**Data Science**

**Lab cycle 1**

**Submitted by**

**Tonu John Varghese**

**S3MCA**

**20MCA054**

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

**Program**

first = int(input("enter lower range"))

last = int(input("enter the last number"))

for n in range(first,last+1):

if n > 1:

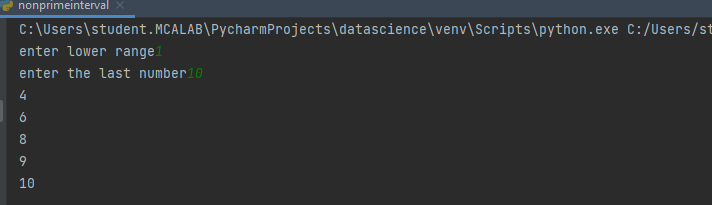
for i in range(2,n):

if (n % i) == 0:

print(n)

break

**output**



1. **Program to print the first N Fibonacci numbers.**

**Program**

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

f1 = 0

f2 = 1

if (n < 1):

print(f1)

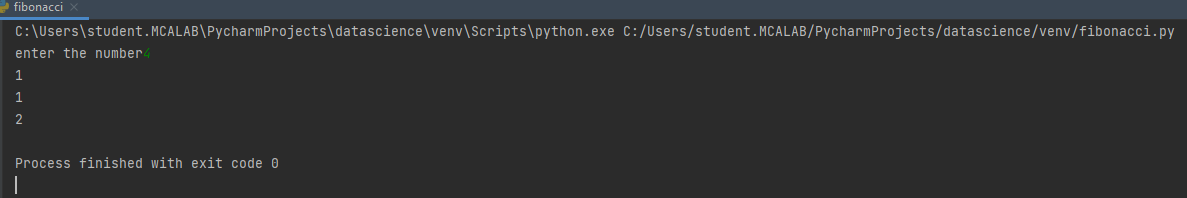
for x in range(1,n):

print(f2)

f3 = f1 + f2

f1 = f2

**output**



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

**Program**

x = int(input("enter first side"))

y = int(input("enter second side"))

z = int(input("enter thrid side"))

if x == y == z:

print("Equilateral Triangle")

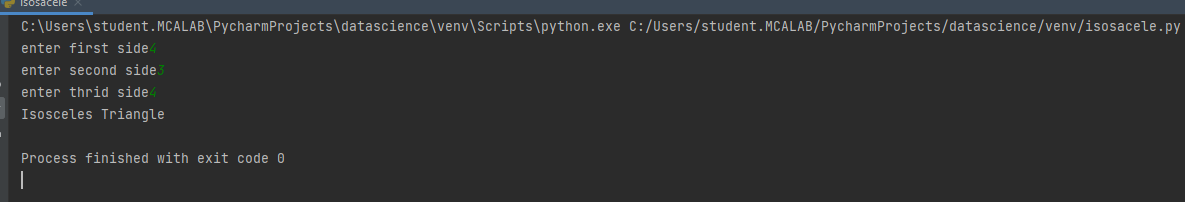
elif x == y or y == z or z == x:

print("Isosceles Triangle")

else:

print("Scalene Triangle")

**output**



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

**Program**

def coprime(a,b):

f=1

for x in range(1,a+1):

if a%x==0 and b%x==0:

f=1

return f==1

n1=int(input("Enter the First number:"))

n2=int(input("Enter the Second number:"))

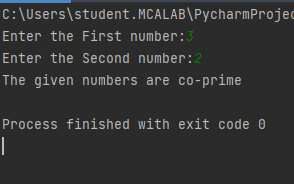
if coprime(n1,n2):

print("The given numbers are co-prime")

else:

print("The given numbers are not coprime")

**Output**



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

**Program**

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:

x1 = (((-b) + sqrt(r)) / (2 \* a))

x2 = (((-b) - sqrt(r)) / (2 \* a))

print("first root ",round(x1,2))

print("second root ", round(x2, 2))

elif r == 0:

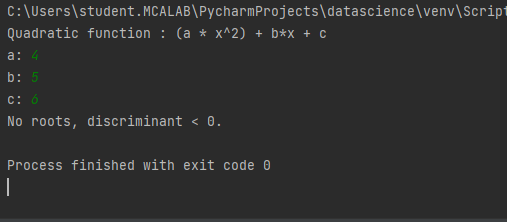
x = (-b) / 2 \* a

print("There is one root: ", x)

else:

print("No roots, discriminant < 0.")

**output**



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

a = int(input("enter the number"))

sum =0

for i in range (1,a):

if (a%i==0):

sum = sum + i

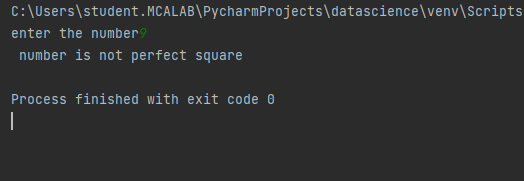
if (sum == a):

print(" number is a perfect square")

else:

print(" number is not perfect square")

**output**



1. **Program to display amstrong numbers upto 1000**

**Program**

for num in range(0,1000):

temp=num

sum=0

while temp > 0:

digit = temp % 10

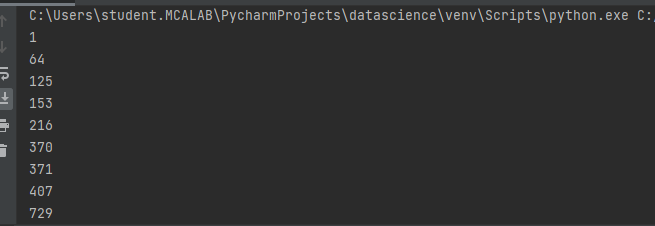
sum = sum + digit \*\* 3

temp = temp // 10

if sum == num:

print(num)

**output**



1. **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.**

**Program**

list = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"]

print(type(list))

print(list)

tuple = ("Sun","Mon","Tue","Wed","Thu","Fri","Sat")

print(type(tuple))

print(tuple)

set = {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}

print(type(set))

print(set)

dict = {

"d1" : "Sun",

"d2" : "Mon",

"d3" : "Tue",

"d4" : "Wed",

"d5" : "Thu",

"d6" : "Fri",

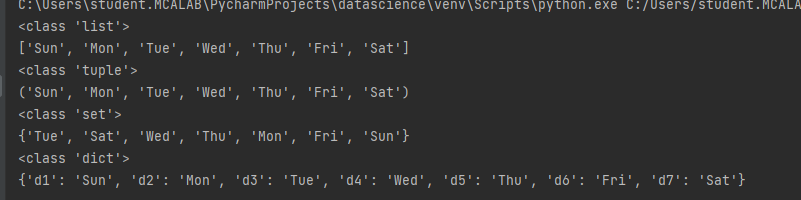
"d7" : "Sat"

}

print(type(dict))

print(dict)

**output**



1. **Write a program to add elements of  given 2 lists**

**Program**

NumList1 = [20, 30, 40, 50]

NumList2 = [45, 25, 35, 45]

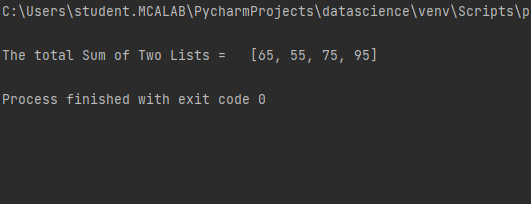
total = []

for j in range(4):

total.append(NumList1[j] + NumList2[j])

print("\nThe total Sum of Two Lists = ", total)

**output**



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

**Program**

x = [[1,12,2],

[4,5,12],

[8,10,15]]

y = [[1,12,2],

[4,5,12],

[8,10,15]]

res = [[0,0,0],

[0,0,0],

[0,0,0]]

for i in range(len(x)):

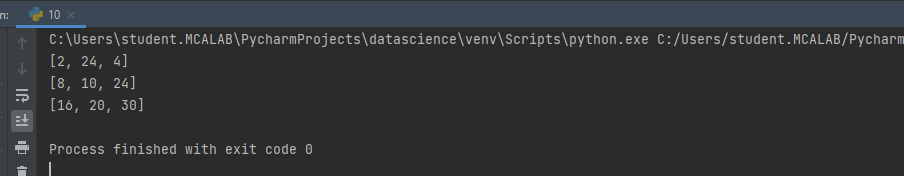
for j in range(len(x[0])):

res[i][j] = x[i][j] + y[i][j]

for r in res:

print(r)

**output**



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

**Program**

a = [35, 10, 31, 11, 26]

print("Before sorting array elements are - ")

for i in a:

print(i, end = " ")

for i in range(0,len(a)):

for j in range(i+1,len(a)):

if a[j]<a[i]:

temp = a[j]

a[j]=a[i]

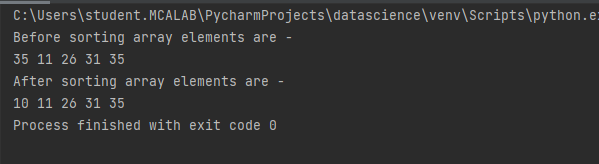
a[i]=temp

print("\nAfter sorting array elements are - ")

for i in a:

print(i, end = " ")

**output**



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

**Program**

string=input("Enter string:")

vowels=0

for i in string:

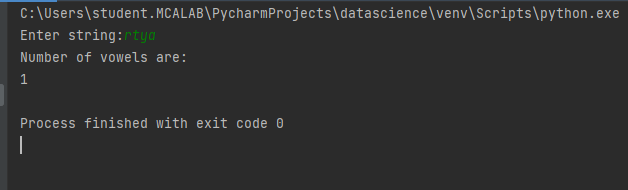
if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):

vowels=vowels+1

print("Number of vowels are:")

print(vowels)

**output**



1. **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**

**Program**

def repeat\_times(n):

s = 0

n\_str = str(n)

while (n > 0):

n -= sum([int(i) for i in list(n\_str)])

n\_str = list(str(n))

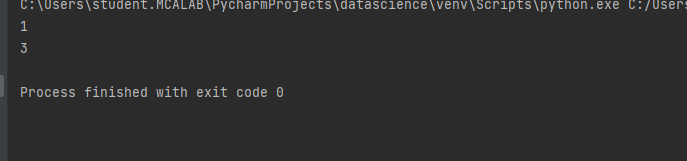
s += 1

return s

print(repeat\_times(9))

print(repeat\_times(21))

**output**



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

**Program**

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([9,5,2,6,0,1,4,6,8,4]))

**output**

