

# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 1\_PAH

Attempt : 1  
Total Mark : 6  
Marks Obtained : 6

### Section 1 : Coding

#### 1. Problem Statement

Shawn, a passionate baker, is planning to bake cookies for a large party. His original recipe makes 15 cookies, with the following ingredient quantities: 2.5 cups of flour, 1 cup of sugar, and 0.5 cups of butter.

Write a program to calculate the amounts of flour, sugar, and butter needed for a different number of cookies. Provide the ingredient quantities for a specified number of cookies, maintaining the original proportions of the recipe.

#### ***Input Format***

The input consists of an integer  $n$ , representing the number of cookies.

#### ***Output Format***

The first line prints "Flour: X cups" where X represents the amount of flour required for n cookies, as a double value rounded to two decimal places.

The second line prints "Sugar: Y cups" where Y represents the amount of Sugar required for n, as a double value rounded to two decimal places.

The third line prints "Butter: Z cups" where Z represents the amount of flour required for n, as a double value rounded to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 15

Output: Flour: 2.50 cups

Sugar: 1.00 cups

Butter: 0.50 cups

### **Answer**

# You are using Python

```
n=int(input())
```

```
print(f"Flour: {(n/15)*2.5:.2f} cups\nSugar: {(n/15):.2f} cups\nButter: {(n/15)*0.5:.2f} cups")
```

**Status :** Correct

**Marks :** 1/1

## **2. Problem Statement**

Liam works at a car dealership and is responsible for recording the details of cars that arrive at the showroom. To make his job easier, he wants a program that can take the car's make, model, and price, and display the information in a formatted summary.

Assist him in the program.

### **Input Format**

The first line of input contains a string, representing the car make.

The second line contains a string, representing the car model.

The third line contains a float value, representing the car price.

### **Output Format**

The first line of output prints "Car Make: ", followed by the car make.

The second line prints "Car Model: ", followed by the car model.

The third line prints "Price: ", followed by the car price, formatted to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: Toyota

Camry

23450.75

Output: Car Make: Toyota

Car Model: Camry

Price: Rs.23450.75

### **Answer**

```
# You are using Python
```

```
x=input()
```

```
y=input()
```

```
z=float(input())
```

```
print(f"car Make: {x}\nCar Model: {y}\nPrice: Rs.{z:.2f}")
```

**Status :** Correct

**Marks :** 1/1

## **3. Problem Statement**

Ella, an avid TV show enthusiast, is planning a binge-watching marathon for a new series. She has a specific routine: after watching a set number of episodes, she takes a short break.

She is provided with the following information:

Each episode of the series has a fixed duration of 45 minutes. After a certain number of episodes, there is a break of 15 minutes.

Ella wants to know the total time she will need to watch the entire series, including the breaks. Your task is to help Ella by calculating the total viewing time.

### ***Input Format***

The first line of input consists of an integer E, representing the total number of episodes in the series.

The second line consists of an integer B, representing the number of episodes watched before taking a break.

### ***Output Format***

The output prints an integer representing the total viewing time required to watch the entire series, including the breaks.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 5

2

Output: 255 minutes

### ***Answer***

# You are using Python

```
E=int(input())
```

```
B=int(input())
```

```
if E%B ==0:
```

```
    z=(E//B)-1
```

```
else:
```

```
    z=E//B
```

```
print(f"{{(E*45)+((z)*15)}} minutes")
```

**Status :** Correct

**Marks :** 1/1

#### 4. Problem Statement

Mandy is debating with her friend Rachel about an interesting mathematical claim. Rachel asserts that for any positive integer  $n$ , the ratio of the sum of  $n$  and its triple to the integer itself is always 4. Mandy, intrigued by this statement, decides to validate it using logical operators and basic arithmetic.

She wants to confirm if the statement holds true for any positive integer  $n$ .

##### ***Input Format***

The input consists of a positive integer  $n$ , representing the integer to be tested.

##### ***Output Format***

The first line of output displays "Sum:" followed by an integer representing the calculated sum.

The second line displays "Rachel's statement is: " followed by a Boolean value indicating whether Rachel's statement is correct.

Refer to the sample output for the formatting specifications.

##### ***Sample Test Case***

Input: 12

Output: Sum: 48

Rachel's statement is: True

##### ***Answer***

```
# You are using Python
```

```
n=int(input())
```

```
print(f"Sum: {n+n*3}\nRachel's statement is: {True}")
```

**Status :** Correct

**Marks :** 1/1

#### 5. Problem Statement

Oliver is planning a movie night with his friends and wants to download a high-definition movie. He knows the file size of the movie in megabytes (MB) and his internet speed in megabits per second (Mbps). To ensure the movie is ready in time, Oliver needs to calculate the download time.

Your task is to write a program that calculates the download time and displays it in hours, minutes, and seconds.

Example

Input:

MB = 800

mbps = 40

Output:

Download Time: 0 hours, 2 minutes, and 40 seconds

Explanation:

Convert the file size to bits ( $800 \text{ MB} \times 8 \text{ bits/byte} = 6400 \text{ megabits}$ ) and divide it by the download speed ( $6400 \text{ Mbps} / 40 \text{ Mbps} = 160 \text{ seconds}$ ). Now, convert the download time in seconds to hours, minutes, and seconds: 160 seconds is equal to 2 minutes and 40 seconds. So, the download time is 0 hours, 2 minutes and 40 seconds.

### **Input Format**

The first line of input consists of an integer N, representing the file size in megabytes (MB).

The second line consists of an integer S, representing the network speed in megabits per second (mbps).

### **Output Format**

The output prints "Download Time: X hours, Y minutes, and Z seconds", where X, Y, and Z are integers representing the hours, minutes, and seconds respectively.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 180

3

Output: Download Time: 0 hours, 8 minutes, and 0 seconds

### **Answer**

# You are using Python

```
N=int(input())
```

```
S=int(input())
```

```
cvt=(N*8)/S
```

```
print(f"Download Time: {int(cvt//3600)}hours,{int((cvt%3600)//60)} minutes,  
and{int(cvt%60)} seconds")
```

**Status :** Correct

**Marks :** 1/1

## **6. Problem Statement**

A smart home system tracks the temperature and humidity of each room. Create a program that takes the room name (string), temperature (float), and humidity (float).

Display the room's climate details.

### **Input Format**

The first line of input consists of a string, representing the room name.

The second line consists of a float value, representing the temperature.

The third line consists of a float value, representing the humidity.

### **Output Format**

The first line of output prints "Room: " followed by the room name (string).

The second line prints "Temperature: " followed by the temperature (float) formatted to two decimal places.

The third line prints "Humidity: " followed by the humidity (float) formatted to two

decimal places and a percentage sign (%).

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: Living Room

23.45

45.78

Output: Room: Living Room

Temperature: 23.45

Humidity: 45.78%

**Answer**

# You are using Python

```
r=input()
```

```
t=float(input())
```

```
h=float(input())
```

```
print(f"Room: {r}\nTemperature:{t:.2f}\nHumidity: {h:.2f}%")
```

**Status :** Correct

**Marks :** 1/1



# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_COD\_Updated

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

#### Section 1 : Coding

##### 1. Problem Statement

As a junior developer working on a text analysis project, your task is to create a program that displays the consonants in a sentence provided by the user, separated by spaces.

You need to implement a program that takes a sentence as input and prints the consonants while skipping vowels and non-alphabetic characters using only control statements.

##### ***Input Format***

The input consists of a string representing the sentence.

##### ***Output Format***

The output displays space-separated consonants present in the sentence.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: Hello World!

Output: H I I W r l d

**Answer**

```
a=input()
b="aeiouAEIOU"
res=[]
for x in a:
    if x.isalpha() and x not in b:
        res.append(x)
result=(" ".join(res))
print(result)
```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Emma, a mathematics enthusiast, is exploring a range of numbers and wants to count how many of them are not Fibonacci numbers.

Help Emma determine the count of non-Fibonacci numbers within the given range [start, end] 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 consists of an integer, representing the ending number of the range.

**Output Format**

The output prints a single integer, representing the count of numbers in the range that are not Fibonacci numbers.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: 1

10

Output: 5

**Answer**

```
a=int(input())
b=int(input())
count=0
p=[0,1]
while(p[-1]<=100):
    p.append(p[-1]+p[-2])
for X in range(a,b+1):
    if(X not in p):
        count+=1
print(count)
```

**Status :** Correct

**Marks :** 10/10

**3. Problem Statement**

Ethan, a curious mathematician, is fascinated by perfect numbers. A perfect number is a number that equals the sum of its proper divisors (excluding itself). Ethan wants to identify all perfect numbers within a given range.

Help him write a program to list these numbers.

**Input Format**

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

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

### **Output Format**

The output prints all perfect numbers in the range, separated by a space.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1

100

Output: 6 28

### **Answer**

```
a=int(input())
b=int(input())
for x in range(a,b+1):
    count=0
    for i in range(1,x):
        if(x%i==0):
            count+=i
    if count==x:
        print(x,end=" ")
```

**Status :** Correct

**Marks : 10/10**

## **4. Problem Statement**

You work as an instructor at a math enrichment program, and your goal is to develop a program that showcases the concept of using control statements to manipulate loops. Your task is to create a program that takes an integer 'n' as input and prints the squares of even numbers from 1 to 'n', while skipping odd numbers.

### **Input Format**

The input consists of a single integer, which represents the upper limit of the range.

### **Output Format**

The output displays the square of even numbers from 1 to 'n' separated by lines.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 10

Output: 4

16

36

64

100

**Answer**

```
a=int(input())
for i in range(2,a+1,2):
    print(i**2)
```

**Status :** Correct

**Marks : 10/10**

**5. Problem Statement**

John, a software developer, is analyzing a sequence of numbers within a given range to calculate their digit sum. However, to simplify his task, he excludes all numbers that are palindromes (numbers that read the same backward as forward).

Help John find the total sum of the digits of non-palindromic numbers in the range [start, end] (both inclusive).

Example:

Input:

10

20

Output:

55

Explanation:

Range [10, 20]: Non-palindromic numbers are 10, 12, 13, 14, 15, 16, 17, 18, 19 and 20.

Digit sums:  $1+0 + 1+2 + 1+3 + 1+4 + 1+5 + 1+6 + 1+7 + 1+8 + 1+9 + 2+0 = 55$ .

Output: 55

### ***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 total sum of the digits of all non-palindromic numbers in the range.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 10  
20

Output: 55

### ***Answer***

```
a=int(input())
b=int(input())
def is_palindrome(n):
    return str(n) == str(n)[::-1]
def sum_of_digits(n):
    s=0
```

```
while n:
    s += n % 10
    n //= 10
return s
def total_digit_sum_non_palindromes(start, end):
    total_sum = 0
    for i in range(start, end + 1):
        if not is_palindrome(i):
            total_sum += sum_of_digits(i)
    return total_sum
start_range = a
end_range = b
result = total_digit_sum_non_palindromes(start_range, end_range)
print(result)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

Students are allowed to work on our computer center machines only after entering the correct secret code. If the code is correct, the message "Logged In" is displayed. They are not allowed to log in to the machine until they enter the correct secret code.

Write a program to allow the student to work only if he/she enters the correct secret code.

Note: Here, secret code means the last three digits should be divisible by the first digit of the number.

#### ***Input Format***

The input consists of an integer n, which represents the secret code.



### Output Format

The output displays either "Logged In" or "Incorrect code" based on the given condition.

Refer to the sample output for the formatting specifications.

### Sample Test Case

Input: 2345

Output: Incorrect code

### Answer

```
# You are using Python
def check_secret_code(n):
    n_str = str(n)
    if len(n_str) < 4:
        return "Incorrect code"
    first_digit = int(n_str[0])
    last_three_digits = int(n_str[1:])
    last_three_digits = int(n_str[-3:])
    if first_digit != 0 and last_three_digits % first_digit == 0:
        return "Logged In"
    else:
        return "Incorrect code"
n = int(input())
print(check_secret_code(n))
```

Status : Correct

Marks : 10/10

## 2. Problem Statement

Max is fascinated by prime numbers and the Fibonacci sequence. He wants to combine these two interests by creating a program that outputs the first n prime numbers within the Fibonacci sequence.

Your task is to help Max by writing a program that prints the first n prime numbers in the Fibonacci sequence using a while loop along with the break statement to achieve the desired functionality.

### ***Input Format***

The input consists of an integer n, representing the number of prime Fibonacci numbers to generate.

### ***Output Format***

The output displays space-separated first n prime numbers found in the Fibonacci sequence.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 5

Output: 2 3 5 13 89

### ***Answer***

# You are using Python

```
def is_prime(n):
```

```
    if n <= 1:
```

```
        return False
```

```
    if n <= 3:
```

```
        return True
```

```
    if n % 2 == 0 or n % 3 == 0:
```

```
        return False
```

```
    i = 5
```

```
    while i*i<=n:
```

```
        if n % i == 0 or n % (i + 2) == 0:
```

```
            return False
```

```
        i += 6
```

```
    return True
```

```
def fibonacci_primes(n):
```

```
    fib_sequence = [0,1]
```

```
    prime_fibs = []
```

```
    i = 1
```

```
    while True:
```

```
        next_fib = fib_sequence[i] + fib_sequence[i-1]
```

```
        fib_sequence.append(next_fib)
```

```
        if is_prime(next_fib):
```

```
prime_fibs.append(next_fib)
if len(prime_fibs) == n:
    break
i += 1
return prime_fibs
num_primes = int(input())
result = fibonacci_primes(num_primes)
print(*result)
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Taylor is tasked with a mathematical challenge that requires finding the smallest positive number divisible by all integers from 1 to n.

Help Taylor to determine the smallest positive number that is divisible by all integers from 1 to n. Make sure to employ the break statement to ensure efficiency in the program.

#### ***Input Format***

The input consists of a single integer, n.

#### ***Output Format***

The output displays the smallest positive number that is divisible by all integers from 1 to n.

Refer to the sample output for the formatting specifications.

#### ***Sample Test Case***

Input: 10

Output: 2520

#### ***Answer***

```
# You are using Python
def smallest_divisible(n):
```

```

if n <= 1:
    return 1

result = 1
for i in range(2,n+1):
    if result % i != 0:
        for j in range(1,n+1):
            if (result*j)%i==0:
                result = result*j
                break
        return result
n= int(input())
print(smallest_divisible(n))

```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Gabriel is working on a wildlife research project where he needs to compute various metrics for different animals based on their characteristics. Each animal type requires a different calculation: a deer's distance traveled, a bear's weight based on footprint size, or a bird's altitude based on its flying pattern.

Conditions:

For Deer (Mode 'D' or 'd'): Distance = speed of sound \* time taken, where the speed of sound in air is 343 meters per second. For Bear (Mode 'B' or 'b'): Weight = footprint size \* average weight, where the average weight per square inch for a bear is 5.0 pounds. For Bird (Mode 'F' or 'f'): Altitude = flying pattern \* distance covered (in meters).

Write a program to help Gabriel analyze the characteristics of animals based on the given inputs.

#### **Input Format**

The first line of input consists of a character, representing the type of animal 'D/d' for deer, 'B/b' for bear, and 'F/f' for bird.

If the choice is 'D' or 'd':

The second line of input consists of a floating-point value T, representing the time taken from the deer's location to the observer.

If the choice is 'B' or 'b':

The second line of input consists of a floating-point value S, representing the size of the bear's footprint in square inches.

If the choice is 'F' or 'f':

1. The second line of input consists of a floating-point value P, representing the bird's flying pattern.
2. The third line consists of a floating-point value D, representing the distance covered by the bird in meters.

### **Output Format**

The output prints one of the following:

If the choice is 'D' or 'd':

The output prints "Distance: X m" where X is a floating point value rounded off to two decimal places, representing the calculated distance traveled by the sound wave in meters.

If the choice is 'B' or 'b':

The output prints "Weight: Y lb" where Y is a floating point value rounded off to two decimal places, representing the estimated weight of the bear in pounds.

If the choice is 'F' or 'f':

The output prints "Altitude: Z m" where Z is a floating point value rounded off to two decimal places, representing the calculated altitude of the bird's flight in meters.

If the given choice is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

### Sample Test Case

Input: d

2.5

Output: Distance: 857.50 m

### Answer

# You are using Python

```
def analyse_animal():
```

```
    choice = input().strip()
```

```
    if choice in ('D','d'):
```

```
        T= float(input().strip())
```

```
        if 1.0<= T <= 10.0:
```

```
            distance=343*T
```

```
            print(f"Distance: {distance:.2f} m")
```

```
        else:
```

```
            print("Invalid")
```

```
    elif choice in ('B','b'):
```

```
        s= float(input().strip())
```

```
        if 1.0 <= s <= 5.0:
```

```
            weight = s*5.0
```

```
            print(f"Weight:{weight:.2f} lb")
```

```
        else:
```

```
            print("Invalid")
```

```
    elif choice in ('F','f'):
```

```
        P= float(input().strip())
```

```
        D=float(input().strip())
```

```
        if 1.0 <= P <= 50.0 and 1.0 <=D <=50.0:
```

```
            altitude=P*D
```

```
            print(f"Altitude: {altitude:.2f} m")
```

```
        else:
```

```
            print("Invalid")
```

```
    else:
```

```
        print("Invalid")
```

```
analyse_animal()
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_PAH\_Updated

Attempt : 1  
Total Mark : 60  
Marks Obtained : 60

#### Section 1 : Coding

##### 1. 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**

```
# You are using Python
def sum_even_odd(n):
    even_sum=sum(i for i in range(1,n+1)if i%2==0)
    odd_sum=sum(i for i in range(1,n+1)if i%2!=0)
    print("Sum of even numbers from 1 to ",n," is ",even_sum)
    print("Sum of odd numbers from 1 to ",n," is ",odd_sum)
n=int(input())
sum_even_odd(n)
```

**Status :** Correct

**Marks :** 10/10

## **2. 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 cal_bill(units):  
    if units<=100:  
        bills=0  
    elif units<=200:  
        bills=(units-100)*5  
    else:  
        bills=(100*5)+(units-200)*10  
    print("Rs.",bills)  
    units=int(input())  
    cal_bill(units)
```

**Status :** Correct

**Marks :** 10/10

## **3. 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_not_multiples_of_3(start,end):
    total=0
    for i in range(start,end+1):
        if i%3==0:
            continue
        total+=i
    print(total)
start=int(input())
end=int(input())
sum_not_multiples_of_3(start,end)
```

**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***

# You are using Python

```
def find_leap_years(start_year, end_year):  
    for year in range(start_year, end_year + 1):
```

```
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
    print(year)  
start_year = int(input())  
end_year = int(input())  
find_leap_years(start_year, end_year)
```

**Status :** Correct

**Marks :** 10/10

## 5. 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))
n=int(input())
for i in range(1,n+1,2):
    fact=math.factorial(i)
    digit_sum=sum_of_digits(fact)
    print(f"{i}!= {fact}, sum of digits={digit_sum}")
```

**Status :** Correct

**Marks :** 10/10

## 6. 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

```
n=int(input())
steps=0
while n!=1:
    if n%2==0:
        n//=2
    else:
        n=3*n+1
    steps+=1
    if steps>=100:
        print("Exceeded 100 steps.Exiting...")
        break
if steps<100:
    print("steps taken to reach 1:",steps)
```

**Status :** Correct

**Marks : 10/10**