**Task -2.1:**

**Program :**

# Function to count non-repeated digits

def count\_non\_repeated\_digits(n):

num\_str = str(n)

count = 0

# Check each digit

for digit in num\_str:

if num\_str.count(digit) == 1:

count += 1

return count

# Input from user

n = int(input("Enter a number: "))

# Output the result

print("Count of non-repeated digits:", count\_non\_repeated\_digits(n))

**output :**

Enter a number: 292

Count of non-repeated digits: 1

Enter a number: 1015

Count of non-repeated digits: 2

Enter a number: 108

Count of non-repeated digits: 3

**Task 2.1:**

**Program :**

# Read the lengths of the three sides from the user

a = float(input("Enter length of side 1: "))

b = float(input("Enter length of side 2: "))

c = float(input("Enter length of side 3: "))

# Check if the given sides can form a valid triangle

if a + b > c and a + c > b and b + c > a:

# Classify the triangle

if a == b == c:

print("The triangle is Equilateral.")

elif a == b or b == c or a == c:

print("The triangle is Isosceles.")

else:

print("The triangle is Scalene.")

else:

print("The given sides do not form a valid triangle.")

**output :**

Sample Input 1 60 60 60

Sample Output 1 That's a equilateral triangle

Sample Input 2 40 40 8

Sample Output 2 That's a isosceles triangle

Sample Input 3 50 60 70

Sample Output 3 That's a scalene triangle

**Task 2.3 :**

**Program :**

# Read an integer from the user

n = int(input("Enter a number: "))

# Calculate square of the number

square = n \* n

# Check if the square ends with the number itself

if str(square).endswith(str(n)):

print("Automorphic")

else:

print("Not Automorphic")

**output :**

Example input: 5

Output: Automorphic

Example input: 25

Output: Automorphic

Example input: 7

Output: Not Automorphic

**Task 2.4 :**

**Program :**

# Function to check if a number is prime

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

return False

return True

# Input from user

start = int(input("Enter the starting number of the range: "))

end = int(input("Enter the ending number of the range: "))

# Count primes in the given range

prime\_count = 0

for n in range(start, end + 1):

if is\_prime(n):

prime\_count += 1

# Output result

print("Count of prime numbers in the range:", prime\_count)

**output :**

Enter the starting number of the range: 2

Enter the ending number of the range: 20

Count of prime numbers in the range: 8

Enter the starting number of the range: 700

Enter the ending number of the range: 725

Count of prime numbers in the range: 3

Task 2.5 :

Program :

# Read number of terms from user

n = int(input("Enter the number of terms: "))

# First two terms

a, b = 0, 1

print("Fibonacci Series:")

# Generate Fibonacci series

for i in range(n):

print(a, end=" ")

 a, b = b, a + b

**output :**

Sample input/output Enter the number of terms: 6 Fibonacci series: 0 1 1 2 3 5