Rajalakshmi Engineering College

Name: Aldrine Linjoe.s
Email: 240701033@rajalakshmi.edu.in
Roll no: 240701033
Phone: 7092049029
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

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
def total_cost(item_cost):
    total=item_cost+s
    return total
item_cost=float(input())
s=item_cost*0.08
SALES_TAX_RATE=0.08

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***

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(n):
    d=[i for i in range(2,n+1)if(n%i==0)]
    s=min(d)
    print(f"The smallest positive divisor of {n} is: {s}")
n=int(input())
smallest(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 ≥ 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
a=input()
l=len(a)
ca=0
```

```
sa=0
n=0
s=0
for i in a:
    if(i.isupper()):
        ca+=1
    elif(i.isdigit()):
        n+=1
    elif(i.islower()):
        sa+=1
    else:
        s+=1
p=0
if(n!=0):
    p+=1
if(sa!=0):
    p+=1
if(ca!=0):
    p+=1
if(s!=0):
    p+=1

if(l<6 or p<2):
    print(f"{a} is Weak")
elif((l>=6 and l<10)or(p==2 or p==3)):
    print(f"{a} is Moderate")
else:
    print(f"{a}is strong")
```

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

4. 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

```
# You are using Python
def count_substrings(n,m):
    c=n.count(m)
    print(f"The substring '{m}'appears {c} times in the text.")
n=input()
m=input()
count_substrings(n,m)
```

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