**LAB CYCLE 1**

1. Program to Print all non-Prime Numbers in an Interval

**Code:**

def is\_prime(num):

if num <= 1:

return False

for i in range(2, num):

if num % i == 0:

return False

return True

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

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

end = int(input("Enter the end number: "))

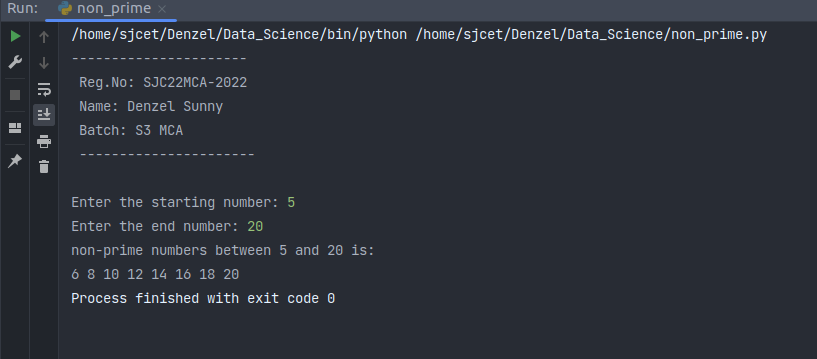
print(f"non-prime numbers between {start} and {end} is: ")

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

if not is\_prime(num):

print(num, end=" ")

**Output:**



2. Program to print the first N Fibonacci numbers.

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

n1, n2 =0, 1

limit = int(input("Enter the range: "))

print("Fibonacci series: ")

print(n1, n2, end=" ")

for i in range(2, limit):

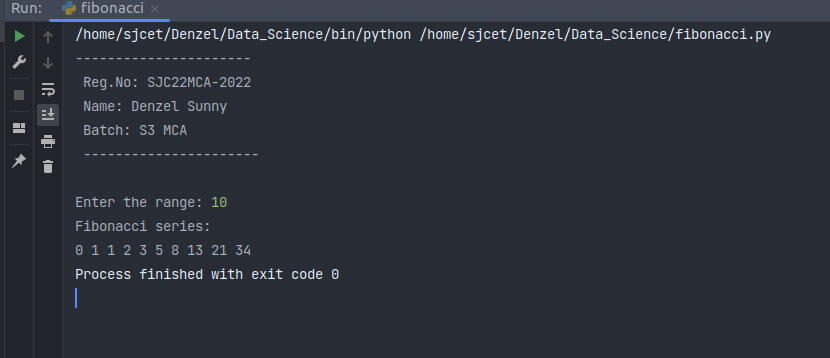
n3 = n1 + n2

n1 = n2

n2 = n3

print(n3, end=" ")

**Output:**



3. Given sides of a triangle, write a program to check whether a given triangle is an isosceles, equilateral or scalene.

**Code:**

def checkTriangle(x, y, z):

if x == y == z:

print("Given Triangle is a Equilateral Triangle")

elif x == y or y == z or z == x:

print("Given Triangle is a Isosceles Triangle")

else:

print("Given Triangle is a Scalene Triangle")

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

print("enter sides of the Triangle")

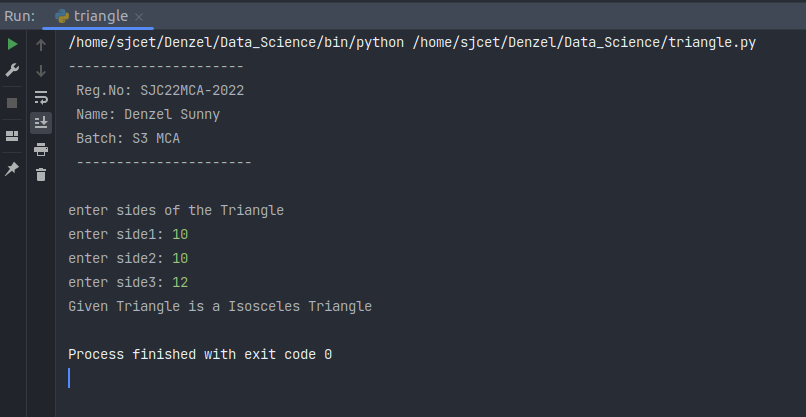
side1=int(input("enter side1: "))

side2=int(input("enter side2: "))

side3=int(input("enter side3: "))

checkTriangle(side1, side2, side3)

**Output:**



4. Program to check whether given pair of number is coprime

**Code:**

from math import gcd

def co\_prime(a, b):

if gcd(a, b) == 1:

print(f"the numbers {a} and {b} are coprime")

else:

print("not co-prime")

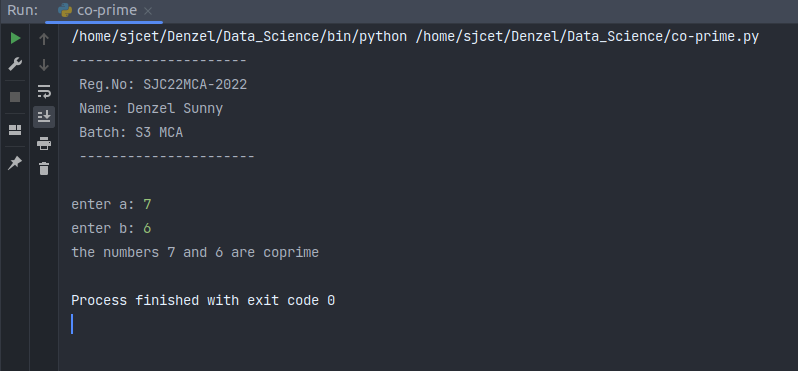
print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

a=int(input("enter a: "))

b=int(input("enter b: "))

co\_prime(a,b)

**Output:**



5. Program to find the roots of a quadratic equation(rounded to 2 decimal places)

**Code:**

import math

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

a = float(input("Enter the coefficient a: "))

b = float(input("Enter the coefficient b: "))

c = float(input("Enter the coefficient c: "))

discriminant = b\*\*2 - 4\*a\*c

if discriminant > 0:

root1 = (-b + math.sqrt(discriminant)) / (2\*a)

root2 = (-b - math.sqrt(discriminant)) / (2\*a)

print(f"Root 1: {round(root1, 2)}")

print(f"Root 2: {round(root2, 2)}")

elif discriminant == 0:

root = -b / (2\*a)

print(f"Root: {round(root, 2)}")

else:

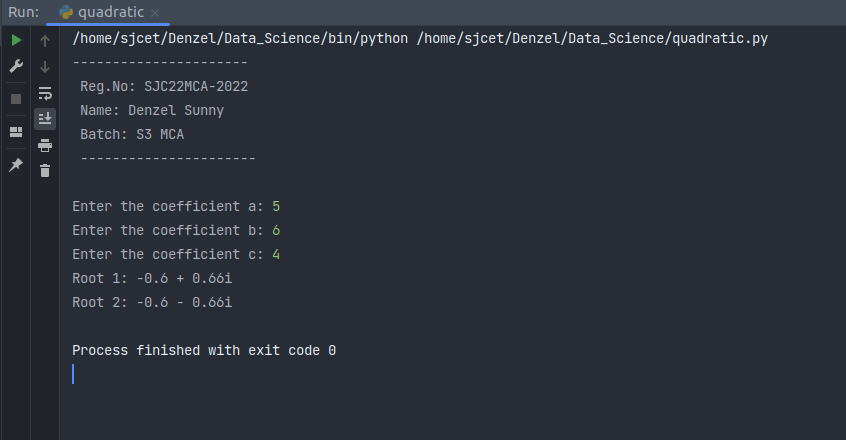
real = -b / (2\*a)

imag = math.sqrt(-discriminant) / (2\*a)

print(f"Root 1: {round(real, 2)} + {round(imag, 2)}i")

print(f"Root 2: {round(real, 2)} - {round(imag, 2)}i")

**Output:**



6. Program to check whether a given number is perfect number or not(sum of factors =number)

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

n = int(input("enter the number: "))

sum = 0

for i in range(1, n):

if n % i == 0:

sum = sum + i

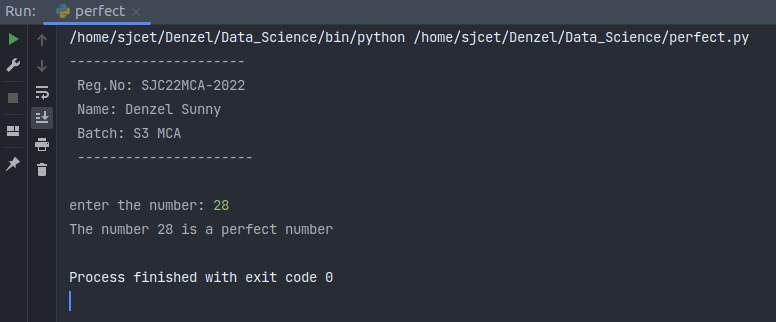
if sum == n:

print(f"The number {n} is a perfect number")

else:

print(f"The number {n} is not a perfect number")

**Output:**



7. Program to display armstrong numbers up to 1000

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

t1=1

t2=1000

for num in range(t1, t2+1):

order = len(str(num))

sum = 0

temp = num

while temp>0:

digit = temp % 10

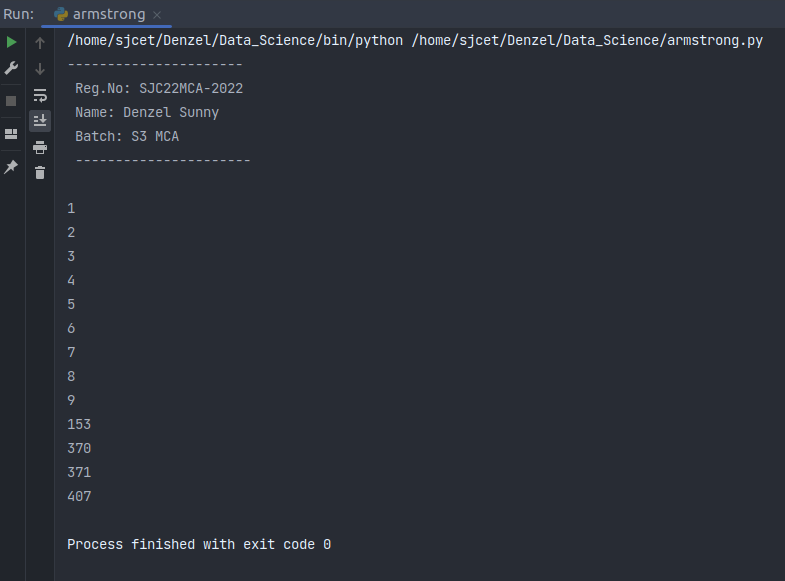
sum += digit \*\* order

temp //= 10

if num == sum:

print(num)

**Output:**



8. Store and display the days of a week as a List, Tuple, Dictionary, Set. Also demonstrate different ways to store values in each of them. Display its type also.

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

days\_list = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

print("List:", days\_list)

print("Type of List:", type(days\_list))

days\_tuple = ("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")

print("Tuple:", days\_tuple)

print("Type of Tuple:", type(days\_tuple))

days\_dict = {1: "Monday", 2: "Tuesday", 3: "Wednesday", 4: "Thursday", 5: "Friday", 6: "Saturday", 7: "Sunday"}

print("Dictionary:", days\_dict)

print("Type of Dictionary:", type(days\_dict))

days\_set = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"}

print("Set:", days\_set)

print("Type of Set:", type(days\_set))

**Output:**



9. Write a program to add elements of given 2 lists

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

list1 = [2, 4, 6, 8, 10]

list2 = [1, 3, 5, 7, 9]

print("The First list is:", list1)

print("The Second list is:", list2)

if len(list1) != len(list2):

print("Lists must have the same length for addition.")

else:

result = []

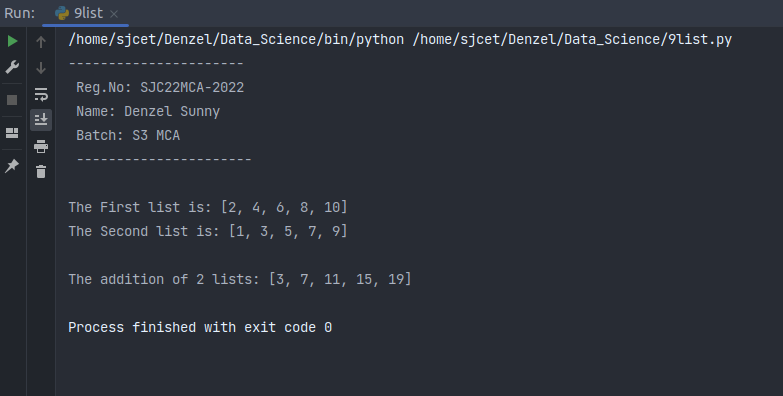
for i in range(len(list1)):

sum\_element = list1[i] + list2[i]

result.append(sum\_element)

print("\nThe addition of 2 lists:", result)

**Output:**



10. Write a program to find the sum of 2 matrices using a nested List.

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

rows = int(input("enter number of rows: "))

columns = int(input("enter number of columns: "))

matrix1 = []

matrix2 = []

print("enter elements of matrix1: ")

for i in range(rows):

row = []

for j in range(columns):

element = int(input(f"enter elements at row {i + 1}, column {j + 1}: "))

row.append(element)

matrix1.append(row)

print("enter elements of matrix2: ")

for i in range(rows):

row = []

for j in range(columns):

element = int(input(f"enter elements at row {i + 1}, column {j + 1}: "))

row.append(element)

matrix2.append(row)

result\_matrix = []

for i in range(rows):

row = []

for j in range(columns):

element = matrix1[i][j] + matrix2[i][j]

row.append(element)

result\_matrix.append(row)

print("sum of the two matrices: ")

for row in result\_matrix:

print(row)

**Output:**



11. Write a program to perform bubble sort on a given set of elements.

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

for j in range(0, n-i-1):

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

input\_str = input("Enter elements separated by spaces: ")

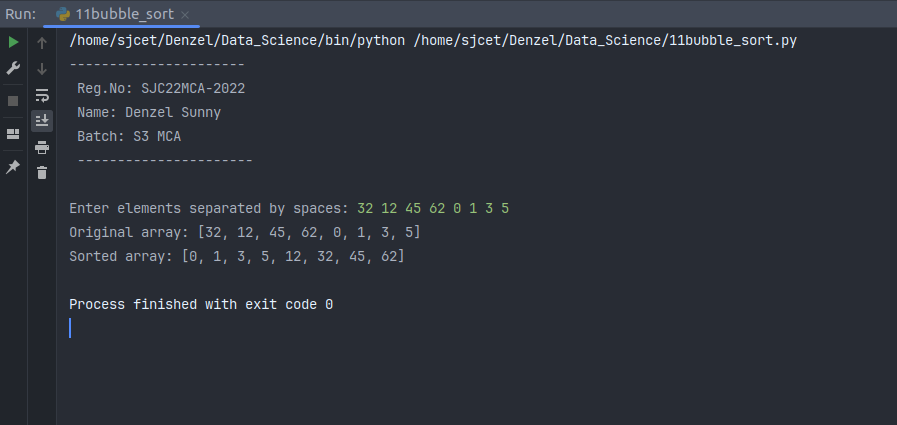
elements = [int(x) for x in input\_str.split()]

print("Original array:", elements)

bubble\_sort(elements)

print("Sorted array:", elements)

**Output:**



12. Program to find the count of each vowel in a string(use dictionary)

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

def count\_vowels(string):

vowel\_count = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0}

string = string.lower()

for char in string:

if char in vowel\_count:

vowel\_count[char] += 1

return vowel\_count

input\_string = input("Enter a string: ")

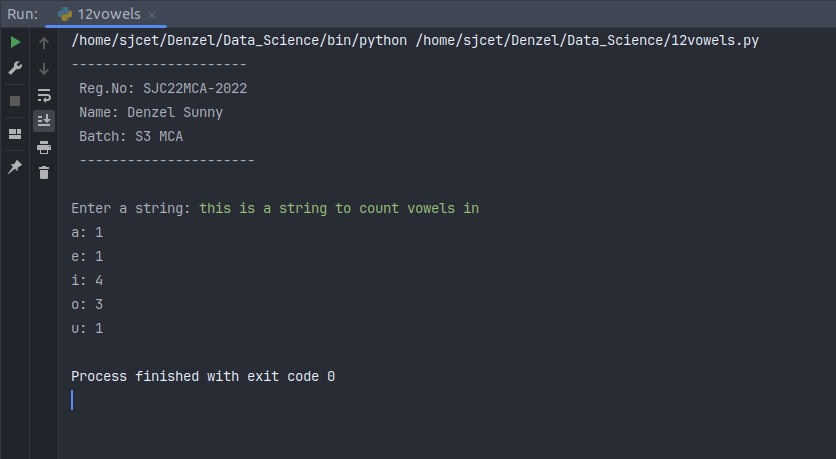
result = count\_vowels(input\_string)

# Print the count of each vowel

for vowel, count in result.items():

print(f'{vowel}: {count}')

**Output:**



13. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive(eg: 256->2+5+6=13

256-13=243

243-9=232……..

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

def sum\_of\_digits(n):

digit\_sum = 0

while n > 0:

digit\_sum += n % 10

n //= 10

return digit\_sum

num = int(input("Enter a positive number: "))

while num > 0:

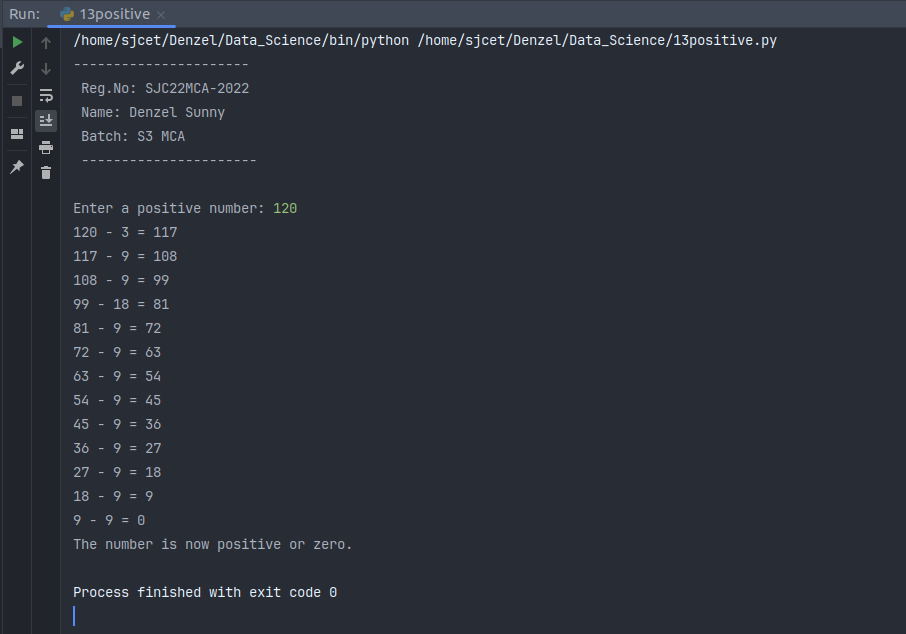
digit\_sum = sum\_of\_digits(num)

num -= digit\_sum

print(f"{num + digit\_sum} - {digit\_sum} = {num}")

print("The number is now positive or zero.")

**Output:**



14. Write a Python program that accepts a 10 digit mobile number, and find the digits which are absent in a given mobile number.

**Code:**

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

def find\_absent\_digits(mobile\_number):

all\_digits = set("0123456789")

number\_digits = set(mobile\_number)

absent\_digits = all\_digits - number\_digits

return absent\_digits

mobile\_number = input("Enter a 10-digit mobile number: ")

if len(mobile\_number) == 10 and mobile\_number.isdigit():

absent\_digits = find\_absent\_digits(mobile\_number)

if absent\_digits:

print("Absent digits:", ', '.join(absent\_digits))

else:

print("All digits are present in the mobile number.")

else:

print("Invalid input. Please enter a 10-digit mobile number.")

**Output:**

