Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
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 : 0

Section 1 : Coding

1. 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

def check_password_strength(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)

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

    if length < 6 or char_types < 2:
        print(f"{password} is Weak")
    elif length >= 10 and char_types == 4:
        print(f"{password} is Strong")
    else:
```

print(f"{password} is Moderate")

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

2. 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
# Global sales tax rate
sales_tax_rate = 0.08

def total_cost(item_cost):
    tax_amount = item_cost * sales_tax_rate
    total = item_cost + tax_amount
    return total

# Read input
item_cost = float(input())

# Calculate total cost
total = total_cost(item_cost)

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

3. Problem Statement

You are tasked with designing a shipping cost calculator program that calculates the shipping cost for packages based on their weight and destination. The program utilizes different shipping rates for domestic, international, and remote destinations. The rates for each destination type are provided as global constants.

Constant Values:

DOMESTIC_RATE = 5.0

INTERNATIONAL_RATE = 10.0

REMOTE_RATE = 15.0

Function Signature: calculate_shipping(weight, destination)

Formula: shipping cost = weight * destination rate

Input Format

The first line of the input consists of a float representing the weight of the package.

The second line consists of a string representing the destinations(Domestic or International or Remote).

Output Format

The program outputs any one of the following:

1. If the input is valid and the destination is recognized, the output should consist of a single line stating the calculated shipping cost for the given weight and destination in the format: "Shipping cost to [destination] for a [weight] kg package: $[calculated cost]" with two decimal places.
2. If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."
3. If the input destination is not one of the valid options, print "Invalid destination."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5.5
Domestic
Output: Shipping cost to Domestic for a 5.5 kg package: $27.50

Answer

```
#
# You are using Python
# Global shipping rates
DOMESTIC_RATE = 5.0
```

```
INTERNATIONAL_RATE = 10.0
REMOTE_RATE = 15.0

def calculate_shipping(weight, destination):
    if weight <= 0:
        print("Invalid weight. Weight must be greater than 0.")
        return

    if destination == "Domestic":
        rate = DOMESTIC_RATE
    elif destination == "International":
        rate = INTERNATIONAL_RATE
    elif destination == "Remote":
        rate = REMOTE_RATE
    else:
        print("Invalid destination.")
        return

    cost = weight * rate
    print(f"Shipping cost to {destination} for a {weight} kg package: ${cost:.2f}")

# Input reading
try:
    weight = float(input())
    destination = input().strip()
    calculate_shipping(weight, destination)
except:
    print("Invalid input.")
if shipping_cost is not None:
    print(f"Shipping cost to {destination} for a {weight} kg package:
${shipping_cost:.2f}")
```

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

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
# Sample Input 1
count_substrings("programming is fun and programming is cool programming", "programming")
# Output: The substring 'programming' appears 2 times in the text.

# Sample Input 2
count_substrings("hello world python", "python")
# Output: The substring 'python' appears 1 times in the text.

# Sample Input 3
count_substrings("python is a popular programming language and Python is great. Python", "Python")
# Output: The substring 'Python' appears 2 times in the text.
```

Status : Wrong *Marks : 0/10*

Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_COD_Updated

Attempt : 1
Total Mark : 50
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Sara is developing a text-processing tool that checks if a given string starts with a specific character or substring. She needs to implement a function that accepts a string and a character (or substring), and returns True if the string starts with the provided character/substring, or False otherwise.

Write a program that uses a lambda function to help Sara perform this check.

Input Format

The first line contains a string `str` representing the main string to be checked.

The second line contains a string `n`, which is the character or substring to check if the main string starts with it.

Output Format

The first line of output prints "True" if the string starts with the given character/ substring, otherwise prints "False".

Refer to the sample for the formatting specifications.

Sample Test Case

Input: Examly
e
Output: False

Answer

```
# You are using Python
def starts_with(s, prefix):
    check = lambda x, y: x.startswith(y)
    return check(s, prefix)

# Read input
main_str = input().rstrip("\n")
prefix = input().rstrip("\n")

# Perform check and print result
print(starts_with(main_str, prefix))
```

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

2. Problem Statement

Imagine you are developing a text analysis tool for a cybersecurity company. Your task is to create a function that analyzes input strings to categorize and count the characters into four categories: uppercase letters, lowercase letters, digits, and special characters. The company needs this tool to process log files and identify potential security threats.

Function Signature: analyze_string(input_string)

Input Format

The input consists of a single string (without space), which may include uppercase letters, lowercase letters, digits, and special characters.

Output Format

The first line contains an integer representing the count of uppercase letters in the format "Uppercase letters: [count]".

The second line contains an integer representing the count of lowercase letters in the format "Lowercase letters: [count]".

The third line contains an integer representing the count of digits in the format "Digits: [count]".

The fourth line contains an integer representing the count of special characters in the format "Special characters: [count]".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: Hello123

Output: Uppercase letters: 1
Lowercase letters: 4
Digits: 3
Special characters: 0

Answer

-

Status : Skipped ***Marks : 0/10***

3. Problem Statement

Imagine you are building a messaging application, and you want to know the length of the messages sent by the users. You need to create a program that calculates the length of a message using the built-in function len().

Input Format

The input consists of a string representing the message.

Output Format

The output prints an integer representing the length of the entered message.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: hello!!
Output: 7

Answer

```
# You are using Python
# Read the input message
message = input()

message_length = len(message)


print(message_length)
```

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

4. Problem Statement

Sneha is building a more advanced exponential calculator. She wants to implement a program that does the following:

Calculates the result of raising a given base to a specific exponent using Python's built-in pow() function.Displays all intermediate powers from base¹ to base^exponent as a list.Calculates and displays the sum of these intermediate powers.

Help her build this program to automate her calculations.

Input Format

The input consists of line-separated two integer values representing base and exponent.

Output Format

The first line of the output prints the calculated result of raising the base to the exponent.

The second line prints a list of all powers from base^1 to base^exponent.

The third line prints the sum of all these powers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
3

Output: 8
[2, 4, 8]
14

Answer

```
# You are using Python
def exponential_calculator(base, exponent):
    result = pow(base, exponent)

    powers = [pow(base, i) for i in range(1, exponent + 1)]

    sum_powers = sum(powers)

    print(result)
    print(powers)
    print(sum_powers)

base = int(input())
exponent = int(input())

exponential_calculator(base, exponent)
```

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

5. Problem Statement

Implement a program that needs to identify Armstrong numbers. Armstrong numbers are special numbers that are equal to the sum of their digits, each raised to the power of the number of digits in the number.

Write a function is_armstrong_number(number) that checks if a given number is an Armstrong number or not.

Function Signature: armstrong_number(number)

Input Format

The first line of the input consists of a single integer, n, representing the number to be checked.

Output Format

The output should consist of a single line that displays a message indicating whether the input number is an Armstrong number or not.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 153

Output: 153 is an Armstrong number.

Answer

```
# You are using Python
def armstrong_number(number):
    # Convert the number to string to easily access its digits
    digits = str(number)
    num_digits = len(digits)

    # Calculate the sum of each digit raised to the power of num_digits
    total = sum(int(digit) ** num_digits for digit in digits)
```

```
    # Check if the total equals the original number
    if total == number:
        print(f"{number} is an Armstrong number.")
    else:
        print(f"{number} is not an Armstrong number.")

# Read input
n = int(input())

# Call the function to check if the number is an Armstrong number
armstrong_number(n)
```

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

Rajalakshmi Engineering College

Name: KEERTHI PRIYA T
Email: 240701258@rajalakshmi.edu.in
Roll no: 2116240701258
Phone: 7397397221
Branch: REC
Department: I CSE FC
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 10

Section 1 : MCQ

1. Which of the following functions can take a lambda function as a parameter in Python?

Answer

All of the mentioned options

Status : Wrong ***Marks : 0/1***

2. What is the output of the following code snippet?

```
def my_function(x):
    x += 5
    return x

a = 10
```

```
result = my_function(a)
print(a, result)
```

Answer

10 15

Status : Correct ***Marks : 1/1***

3. What is the main advantage of using lambda functions in Python?

Answer

They allow you to write shorter code than regular functions

Status : Correct ***Marks : 1/1***

4. What is the output of the code shown?

```
def f1():
 global x
 x+=1
 print(x)
x=12
print("x")
```

Answer

x

Status : Correct ***Marks : 1/1***

5. What will be the output of the following Python code?

```
multiply = lambda x, y: x * y
print(multiply(2, 'Hello'))
```

Answer

HelloHelloHelloHello

Status : Wrong ***Marks : 0/1***

6. What will be the output of the following code?

```
number = 7
result = abs(number) + pow(number, 2)
print(result)
```

Answer

56

Status : Correct ***Marks : 1/1***

7. What is the output of the following code?

```
x=12
def f1(a,b=x):
 print(a,b)
x=15
f1(4)
```

Answer

12 4

Status : Wrong ***Marks : 0/1***

8. What will be the output of the following code?

```
num1 = 10
num2 = -10
result = abs(num1) + abs(num2)
print(result)
```

Answer

20

Status : Correct ***Marks : 1/1***

9. What keyword is used to define a lambda function in Python?

Answer

lambda

Status : Correct **Marks : 1/1**

10. What will be the output of the following Python code?

```
def cube(x):
    return x * x * x
x = cube(3)
print(x)
```

Answer

27

Status : Correct **Marks : 1/1**

11. What will be the output of the following code?

```
num = -5
result = abs(num)
print(result)
```

Answer

5

Status : Correct **Marks : 1/1**

12. What will be the output of the following Python code?

```
def absolute_value(x):
    if x < 0:
        return -x
    return x

result = absolute_value(-9)
print(result, absolute_value(5))
```

Answer

9 5

Status : Correct **Marks : 1/1**

13. What is the output of the code shown below?

```
def f1(x):
    x += 1
    print(x)

global_variable = 15
f1(global_variable)
print("hello")
```

Answer

16hello

Status : Correct **Marks : 1/1**

14. What will be the output of the following Python code?

```
def C2F(c):
    return c * 9/5 + 32
print(C2F(100))
print(C2F(0))
```

Answer

314.024.0

Status : Wrong **Marks : 0/1**

15. What will be the output of the following code?

```
def display(*args):
    for arg in args:
        print(arg)

display(10, 20, 30)
```

Answer

(10, 20, 30)

Status : Wrong *Marks : 0/1*