

# 235229122-pylab2-pavithiran-v

July 18, 2023

NAME: PAVITHIRAN. VROLL NO:235229122

Programming and Data Structures with Python 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)

```
[7]: def prime(num):  
    flag = 0  
    if num > 1:  
        for i in range(2, int(num/2) + 1):  
            if num % i == 0:  
                return False  
                flag = -1  
                break  
  
    else:  
        return True  
    if flag == 0:  
        return True  
  
for i in range(1, 101):  
    if prime(i):  
        print(i, end = " ")
```

1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

---

## 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.  $50/3$

```
[2]: def adding(a, b):
      return a + b
def sub(a, b):
      return a - b
def mul(a, b):
      return a * b
def div(a, b):
      return a / b
while True:
    val = input("Enter operator (q to quit): ")
    if val == "q":
        break
    var1 = int(input("Enter value 1: "))
    var2 = int(input("Enter value 2: "))
    if val == "+":
        result = adding(var1, var2)
    elif val == "-":
        result = sub(var1, var2)
    elif val == "*":
        result = mul(var1, var2)
    elif val == "/":
        result = div(var1, var2)
    print("result = {}".format(result))
```

```
Enter operator (q to quit): +
Enter value 1: 4
Enter value 2: 2
result = 6
Enter operator (q to quit): -
Enter value 1: 6
Enter value 2: 4
result = 2
Enter operator (q to quit): *
Enter value 1: 2
Enter value 2: 6
result = 12
Enter operator (q to quit): /
Enter value 1: 4
Enter value 2: 2
result = 2.0
Enter operator (q to quit): q
```

---

### Question 3

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)`

```
[10]: def factorial(n):
        if n < 0:
            print("There is no factorial for the given number.")
        elif n == 0:
            return 1
        else:
            result = 1
            for i in range(1, n + 1):
                result *= i
            return result
```

```
[12]: #import factorial_definition

print(factorial(3))
print(factorial(5))
print(factorial(10))
```

```
6
120
3628800
```

---

#### Question 4

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

```
[6]: def count_letter(string, search):
        count = 0
        for i in string:
            if i == search:
                count += 1
        return count
str1 = input("Enter the string :")
char1 = input("Enter the character :")
print(count_letter(str1.lower(), char1.lower()))
```

Enter the string : hello w0rld

Enter the charactor : o

2

---

### Question 5

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

```
[14]: vowels = ['a', 'e', 'i', 'o', 'u']
str1 = input("Enter the String :").lower()
space_count = 0
digit_count = 0
vowels_count = 0
consonants_count = 0
for i in str1:
    if i == " ":
        space_count += 1
    elif i.isdigit():
        digit_count += 1
    else:
        if i not in vowels:
            consonants_count += 1
        if i in vowels:
            vowels_count += 1

print("No. of spaces: {}".format(space_count))
print("No. of digits: {}".format(digit_count))
print("No. of vowels: {}".format(vowels_count))
print("No. of consonants: {}".format(consonants_count))
```

Enter the String : aeiou 12345 qwerty

No. of spaces: 2

No. of digits: 5

No. of vowels: 5

No. of consonants: 5

---

### Question 6

Develop a function `remove_punctuation(str)` that returns the string after removing the following punctuations.

```
[16]: # 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"

```
[1]: def remove_punctuation(str1):
    punctuation_list = "!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~"
    result = ""
    for char in str1:
        if char not in punctuation_list:
            result += char
    return result

str1 = input("Enter the String :")
print(remove_punctuation(str1))
```

Enter the String : Bishop's College !...

Bishop's College

---

### Question 7

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.

```
[9]: def piglatin(word):
    vowels = ['a', 'e', 'i', 'o', 'u']
    if word[0] in vowels:
        word2 = word + "-way"
    else:
        word3 = ""
        i = 0
        while i < len(word) and word[i] not in vowels:
            word3 += word[i]
            i += 1
        word2 = word[i:] + "-" + word3 + "ay"
    return word2

words = input("Enter a word: ")
print("output: {}".format(piglatin(words)))
```

Enter a word: banana

output: anana-bay

---