

## **Programming and Data Structures with Python Lab**

### **Lab2. Python Functions, Modules and String Processing**

**Question 1.** Create a function **prime()** that receives an integer and returns whether n is prime or not. Print all prime numbers from 1 to 100 by calling **prime()** function. For example,

```
prime(1)
prime(2)
.....
prime(100)
```

**Question 2.** Develop a simple arithmetic calculator for 4 operations. The program should continue calculation until user types 'q' to quit. A sample user interaction can be:

- Enter operator (q to quit): +
- Enter value 1: 10
- Enter value 2: 20
- Result = 30

Create 4 functions **add()**, **subtract()**, **multiply()** and **divide()** that receives two values and returns the result of the operation.

Now, perform the following operations by calling the corresponding functions. Validate your outputs.

1. 10+20
2. 20-5
3. 8\*5
4. 50/3

**Question3.** Create a function **factorial()** that takes an integer and returns its factorial value.

- You can create as a non-recursive version of factorial.
- Also, check factorial of negative number does not exist.
- Factorial of 0 is 1.
- Save this Python file as **factorial\_definition.py**.

Now, open another file and you can import factorial\_definition.py as follows:

- **import factorial\_definition**
- You can call factorial function as **factorial\_definition.factorial()**.

Now, print the following factorial values:

1. factorial\_definition.factorial(3)
2. factorial\_definition.factorial(5)
3. factorial\_definition.factorial(10)

**Question4.** Develop a function **count\_letter(string, search)** that returns the number of times search character appears in a string.

Test cases:

1. Str = "hello world". Search = 'o'. Calling **count\_letter(str, search)** should return output 2
2. Str = "HeLlo wOrld". Search = 'o'. Then, **calling count\_letter(str, search)** will return output 1

Modify **count\_letter()** so that it ignores case sensitivity, so that **o** and **O** are same.

3. Str = "HeLlo wOrld". Search = 'o'. Calling **count\_letter(str, search)** will return output 2

**Question5.** Write a program that counts the number of spaces, digits, vowels and consonants in a string that the user inputs. Print the string, no of spaces, no of digits, no of vowels and no of consonants.

Test case: Enter a string: **Bishop Heber College 17**. Then output should be:

Given string: Bishop Heber College 17

No. of spaces: 3

No. of digits: 2

No. of vowels: 7

No. of consonants: 12

**Question6.** Develop a function **remove\_punctuation(str)** that returns the string after removing the following punctuations.

Punctuation List = "!\"#\$%&'()\*+,-./:;<=>?@[\\]^\_`{|}~"

Test cases:

1. Str = "Bishop's College !.....". Calling **remove\_punctuation(str)** should return output as "Bishops College"

2. Str = "#bhc trending @cs \$placements::>." Calling **remove\_punctuation(str)** should return output as "bhc trending cs placements"

**Question7.** Write a program that asks the user for a word. Translate their word into Pig Latin. Pig Latin game takes the first consonant (or set of first consonants) of an English word, moves it to the end of the word and suffixes an **ay**. If the first letter is a vowel, do not move that vowel, but instead add “**way**” at the end of the word.

Test Cases:

1. Enter a word: **pig**  
Output: **ig-pay**
2. Enter a word: **banana**  
Output: **anana-bay**
3. Enter a word: **trash**  
Output: **ash-tray**
4. Enter a word: **apple**  
Output: **apple-way**
5. Enter a word: **orange**  
Output: **orange-way**

Modify your program so that it becomes a function **piglatin(word)** and returns translated word as output. Call this function 3 times with the same inputs and validate the outputs.