Rajalakshmi Engineering College

Name: Thivyasri T
Email: 240701571@rajalakshmi.edu.in
Roll no: 240701571
Phone: 9043886744
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_PAH_Updated

Attempt : 1
Total Mark : 60
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Rajesh wants to design a program that simulates a real-time scenario based on a mathematical concept known as the Collatz Conjecture. This concept involves the repeated application of rules to a given starting number until the number becomes 1. The rules are as follows:

If the number is even, divide it by 2.If the number is odd, multiply it by 3 and add 1.

Your task is to write a program that takes a positive integer as input, applies the Collatz Conjecture rules to it, counts the number of steps taken to reach 1, and provides an output accordingly. If the process exceeds 100 steps, the program should print a message indicating so and use break to exit.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the total number of steps taken to reach 1 if it's under 100.

If it's more than 100, it displays "Exceeded 100 steps. Exiting...".

Refer to sample output for the formatting specifications.

Sample Test Case

Input: 6
Output: Steps taken to reach 1: 8

Answer

```
def collatz_conjecture(n):
    steps=0
    while n!=1:
        if steps>=100:
            print("Exceeded 100 steps.Exiting...")
            return
        if n%2==0:
            n//=2
        else:
            n=3*n+1
        steps+=1
    print(f"Steps taken to reach 1:{steps}")
n=int(input())
if 1<=n<=100:
    collatz_conjecture(n)
else:
    print("The input must be a positive integer between 1 and 100.")
```

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

2. Problem Statement

Aarav is fascinated by the concept of summing numbers separately based on their properties. He plans to write a program that calculates the sum of even numbers and odd numbers separately from 1 to a given positive integer.

Aarav wants to input an integer value to represent the upper limit of the range. Help Aarav by developing a program that computes and displays the sum of even and odd numbers separately.

Input Format

The input consists of a single integer N, where N is the upper limit of the range.

Output Format

The output consists of two lines:

- The first line displays the sum of even numbers from 1 to N.
- The second line displays the sum of odd numbers from 1 to N.

Refer to the sample output for the exact format.

Sample Test Case

Input: 10
Output: Sum of even numbers from 1 to 10 is 30
Sum of odd numbers from 1 to 10 is 25

Answer

```
def sum_even_odd(n):
    sum_even=sum_odd=0
    for i in range(1,n+1):
        if i%2==0:
            sum_even+=i
        else:
            sum_odd+=i
    print(f"Sum of even numbers from 1 to {n} is {sum_even}")
    print(f"Sum of odd numbers from 1 to {n} is {sum_odd}" )
if __name__=="__main__":
    N=int(input())
```

```
if 1<=N<=30:
    sum_even_odd(N)
else:
    print("N is out of the allowed range (1 to 30).")
```

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

3. Problem Statement

Imagine being entrusted with the responsibility of creating a program that simulates a math workshop for students. Your task is to develop an interactive program that not only calculates but also showcases the charm of factorial values.Your program should efficiently compute and present the sum of digits for factorial values of only odd numbers within a designated range. This approach will ingeniously keep even factorials at bay, allowing students to delve into the intriguing world of mathematics with enthusiasm and clarity.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the factorial and sum of digits of the factorial of odd numbers within the given range.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 6

Output: 1! = 1, sum of digits = 1
3! = 6, sum of digits = 6
5! = 120, sum of digits = 3

Answer

```
import math
def sum_of_digits(num):
```

```
    return sum(int(digit)for digit in str(num))
def compute_factorials_and_digit_sum(n):
    for i in range(1,n+1,2):
        factorial=math.factorial(i)
        digit_sum=sum_of_digits(factorial)
        print(f"{i}!={factorial},sum of digits={digit_sum}")
n=int(input())
if(1<=n<=20):
    compute_factorials_and_digit_sum(n)
else:
    print("The input must be an integer between 1 and 20.")
```

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

4. Problem Statement

As a software engineer, your goal is to develop a program that facilitates the identification of leap years in a specified range. Your task is to create a program that takes two integer inputs, representing the start and end years of the range and then prints all the leap years within that range.

Input Format

The first line of the input consists of an integer, which represents the start year.

The second line consists of an integer, which represents the end year.

Output Format

The output displays the leap years within the given range, separated by lines.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2000
2053
Output: 2000
2004
2008

2012
2016
2020
2024
2028
2032
2036
2040
2044
2048
2052

Answer

```
def is_leap_year(year):
    if year%4==0:
        if year%100==0:
            if year%400 ==0:
                return True
            return False
        return True
    return False

def print_leap_years(starts,end):
    for year in range(start,end+1):
        if is_leap_year(year):
            print(year)

start_year=int(input(""))
end_year=int(input(""))

if 1000<=start_year<=3000 and 1000<=end_year<=3000:
    print(start_year,end_year)
else:
    print("Input years must be between 1000 and 3000")
```

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

5. Problem Statement

Kamali recently received her electricity bill and wants to calculate the amount she needs to pay based on her usage. The electricity company

charges different rates based on the number of units consumed.

For the first 100 units, there is no charge. For units consumed beyond 100 and up to 200, there is a charge of Rs. 5 per unit. For units consumed beyond 200, there is a charge of Rs. 10 per unit.

Write a program to help Kamali calculate the amount she needs to pay for her electricity bill based on the units consumed.

Input Format

The input consists of an integer, representing the number of units.

Output Format

The output prints the total amount of the electricity bill, an integer indicating the amount Kamali needs to pay in the format "Rs. amount".

Refer to the sample output for the exact format.

Sample Test Case

Input: 350

Output: Rs. 2000

Answer

```
def calc_elec_bill(units):
    total_cost=0

    if units> 200:
        total_cost+=(units-200)*10
        units=200
    if units>100:
        total_cost+=(units-100)*5
    return total_cost
units= int(input(""))
bill_amount=calc_elec_bill(units)
print(f"Rs.{bill_amount}")
```

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

6. Problem Statement

Sophia, a primary school teacher, wants to calculate the sum of numbers within a given range, excluding those that are multiples of 3.

Write a program to help Sophia compute the sum of all numbers between start and end (inclusive) that are not divisible by 3 using the continue statement.

Input Format

The first line of input consists of an integer, representing the starting number of the range.

The second line of input consists of an integer, representing the ending number of the range.

Output Format

The output prints a single integer, representing the sum of numbers in the range that are not multiples of 3.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
10
Output: 37

Answer

```
def sum_three(start,end):
    total_sum=0
    for num in range(start,end+1):
        if num%3==0:
            continue
        total_sum+=num
    return total_sum
start = int(input(""))
end=int(input(""))
```

```
if start <1 or end<1 or start>100 or end>100:
    print("The start and end numbers must be between 1 and 100.")
else:
    result=sum_three(start,end)
    print(result)
```

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