  14  16  18  20


Now rolling ...

You rolled a 1!

You didn't win :(


You are playing for CASE 4

You will win for the following numbers:

10  12  14  16  18  20


Now rolling ...

You rolled a 19!

You didn't win :(
```

```
You are playing for CASE 5
You will win for the following numbers:
3  6  9  12  15  18

Now rolling ...
You rolled a 20!
You didn't win :(

You are playing for CASE 4
You will win for the following numbers:
10  12  14  16  18  20

Now rolling ...
You rolled a 14!
You won 50 points! :)

You are playing for CASE 4
You will win for the following numbers:
10  12  14  16  18  20

Now rolling ...
You rolled a 12!
You won 50 points! :)

You are playing for CASE 3
You will win for the following numbers:
```

```
5  6  7  8  9  10

Now rolling ...
You rolled a 19!
You didn't win :(

You are playing for CASE 4
You will win for the following numbers:
10  12  14  16  18  20

Now rolling ...
You rolled a 11!
You didn't win :(

You are playing for CASE 4
You will win for the following numbers:
10  12  14  16  18  20

Now rolling ...
You rolled a 2!
You didn't win :(

You are playing for CASE 3
You will win for the following numbers:
5  6  7  8  9  10

Now rolling ...
```

```
You rolled a 8!
You won 50 points! :)

You are playing for CASE 1
You will win for the following numbers:
2  4  6  8  10  12  14  16  18  20

Now rolling ...
You rolled a 9!
You didn't win :(

Your total score is: 300
Thanks for playing!
```

Deliverables and Submission Instructions

- You may include part 1 and 2 in the same file as long as it is very clear through comments which part is which. You may also separate parts 1 and 2 into two files, as long as the names of the files are clear.
- Create a folder on your computer called **`ITP115_A4_`*`LastName_FirstName`*** (replace *LastName* with your last/family name and FirstName with your first name).
- Inside the folder, put your python source code.
- Compress the folder (make a zip file). This cannot be done within PyCharm. Find the folder on your computer and compress it.
  a. Windows*:*
     1. Using File Explorer, select your lab file
     2. Right click
     3. Send to ->
     4. Compressed (zipped) folder
  b. Mac OSX:
     1. Using Finder, select your lab file
     2. Right click
     3. Compress "*FileName*"
- Upload the zip file to your Blackboard section:
  1. On Blackboard, click on the Assignments item in the course menu on the left.
  2. Click on the specific item for this assignment (starts with A and a number).
  3. Click on the Browse My Computer button and select your zip file.
  4. Click the Submit button.

Grading

Item	Points
Part 1: Character Counter	15
Part 2: D20 Dice Game	15
Total*	**30**

** Points will be deducted for poor code style, lack of error checking, improper submission.*