

# Rajalakshmi Engineering College

Name: Deepthi Dharunya R  
Email: 240701105@rajalakshmi.edu.in  
Roll no: 240701105  
Phone: 7667600005  
Branch: REC  
Department: I CSE FA  
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

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 maxx(price):
    p = list(map(int,price.strip().split()))
    e = []
    for i in p:
        if i%2==0:
            e.append(i)
    if e:
        return "The maximum even price is:"+ str(max(e))
    else:
        return "No even prices were found"
n = input()
print(maxx(n))
```

**Status : Correct**

**Marks : 10/10**

## **2. Problem Statement**

Arjun is working on a mathematical tool to manipulate lists of numbers. He needs a program that reads a list of integers and generates two lists: one containing the squares of the input numbers, and another containing the cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the

input list using lambda functions.

### **Input Format**

The input consists of a single line of space-separated integers representing the list of input numbers.

### **Output Format**

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 1 2 3

Output: [1, 4, 9]

[1, 8, 27]

### **Answer**

```
# You are using Python
n = list(map(int, input().split()))
sq = lambda x: x ** 2
cube = lambda x: x**3
sqs = list(map(sq, n))
cubes = list(map(cube, n))
print(sqs)
print(cubes)
```

**Status :** Correct

**Marks :** 10/10

## **3. 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 P

SALES\_TAX\_RATE = 0.08

def total\_cost(item\_cost):

tax\_amount = item\_cost \* SALES\_TAX\_RATE

return item\_cost + tax\_amount

```
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

#### 4. 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 ≥ 6 and at least 2 different character types, the output prints "<password> is Moderate"

If Password length ≥ 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**

# You are using Python

import string

def pass\_strength(passw):

    lower = any(c.islower() for c in passw)

    upper = any(c.isupper() for c in passw)

    digit = any(c.isdigit() for c in passw)

    spec = any(c in string.punctuation for c in passw)

    count = sum([lower, upper, digit, spec])

    leng = len(passw)

    if leng >= 10 and count == 4:

        print(f"{passw} is Strong")

    elif leng >= 6 and count >= 2:

        print(f"{passw} is Moderate")

    else:

        print(f"{passw} is Weak")

a = input()

pass\_strength(a)

**Status : Correct**

**Marks : 10/10**