

# Rajalakshmi Engineering College

Name: Rayvan Sanjai  
Email: 240701425@rajalakshmi.edu.in  
Roll no: 2116240701425  
Phone: 9380572043  
Branch: REC  
Department: I CSE FD  
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

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***

#

DOMESTIC\_RATE=5.0

INTERNATIONAL\_RATE=10.0

REMOTE\_RATE=15.0

weight=float(input())

destination=input()

shipping\_cost=None

if weight<=0:

```
print("Invalid weight.")
print("Weight must be greater than 0.")
elif destination=="Domestic":
    shipping_cost=weight*DOMESTIC_RATE
elif destination=="International":
    shipping_cost=weight*INTERNATIONAL_RATE
elif destination=="Remote":
    shipping_cost=weight*REMOTE_RATE
else:
    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

## 2. Problem Statement

Develop a text analysis tool that needs to count the occurrences of a specific substring within a given text string.

Write a function `count_substrings(text, substring)` that takes two inputs: the text string and the substring to be counted. The function should count how many times the substring appears in the text string and return the count.

Function Signature: `count_substrings(text, substring)`

### ***Input Format***

The first line of the input consists of a string representing the text.

The second line consists of a string representing the substring.

### ***Output Format***

The output should display a single line of output containing the count of occurrences of the substring in the text string.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: programming is fun and programming is cool  
programming

Output: The substring 'programming' appears 2 times in the text.

### **Answer**

```
# You are using Python
def count_substrings(text: str, substring: str) -> None:
    count = text.count(substring)
    print(f"The substring '{substring}' appears {count} times in the text.")

# Example usage:
text = input()
substring = input()
count_substrings(text, substring)
```

**Status :** Correct

**Marks : 10/10**

## **3. Problem Statement**

Implement a program for a retail store that needs to find the highest even price in a list of product prices. Your goal is to efficiently determine the maximum even price from a series of product prices. Utilize the max() inbuilt function in the program.

For example, if the prices are 10 15 24 8 37 16, the even prices are 10 24 8 16. So, the maximum even price is 24.

### **Input Format**

The input consists of a series of product prices separated by a space.

The prices should be entered as a space-separated string of numbers.

### **Output Format**

If there are even prices in the input, the output prints "The maximum even price is: " followed by the maximum even price.

If there are no even prices in the input, the output prints "No even prices were found".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 10 15 24 8 37 16

Output: The maximum even price is: 24

### **Answer**

```
prices = list(map(int,input().split()))
max_even = None
for price in prices:
    if price % 2 == 0:
        if max_even is None or price > max_even:
            max_even = price
if max_even is not None:
    print(f"The maximum even price is: {max_even}")
else:
    print("No even prices were found")
```

**Status :** Correct

**Marks : 10/10**

## **4. 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())
```

```
numbers = list(map(int,input().split()))
```

```
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
```

```
odd_numbers = list(filter(lambda x : x % 2 != 0, numbers))
```

```
print(len(even_numbers))
```

```
print(len(odd_numbers))
```

**Status :** Correct

**Marks :** 10/10