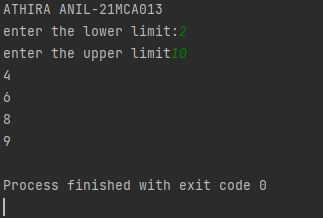
**LAB CYCLE 1**

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

n1= int(input("enter the lower limit:"))  
n2=int(input("enter the upper limit"))  
for i in range(n1,n2):  
 for j in range(2,i):  
 if i % j == 0:  
 print(i)  
 break

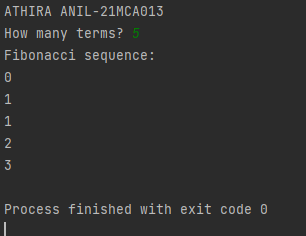
**output**



2. Program to print the first N Fibonacci numbers.

nterms = int(input("How many terms? "))  
  
n1, n2 = 0, 1  
count = 0  
  
if nterms <= 0:  
 print("Please enter a positive integer")  
  
elif nterms == 1:  
 print("Fibonacci sequence upto",nterms,":")  
 print(n1)  
  
else:  
 print("Fibonacci sequence:")  
 while count < nterms:  
 print(n1)  
 nth = n1 + n2  
  
 n1 = n2  
 n2 = nth  
 count += 1

**output**

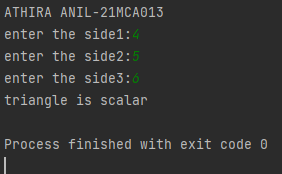


3. Given sides of a triangle, write a program to check whether given triangle is an

isosceles, equilateral or scalene.

n1 = int(input("enter the side1:"))  
n2 = int(input("enter the side2:"))  
n3 = int(input("enter the side3:"))  
if n1 == n2 and n1 == n3:  
 print("triangle is equilatrel")  
elif n1 == n2 or n1 == n3 or n1 == n3:  
 print("triangle is isoceless")  
else:  
 print("triangle is scalar")

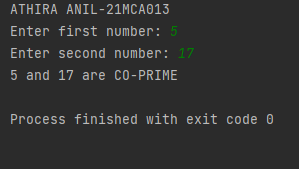
**output**



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

def are\_coprime(a, b):  
 hcf = 1  
  
 for i in range(1, a + 1):  
 if a % i == 0 and b % i == 0:  
 hcf = i  
  
 return hcf == 1  
first = int(input('Enter first number: '))  
second = int(input('Enter second number: '))  
  
if are\_coprime(first, second):  
 print('%d and %d are CO-PRIME' % (first, second))  
else:  
 print('%d and %d are NOT CO-PRIME' % (first, second))

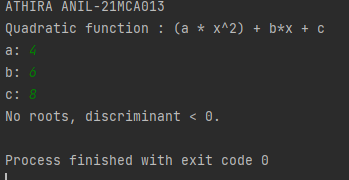
**output**



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

from math import sqrt  
print("Quadratic function : (a \* x^2) + b\*x + c")  
a = float(input("a: "))  
b = float(input("b: "))  
c = float(input("c: "))  
r = b\*\*2 - 4\*a\*c  
if r > 0:  
 num\_roots = 2  
 x1 = (((-b) + sqrt(r))/(2\*a))  
 x2 = (((-b) - sqrt(r))/(2\*a))  
 print("There are 2 roots: %f and %f" % (x1, x2))  
elif r == 0:  
 num\_roots = 1  
 x = (-b) / 2\*a  
 print("There is one root: ", x)  
else:  
 num\_roots = 0  
 print("No roots, discriminant < 0.")  
 exit()

**output**

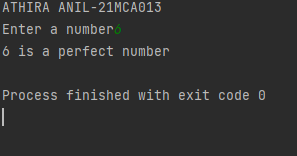


6. Program to check whether a given number is perfect number or not(sum of factors

=number)

num = int(input("Enter a number"))  
sum = 0  
for i in range (1,num):  
 if(num % i == 0):  
 sum = sum + i  
if(sum == num):  
 print(num,"is a perfect number")  
else:  
 print(num,"is not a perfect number")

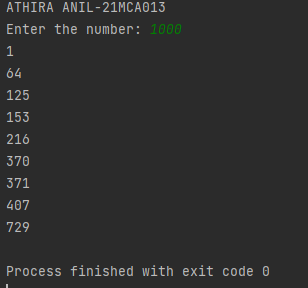
**output**



7. Program to display amstrong numbers upto 1000

n = int(input("Enter the number: "))  
s=0  
for num in range(s, n + 1):  
 sum = 0  
 temp = num  
 while temp > 0:  
 digit = temp % 10  
 sum += digit \*\* 3  
 temp //= 10  
 if num == sum:  
 print(num)

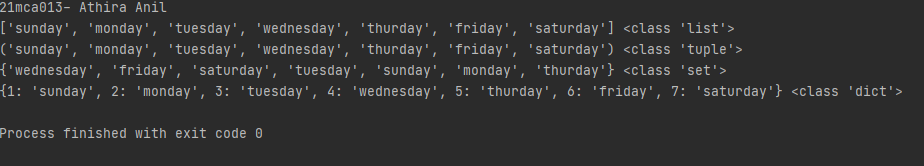
**output**



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.

print("21mca013- Athira Anil")  
list1=["sunday","monday","tuesday","wednesday","thurday","friday","saturday"]  
tuple=("sunday","monday","tuesday","wednesday","thurday","friday","saturday")  
set={"sunday","monday","tuesday","wednesday","thurday","friday","saturday"}  
dict={1:"sunday",2:"monday",3:"tuesday",4:"wednesday",5:"thurday",6:"friday",7:"saturday"}  
print(list1,type(list1))  
print(tuple,type(tuple))  
print(set,type(set))  
print(dict,type(dict))

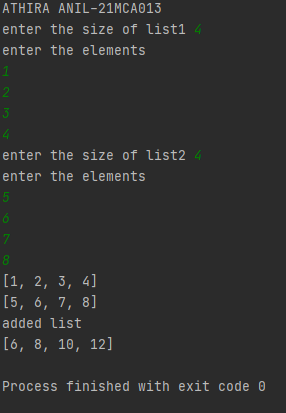
**output**



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

n1=int(input("enter the size of list1"))  
list1=[]  
print("enter the elements")  
for i in range(0,n1):  
 ele=int(input())  
 list1.append(ele)  
  
n2=int(input("enter the size of list2"))  
list2=[]  
print("enter the elements")  
for i in range(0,n2):  
 ele=int(input())  
 list2.append(ele)  
print(list1)  
print(list2)  
list3=[]  
if n1 == n2:  
 print("added list")  
 for i in range(0,n1):  
 ele=list1[i]+list2[i]  
 list3.append(ele)  
 print(list3)  
else:  
 print("lists are of different size,addition not possible")

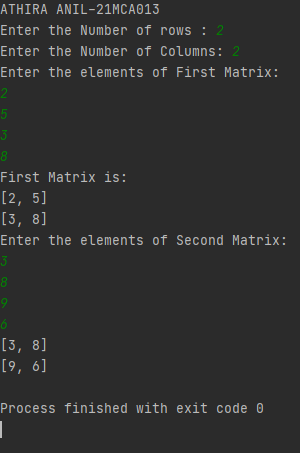
**Output**



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

rows = int(input("Enter the Number of rows : "))  
column = int(input("Enter the Number of Columns: "))  
print("Enter the elements of First Matrix:")  
matrix\_a = [[int(input()) for i in range(column)] for i in range(rows)]  
print("First Matrix is: ")  
for n in matrix\_a:  
 print(n)  
print("Enter the elements of Second Matrix:")  
matrix\_b = [[int(input()) for i in range(column)] for i in range(rows)]  
for n in matrix\_b:  
 print(n)  
result = [[0 for i in range(column)] for i in range(rows)]  
for i in range(rows):  
 for j in range(column):  
 result[i][j] = matrix\_a[i][j] + matrix\_b[i][j]

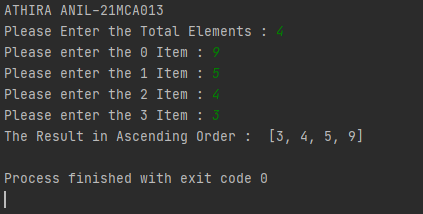
**output**



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

a = []  
number = int(input("Please Enter the Total Elements : "))  
for i in range(number):  
 value = int(input("Please enter the %d Item : " %i))  
 a.append(value)  
for i in range(number -1):  
 for j in range(number - i - 1):  
 if(a[j] > a[j + 1]):  
 temp = a[j]  
 a[j] = a[j + 1]  
 a[j + 1] = temp  
print("The Result in Ascending Order : ", a)

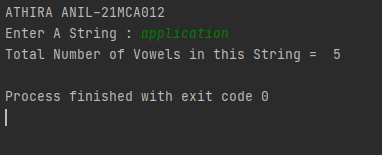
**output**



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

str1 = input("Enter A String : ")  
vowels = 0  
  
for i in str1:  
 if (i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u'  
or i == 'A' or i == 'E' or i == 'I' or i == 'O' or i == 'U'):  
 vowels = vowels + 1  
print("Total Number of Vowels in this String = ", vowels)

**output**



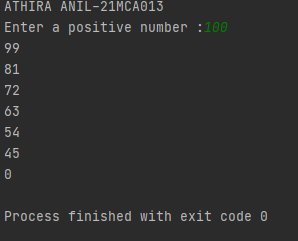
13.Write a Python program that accepts a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive(eg: 256->2+5+6=13

                       256-13=243

                       243-9=232……..

num=int(input("Enter a positive number :"))  
digsum=0  
new\_num=num  
while new\_num >= digsum:  
 list1 = [int(x) for x in str(new\_num)]  
 for i in list1:  
 digsum=digsum+i  
 new\_num=num-digsum  
 print(new\_num)  
print(new\_num-new\_num)

**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

num = int(input("Enter a 10 digit mobile number : "))  
nums = []  
for i in range(0, 10):  
 n = num % 10  
 nums.append(n)  
 num = num // 10  
print("numbers not in the mobile number are : ")  
for i in range(0, 10):  
 if i not in nums:  
 print(i)

**output**

