**DATA SCEINCE & MACHINE LEARNING:**

**LAB CYCLE 1**

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

**Program**

first=int(input("Enter the First Limit:"))

last=int(input("Enter the last limit"))

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

if num > 1:

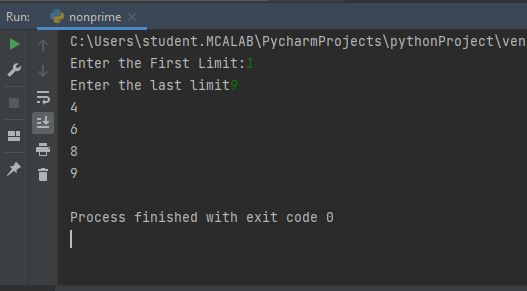
for i in range(2,num):

if(num % i == 0):

print(num)

break

**Output**



1. Program to print the first N Fibonacci numbers.

**Program**

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)

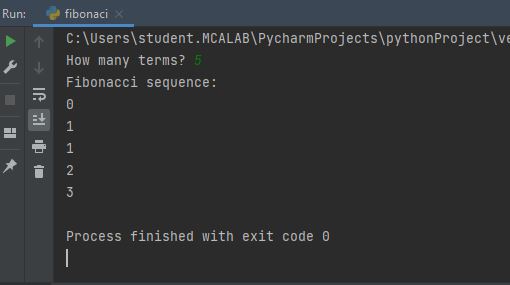
nth = n1 + n2

n1 = n2

n2 = nth

count += 1

**Output**



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

**Program**

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")

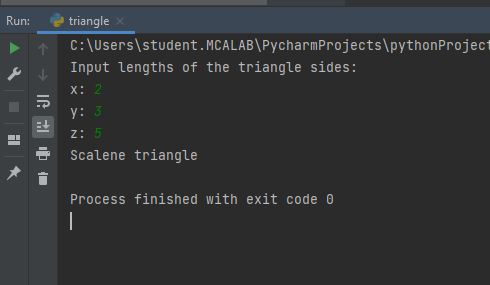
elif x==y or y==z or z==x:

print("isosceles triangle")

else:

print("Scalene triangle")

**Output**



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

**Program**

def gcd(p,q):

while q != 0:

p, q = q, p%q

return p

def is\_coprime(x, y):

return gcd(x, y) == 1

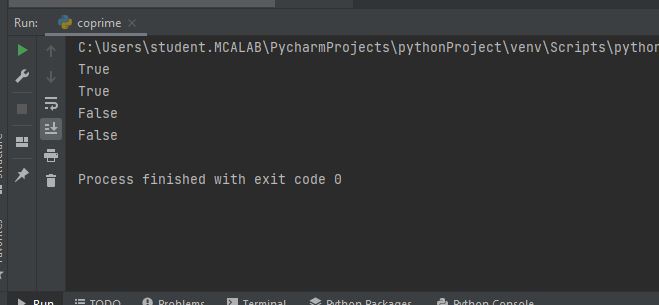
print(is\_coprime(17, 13))

print(is\_coprime(17, 21))

print(is\_coprime(15, 21))

print(is\_coprime(25, 45))

**Output**



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

**Program**

import cmath

a = int(input("Enter the value of a :"))

b = int(input("Enter the value of b :"))

c = int(input("Enter the value of c :"))

# calculate the discriminant

d = (b\*\*2) - (4\*a\*c)

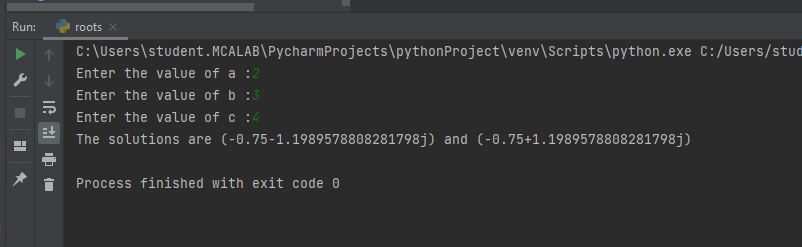
# find two solutions

sol1 = (-b-cmath.sqrt(d))/(2\*a)

sol2 = (-b+cmath.sqrt(d))/(2\*a)

print('The solutions are {0} and {1}'.format(sol1,sol2))

**Output**



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

**Program**

n = int(input(" Please Enter any Number: "))

Sum = 0

for i in range(1, n):

if(n % i == 0):

Sum = Sum + i

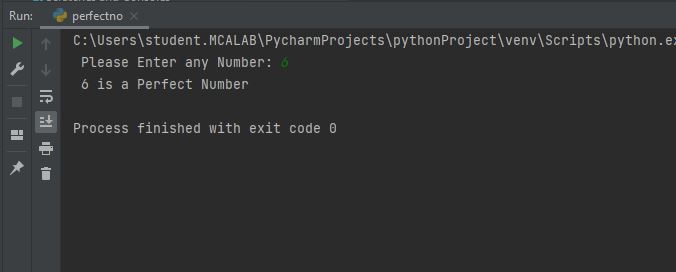
if (Sum == n):

print(" %d is a Perfect Number" %n)

else:

print(" %d is not a Perfect Number" %n)

**Output**



1. Program to display amstrong numbers upto 1000

**Program**

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

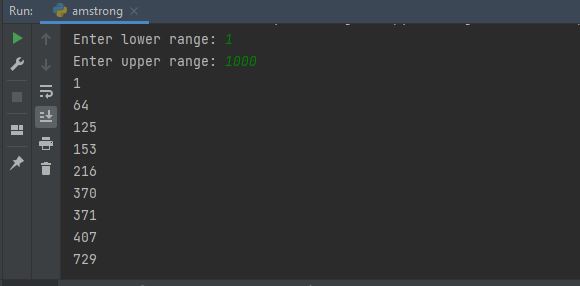
sum += digit \*\* 3

temp //= 10

if num == sum:

print(num)

**Output**



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",

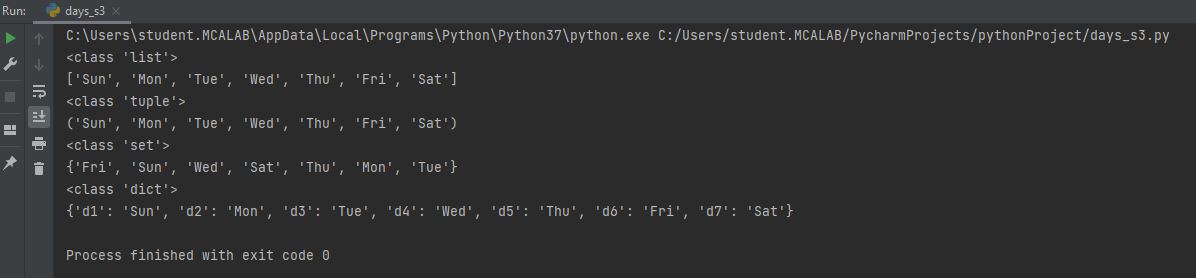
"d7" : "Sat"

}

print(type(dict))

print(dict)

**Output**

****

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

**Program**

lst1 = []

n1 = int(input("Enter number of elements : "))

print("Enter elements in to list 1:")

for i in range(0, n1):

ele = int(input())

lst1.append(ele)

print("List 1: ",lst1)

lst2 = []

n2 = int(input("Enter number of elements : "))

print("Enter elements in to list 1:")

for i in range(0, n2):

ele = int(input())

lst2.append(ele)

print("List 2: ",lst2)

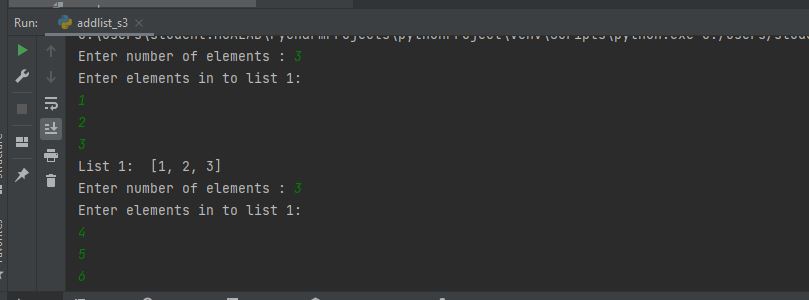
res\_lt = []

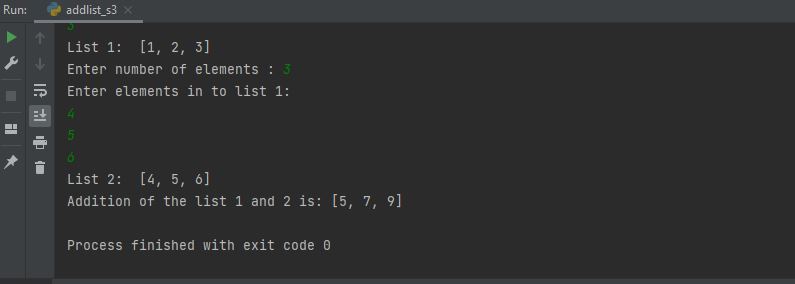
for x in range(0, len(lst1)):

res\_lt.append(lst1[x] + lst2[x])

print("Addition of the list 1 and 2 is: " + str(res\_lt))

**Output**





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

**Program**

a = []

a1 = [2, 4, 6]

a.append(a1)

a2 = [4, 5, 6]

a.append(a2)

print(a)

b = []

b1 = [8, 10, 12]

b.append(b1)

b2 = [2, 4, 2]

b.append(b2)

print(b)

result = [[0, 0, 0],

[0, 0, 0]]

print("Resultant matrix : ")

for i in range(len(a)):

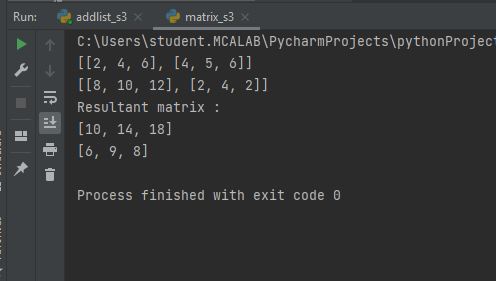
for j in range(len(a[0])):

result[i][j] = a[i][j] + b[i][j]

for r in result:

print(r)

**Output**



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

Program

def bubbleSort(arr):

for i in range(n - 1):

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

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

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

arr = []

n = int(input("Enter limit :"))

print("Enter elements :")

for i in range(0, n):

arr.append(int(input()))

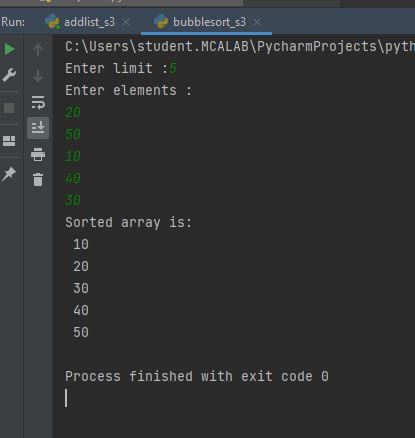
bubbleSort(arr)

print("Sorted array is:")

for i in range(len(arr)):

print("% d" % arr[i])

**Output**



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

**Program**

vowels = 'aeiou'

ip\_str = 'hai welcome'

ip\_str = ip\_str.casefold()

count = {}.fromkeys(vowels,0)

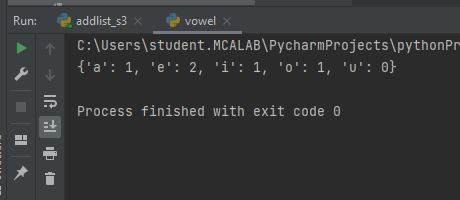
for char in ip\_str:

if char in count:

count[char] += 1

print(count)

**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))

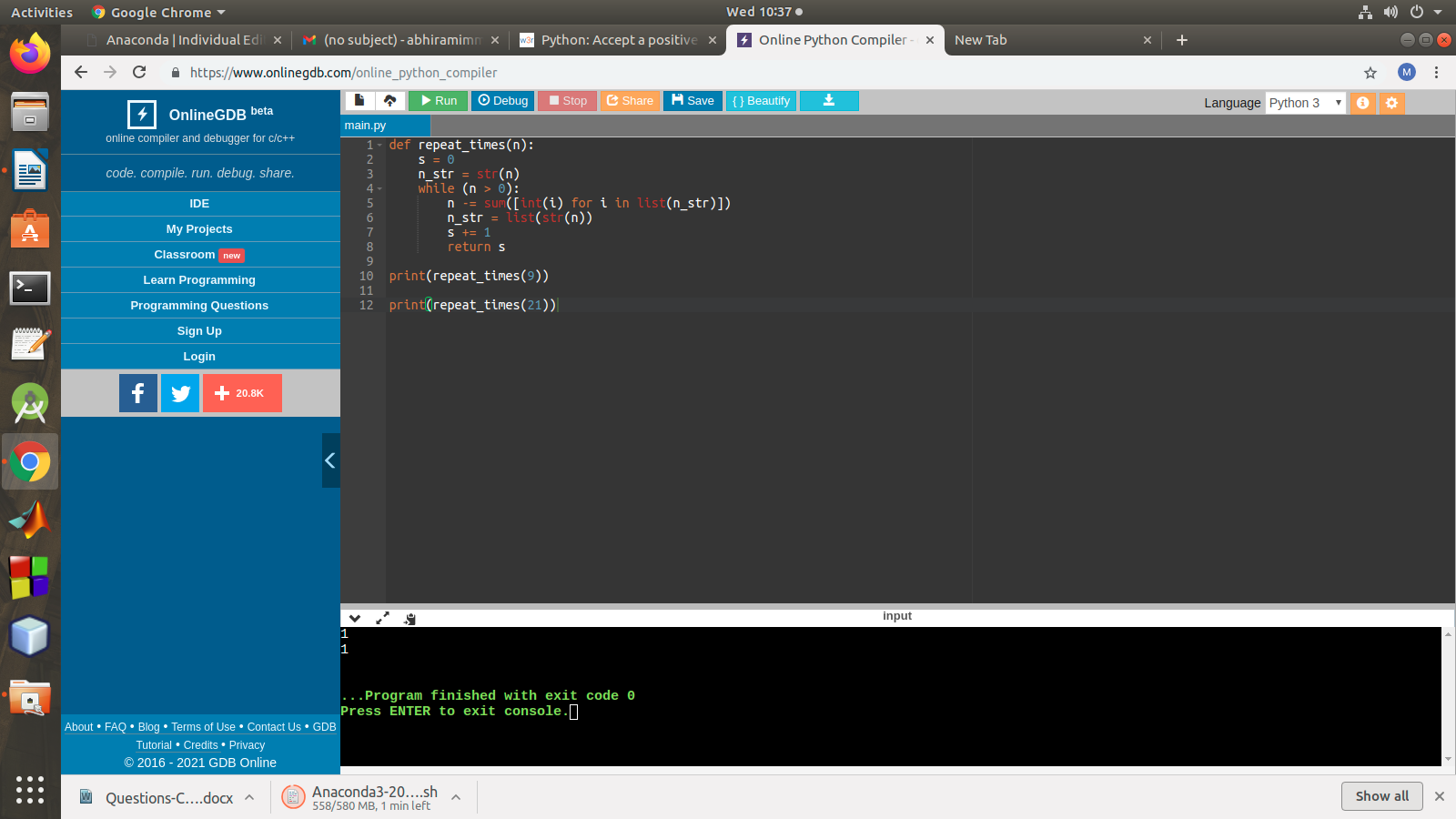
s += 1

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**

mobile = input('Please enter a mobile number: ' )

all = '0123456789'

print('missing digits are ', set(all) – set(mobile))

**Output**

