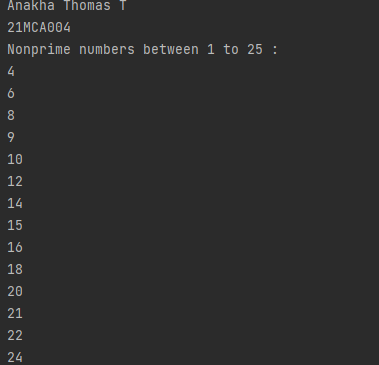
DATA SCEINCE & MACHINE LEARNING:

LAB CYCLE 1

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

print("Anakha Thomas T")  
print("21MCA004")  
import math  
def is\_not\_prime(n):  
 ans = False  
 for i in range(2, int(math.sqrt(n)) + 1):  
 if n % i == 0:  
 ans = True  
 return ans  
print("Nonprime numbers between 1 to 100:")  
for x in filter(is\_not\_prime, range(1, 101)):  
 print(x)

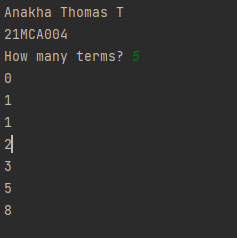
**OUTPUT:**



1. Program to print the first N Fibonacci numbers.

print("Anakha Thomas T")  
print("21MCA004")  
nterms = int(input("How many terms? "))  
n1, n2 = 0, 1  
count = 0  
print(n1)  
print(n2)  
for i in range (0,nterms):  
 count=n1+n2  
 print(count)  
 n1=n2  
 n2=count

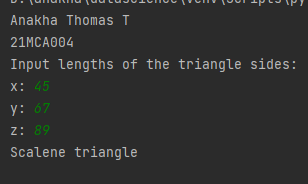
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
print("Input lengths of the triangle sides: ")  
x = int(input("x: "))  
y = int(input("y: "))  
z = int(input("z: "))  
  
if x == y == z:  
 print("Equilateral triangle")  
elif x==y or y==z or z==x:  
 print("isosceles triangle")  
else:  
 print("Scalene triangle")

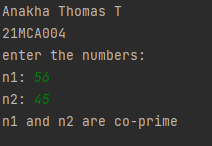
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
print("enter the numbers: ")  
n1 = int(input("n1: "))  
n2 = int (input("n2: "))  
for i in range(1,n1):  
 if n1%i==0 and n2%i==0:  
 hcf=i  
if hcf==1:  
 print("n1 and n2 are co-prime")  
else:  
 print("n1 nd n2 are not co-prime")

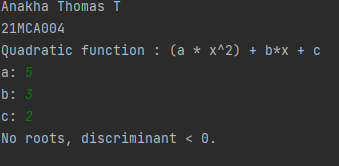
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
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:

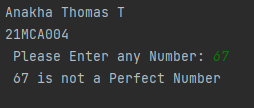


6. Program to check whether a given number is perfect number or not (sum of factors

=number)

print("Anakha Thomas T")  
print("21MCA004")  
Number = int(input(" Please Enter any Number: "))  
Sum = 0  
for i in range(1, Number):  
 if(Number % i == 0):  
 Sum = Sum + i  
if (Sum == Number):  
 print(" %d is a Perfect Number" %Number)  
else:  
 print(" %d is not a Perfect Number" %Number)

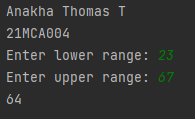
output:



7.Program to display Armstrong numbers unto 1000

print("Anakha Thomas T")  
print("21MCA004")  
lower = int(input("Enter lower range: "))  
upper = int(input("Enter upper range: "))  
  
for num in range(lower, upper + 1):  
 sum = 0  
 temp = num  
 while temp > 0:  
 digit = temp % 10  
 sum += digit \*\* 3  
 temp //= 10  
 if num == sum:  
 print(num)

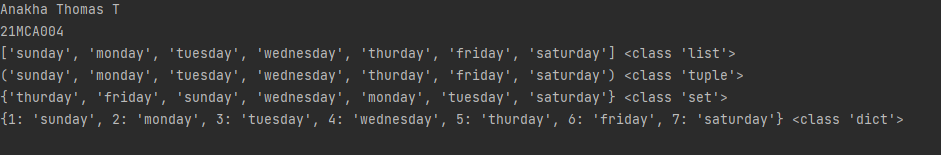
output:



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.

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))

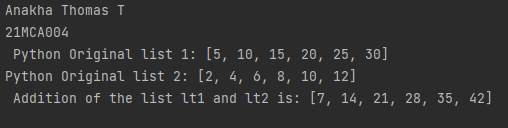
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
l1 = [5, 10, 15, 20, 25, 30]  
l2 = [2, 4, 6, 8, 10, 12]  
print(" Python Original list 1: " + str(l1))  
print("Python Original list 2: " + str(l2))  
res\_lt = []  
for x in range(0, len(l1)):  
 res\_lt.append(l1[x] + l2[x])  
print(" Addition of the list lt1 and lt2 is: " + str(res\_lt))

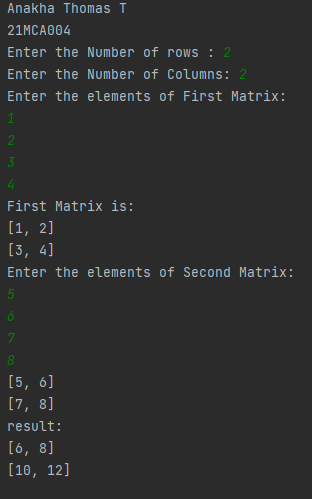
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
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]  
print("result: ")  
for r in result:  
 print(r)

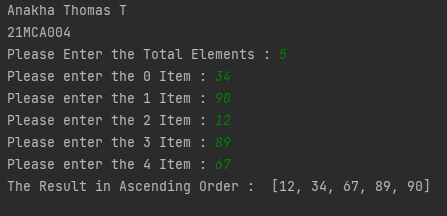
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
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)

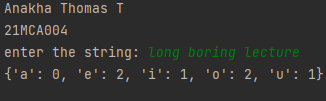
output:



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

print("Anakha Thomas T")  
print("21MCA004")  
a= input("enter the string: ")  
a= a.casefold()  
count= {x:sum([1 for char in a if char == x]) for x in 'aeiou'}  
print(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

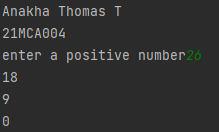
Positive (e.g.: 256->2+5+6=13

256-13=243

243-9=232……..

print("Anakha Thomas T")  
print("21MCA004")  
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.

print("Anakha Thomas T")  
print("21MCA004")  
def absent\_digits(n):  
 all\_nums = set([0,1,2,3,4,5,6,7,8,9])  
 n = set([int(i) for i in n])  
 n = n.symmetric\_difference(all\_nums)  
 n = sorted(n)  
 return n  
print(absent\_digits([9,8,3,2,2,0,9,7,6,3]))

output:

