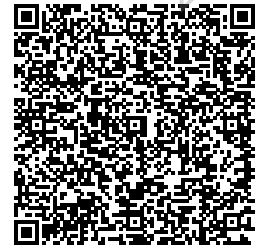


# Rajalakshmi Engineering College

Name: Keertana Anjimeeti Srihari  
Email: 240701252@rajalakshmi.edu.in  
Roll no: 240701252  
Phone: 9884916024  
Branch: REC  
Department: I CSE AH  
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 : 36

### Section 1 : Coding

#### 1. 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**

```
# You are using Python
def find_max_even_price(prices):
    price_list = list(map(int, prices.split()))

    even_prices = [price for price in price_list if price % 2 == 0]

    if even_prices:

        print(f"The maximum even price is: {max(even_prices)}")
    else:
        print("No even prices were found")

prices = input()

find_max_even_price(prices)
```

**Status :** Correct

**Marks :** 10/10

## **2. 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**

```
# You are using Python
def smallest_divisor(n):
```

```
    divisors = [i for i in range(2, n + 1) if n % i == 0]
```

```
    return min(divisors, default=n)
```

```
n = int(input())
```

```
print(f"The smallest positive divisor of {n} is: {smallest_divisor(n)}")
```

**Status :** Correct

**Marks :** 10/10

## **3. 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***

```
import string
```

```
def check_password_strength(password):
```

```
    length = len(password)
```

```
char_types = 0
```

```
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 string.punctuation for c in password)
```

```
if has_lower:
```

```
    char_types += 1
```

```
if has_upper:
```

```
    char_types += 1
```

```
if has_digit:
```

```
    char_types += 1
```

```
if has_special:
```

```
    char_types += 1
```

```
if length < 6 or char_types < 2:
```

```
    return f"{password} is Weak"
```

```
elif length >= 6 and char_types >= 2:
```

```
    return f"{password} is Moderate"
```

```
elif length >= 10 and char_types == 4:
```

```
    return f"{password} is Strong"
```

```
else:
```

```
    return f"{password} is Moderate"
```

```
if __name__ == "__main__":
```

```
    password = input()
```

```
    strength = check_password_strength(password)
```

```
    print(strength)
```

**Status :** Partially correct

**Marks :** 6/10

#### 4. 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***

```
#
```

```
# You are using Python
```

```
SALES_TAX_RATE = 0.08
```

```
def total_cost(item_cost):
```

```
tax_amount = item_cost * SALES_TAX_RATE
total_cost_value = item_cost + tax_amount
return total_cost_value

if __name__ == "__main__":

    item_cost = float(input())
    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