# Rajalakshmi Engineering College

Name: Jhanani shree
Email: 240701215@rajalakshmi.edu.in
Roll no: 240701215
Phone: 7373333511
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 : 26

## Section 1 : Coding

1. 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***

```
# You are using Python
def count_even_odd(numbers):
    count_even = len(list(filter(lambda x: x % 2 == 0, numbers)))
    count_odd = len(list(filter(lambda x: x % 2 != 0, numbers)))
    return count_even, count_odd

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

count_even, count_odd = count_even_odd(numbers)

print(count_even)
print(count_odd)
```

***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_positive_divisor(n):
    divisors = [i for i in range(1, n + 1) if n % i == 0]
    return min(divisors)

n = int(input())
divisor = smallest_positive_divisor(n)

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

***Status :*** Wrong ***Marks : 0/10***

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 ≥ 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 password_strength(password):
    length = len(password)
    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)
```

```
    character_types = sum([has_lower, has_upper, has_digit, has_special])

    if length < 6 or character_types < 2:
        return f"{password} is Weak"
    elif length >= 6 and character_types >= 2:
        return f"{password} is Moderate"
    elif length >= 10 and character_types == 4:
        return f"{password} is Strong"

password = input()
result = password_strength(password)
print(result)
```

***Status :*** Partially correct ***Marks : 6/10***

4. 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 max_even_price(prices):
    even_prices = [price for price in prices if price % 2 == 0]

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

input_prices = input().strip().split()
prices = list(map(int, input_prices))

result = max_even_price(prices)
print(result)
```

***Status :*** Correct ***Marks : 10/10***