

# Rajalakshmi Engineering College

Name: SIVASURIYAN .  
Email: 240701518@rajalakshmi.edu.in  
Roll no:  
Phone: 7010603376  
Branch: REC  
Department: I CSE FF  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z) Uppercase letters (A-Z) Digits (0-9) Special characters (from string.punctuation, e.g. @, !, #, \$)

### ***Input Format***

The input consists of a single string representing the user's password.

### ***Output Format***

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length  $\geq 6$  and at least 2 different character types, the output prints "<password> is Moderate"

If Password length  $\geq 10$  and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: password123

Output: password123 is Moderate

### ***Answer***

```
p = input()
l,u,d,s,c = 0,0,0,0,0
for i in p:
    if i.islower():
        l = l+1
    if i.isupper():
        u = u+1
    if i.isdigit():
        d = d + 1
    if not i.isalnum():
        s = s+1
if(l>0):
    c = c+1
if(u>0):
    c = c+1
if(d>0):
```

```
c = c+1
if(s>0):
    c = c+1
t = len(p)
if(t>=10 and c == 4):
    print(p," is Strong")
elif(t >= 6 and c >= 2):
    print(p," is Moderate")
elif(t<6 or c<=2):
    print(p," is Weak")
```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Meena is analyzing a list of integers and needs to count how many numbers in the list are even and how many are odd. She decides to use lambda functions to filter the even and odd numbers from the list.

Write a program that takes a list of integers, counts the number of even and odd numbers using lambda functions, and prints the results.

### ***Input Format***

The first line contains an integer  $n$ , representing the number of integers in the list.

The second line contains  $n$  space-separated integers.

### ***Output Format***

The first line of output prints an integer representing the count of even numbers.

The second line of output prints an integer representing the count of odd numbers.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 7  
12 34 56 78 98 65 23  
Output: 5  
2

**Answer**

```
n = int(input())
l = list(map(int,input().split()))
odd = lambda a : a%2 == 1
even = lambda a : a%2 == 0
e,o = 0,0
for i in range(len(l)):
    if(odd(l[i])):
        o=o+1
    if(even(l[i])):
        e=e+1
print(e)
print(o)
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Create a program for a mathematics competition where participants need to find the smallest positive divisor of a given integer n. Your program should efficiently determine this divisor using the min() function and display the result.

**Input Format**

The input consists of a single positive integer n, representing the number for which the smallest positive divisor needs to be found.

**Output Format**

The output prints the smallest positive divisor of the input integer in the format: "The smallest positive divisor of [n] is: [smallest divisor]".

Refer to the sample output for the exact format.

### **Sample Test Case**

Input: 24

Output: The smallest positive divisor of 24 is: 2

### **Answer**

```
div = []
n = int(input())
for i in range(2,n+1):
    if(n%i == 0):
        div.append(i)
print("The smallest positive divisor of",n,"is:",min(div))
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**

You are tasked with designing a shipping cost calculator program that calculates the shipping cost for packages based on their weight and destination. The program utilizes different shipping rates for domestic, international, and remote destinations. The rates for each destination type are provided as global constants.

Constant Values:

DOMESTIC\_RATE = 5.0

INTERNATIONAL\_RATE = 10.0

REMOTE\_RATE = 15.0

Function Signature: calculate\_shipping(weight, destination)

Formula: shipping cost = weight \* destination rate

### **Input Format**

The first line of the input consists of a float representing the weight of the package.

The second line consists of a string representing the destinations(Domestic or

International or Remote).

### ***Output Format***

The program outputs any one of the following:

1. If the input is valid and the destination is recognized, the output should consist of a single line stating the calculated shipping cost for the given weight and destination in the format: "Shipping cost to [destination] for a [weight] kg package: \$[calculated cost]" with two decimal places.
2. If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."
3. If the input destination is not one of the valid options, print "Invalid destination."

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 5.5

Domestic

Output: Shipping cost to Domestic for a 5.5 kg package: \$27.50

### ***Answer***

```
#
```

```
weight = float(input())
```

```
destination = input()
```

```
f = 0
```

```
if (int(weight) < 0):
```

```
    print("Invalid weight.Weight must be greater than 0.")
```

```
    shipping_cost = None
```

```
    f = 1
```

```
else:
```

```
    if(destination == 'Domestic'):
```

```
        shipping_cost = weight * 5.0
```

```
    elif(destination == 'International'):
```

```
        shipping_cost = weight * 10.0
```

```
    elif(destination == 'Remote'):
```

```
        shipping_cost = weight * 15.0
```

```
    else:
```

```
    shipping_cost = None
if shipping_cost is None:
    if f == 0 :
        print("Invalid destination.")

if shipping_cost is not None:
    print(f"Shipping cost to {destination} for a {weight} kg package:
    ${shipping_cost:.2f}")
```

**Status :** Correct

**Marks :** 10/10