Rajalakshmi Engineering College

Name: Rittvik S
Email: 240701616@rajalakshmi.edu.in
Roll no: 240701616
Phone: null
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

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

```
numbers_str = input().split()
numbers = [int(num) for num in numbers_str]

squared = list(map(lambda x: x**2, numbers))
cubed = list(map(lambda x: x**3, numbers))

print(squared)
print(cubed)
```

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

```
text = input()
substring = input()
occurrences = text.count(substring)
print(f"The substring '{substring}' appears {occurrences} times in the text.")
```

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

3. 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

```
n = int(input())

if n == 1:
    print(f"The smallest positive divisor of [{n}] is: 1")
else:
    divisors = [i for i in range(2, n + 1) if n % i == 0]
    smallest_divisor = min(divisors) if divisors else n
    print(f"The smallest positive divisor of {n} is: {smallest_divisor}")
```

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

```
import string

password = input().strip()

has_lower = any(char.islower() for char in password)
has_upper = any(char.isupper() for char in password)
has_digit = any(char.isdigit() for char in password)
has_special = any(char in string.punctuation for char in password)

char_types = sum([has_lower, has_upper, has_digit, has_special])

if len(password) < 6 or char_types < 2:
    strength = "Weak"
elif len(password) >= 10 and char_types == 4:
    strength = "Strong"
else:
    strength = "Moderate"
```

```
print(f"{password} is {strength}")
```

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