

# Rajalakshmi Engineering College

Name: G Trishanth  
Email: 240701572@rajalakshmi.edu.in  
Roll no: 240701572  
Phone: 9840634543  
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

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**

```
def main():
    n = int(input().strip())
    if n == 1:
        smallest = 1
    else:
        smallest = min(d for d in range(2, n+1) if n % d == 0)
    print(f"The smallest positive divisor of {n} is: {smallest}")

if __name__ == "__main__":
    main()
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: `total_cost(item_cost)`

### **Input Format**

The input consists of a single line containing a positive floating-point number representing the cost of the item.

### **Output Format**

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 50.00

Output: Item Cost: \$50.00

Sales Tax Rate: 8.0%

Total Cost: \$54.00

### **Answer**

#

```
SALES_TAX_RATE = 0.08
```

```
def total_cost(item_cost):  
    return item_cost * (1 + SALES_TAX_RATE)
```

```
item_cost = float(input().strip())
```

```
total_cost = total_cost(item_cost)  
print(f"Item Cost: ${item_cost:.2f}")  
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")  
print(f"Total Cost: ${total_cost:.2f}")
```

**Status :** Correct

**Marks : 10/10**

### **3. 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

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***

```
password = input().strip()
```

```
has_lower = any(c.islower() for c in password)
has_upper = any(c.isupper() for c in password)
has_digit = any(c.isdigit() for c in password)
has_special = any(c in "!@#$%^&*()-_+=[]{};'\",.<>/?\\" for c in password)
char_types = sum([has_lower, has_upper, has_digit, has_special])
if len(password) < 6 or char_types < 2:
    print(f"{password} is Weak")
elif len(password) >= 10 and char_types == 4:
    print(f"{password} is Strong")
else:
    print(f"{password} is Moderate")
```

**Status :** Correct

**Marks :** 10/10