CS-UY 1114: Lab 5

Strings

You must get checked out by your lab CA prior to leaving early. **If you leave without being checked out, you will receive 0 credits for the lab.**

Restrictions

The Python structures that you use in this lab should be restricted to those you have **learned in lecture so far (topics learned in Rephactor may only be used if they have also been taught in lecture)**. Please check with your teaching assistants in case you are unsure whether something is or is not allowed!

Create a new python file for each of the following problems.

Your files should be named *lab[num]_q[num].py* similar to homework naming conventions.

Problem 1: *String Fun!*

Solve this problem by hand on paper

1. Given the assignment statement below, write expressions that evaluate to the specified strings. For example, given the assignment statement below, the Python expression name[0] evaluates to "G".

    ```
    name = "Grace Hopper"
    ```

    - Write an expression whose value is "r".
    - Write an expression using the function len whose value is "r".
    - Write an expression whose value is "Grace". Use the slicing operator.

2. Given the assignment statement below, write expressions that evaluate to the specified strings. Use the slicing operator.

   ```
   name = "alan turing"
   ```

   - "Alan"
   - "Turing"

3. Given the following use a combination of slicing and concatenation to write expressions that evaluate to the specified strings.

   ```
   S = "Computer Science"
   ```

   - "Cmue cec"
   - "Comp Sci"

4. Given the following use a combination of slicing and concatenation to write expressions that evaluate to the specified strings.

   ```
   place = "Bletchley Park, England"
   ```

   - "England"
   - "yelhctelB"

Problem 2: *Formatting Fun*

Isn't formatting just the best? It makes writing more complicated strings much easier. In this question, you will be given variables and the final printed string, and you must implement the string using both .format() and fstrings. **DO NOT create new variables when creating your program.**

a. age = 20, name = "Oleg"

```
My name is Oleg, and I'm 20 years old.
```

b. friday_steps = 2400, saturday_steps = 5630, sunday_steps = 1094

```
I walked a total of 9124 steps this weekend!
```

c. exam1 = 65, exam2 = 85, final = 80.

```
I averaged a 76.66666666666667 across all my exams!
```

d. Wk1_temp = 81, wk2_temp = 75.

```
Last week it was 81 degrees, and this week it's 75 degrees.
That's a 6 degree difference!
```

Problem 3A: *Reverse It Slicing*

Note: You may **not** reverse the string for this problem through the `.reverse()` method

Write a program that will take an input from the user and reverse it. In this version of the problem, reverse the string using string slicing. This is a quick and straightforward problem that focuses on indexing with strings.

Here are some sample executions of the program:

```
Enter your phrase: hello
olleh
```

```
Enter your phrase: CS 1114
4111 SC
```

Problem 3B: *Reverse It Looping*

Implement 3A again except this time, reverse the string using a loop instead of string slicing

Sample executions of the program:

```
Enter your phrase: hello
olleh
```

```
Enter your phrase: CS 1114
4111 SC
```

Problem 4: *Mad For Mad Libs!*

Ever heard of mad libs? It’s a fun game where you’re given part of a story and you get to fill in various things like nouns or verbs to make the story your own.

Write a program that upon detection of certain characters, replace the initial string with the word of your choice.

- When a ‘V’ is detected, replace the character with a Verb
- When a ‘N’ is detected, replace the character with a Noun
- When a ‘U’ is detected, replace the character with a Number
- When a ‘A’ is detected, replace the character with an Adjective

A sample output of the program looks like this:

```
Input the mad lib text: It’s a A day today. Just perfect for V.
What's your adjective? rainy
What's your verb? fishing
```

```
Creating mad lib...
It's a rainy day today. Just perfect for fishing.
```

Here are some other input strings you can use:

```
It was a A and A October night. My pet N and I were V to find
out who was the best. Turns out my pet is better than me! Even
though I'm A, I still had a good time.
```

```
Welcome to the daily News report. It's a A day today, so
prepare yourselves! BREAKING NEWS! U N have broken out of the
zoo! I would recommend V to stay safe. Have a great day
everyone!
```

Hint: Since strings are immutable, how can we store these changes we want to make?

Problem 5: *It's A Secret…*

Write a program that takes as input a line of text and outputs a new line of text with the digits of the integer numbers replaced by the character X.
Below is a sample output of the program:

```
Enter a phrase: My userID is john17 and my 4 digit pin is 1234
which is secret
My userID is john17 and my X digit pin is XXXX which is secret
```

Notice that the digits in the username `john17` were **not** changed to x. If the digit is part of a word, it should **NOT** be changed. You may assume that the text entered by the user will contain only letters, digits and may also assume that there is exactly one space between each word in the line of text.

Hint: One approach to solving this problem is to use the slicing operator to isolate each word in the line. In order to do this, you will need to know the index position at which the word starts and ends. You can use the space character `" "` as an indicator of separation between words. You will need to iterate over the string and check if you encounter the space character `" "`. You will also need to store the index of the previous space you encountered so that when you encounter a new space, you will know what indices of the original input correspond to the start and end of a word. Note that you will need to account for the last word in a string differently.

Once the words are isolated the problem is more straightforward consider using the `isdigit()` method to your advantage.

Problem 6: *Secret Message Decoder*

You receive an encoded message from the user where some words of their message are capitalized. Putting together the first letter in each capitalized word spells out a secret message.

Create a program that accepts user input, decodes the message according to the algorithm described above, and displays the secret message. You may assume there is always a single whitespace `" "` between words.

Here are some example outputs:

```
Please enter your message:
Originally, the Green goBlin Rested in HiS forest for maNY Eras
Your secret message is: OGRE
```

```
Please enter your message:
bob went Out oF Office so he cAn Get SOme Apples
Your secret message is: OOGA
```