**DATA SCIENCE LAB**

**Submitted by,**

**Alex Sebastian**

**S3 MCA**

**Roll No:09**

**LAB CYCLE 1**

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

**PROGRAM**

lower = int(input("Enter the lower interval: "))  
upper = int(input("Enter the upper interval: "))  
for num in range(lower,upper+1):  
 if num>1:  
 for m in range(2,num):  
 if(num % m)==0:  
 print(num)  
 break

**OUTPUT**

Enter the lower interval: 3

Enter the upper interval: 10

4

6

8

9

10

1. Program to print the first N Fibonacci numbers.

**PROGRAM**

n = int(input("How many terms? "))  
fst=0  
sec=1  
count=0  
if n==1:  
 print(fst)  
else:  
 print("Fibonacci sequence:")  
 while count < n:  
 print(fst)  
 nth = fst + sec  
 fst = sec  
 sec = nth  
 count += 1

**OUTPUT**

How many terms? 5

Fibonacci sequence:

0

1

1

2

3

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

**PROGRAM**

print("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**

lengths of the triangle sides:

x: 6

y: 8

z: 9

Scalene triangle

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**

Enter the First number:4

Enter the Second number:9

The given numbers are co-prime

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

**PROGRAM**

import math  
a=int(input("Enter the value of a:"))  
b=int(input("Enter the value of b:"))  
c=int(input("Enter the value of c:"))  
disc=b \* b - 4 \* a \* c  
sol1= (-b+ math.sqrt(abs(disc))/2\*a)  
sol2= (-b- math.sqrt(abs(disc))/2\*a)  
if (disc>0):  
 print("There are 2 real solutions:")  
 print("solution 1:",sol1)  
 print("solution 2:",sol2)  
elif disc==0:  
 print("There are 1 real solution:")  
 print("solution is:", sol1)  
elif disc<0:  
 print("The roots are imaginary!!")

**OUTPUT**

Enter the value of a:3

Enter the value of b:5

Enter the value of c:7

The roots are imaginary!!

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

**PROGRAM**

n = int(input("Enter any number:"))  
sum1 = 0  
for i in range(1, n):  
 if(n % i == 0):  
 sum1 = sum1 + i  
if (sum1 == n):  
 print("The number is a Perfect number")  
else:  
 print("The number is not a Perfect number")

**OUTPUT**

Enter any number:7

The number is not a Perfect number

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**

Enter lower range: 0

Enter upper range: 1000

0

1

153

370

371

407

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**

<class 'list'>

['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']

<class 'tuple'>

('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')

<class 'set'>

{'Fri', 'Sat', 'Tue', 'Thu', 'Wed', 'Mon', 'Sun'}

<class 'dict'>

{'d1': 'Sun', 'd2': 'Mon', 'd3': 'Tue', 'd4': 'Wed', 'd5': 'Thu', 'd6': 'Fri', 'd7': 'Sat'}

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

**PROGRAM**

list1 = [9, 3, 4, 8, 8]  
list2 = [4, 5, 7, 2, 10]  
print("list 1 : " + str(list1))  
print("list 2 : " + str(list2))  
list3 = []  
for i in range(0, len(list1)):  
 list3.append(list1[i] + list2[i])  
print("Resultant list: " + str(list3))

**OUTPUT**

list 1 : [9, 3, 4, 8, 8]

list 2 : [4, 5, 7, 2, 10]

Resultant list: [13, 8, 11, 10, 18]

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

**PROGRAM**

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

**OUTPUT**

Enter the Number of rows : 2

Enter the Number of Columns: 3

Enter the elements of First Matrix:

3

4

5

6

7

8

First Matrix is:

[3, 4, 5]

[6, 7, 8]

Enter the elements of Second Matrix:

2

3

4

5

6

7

[2, 3, 4]

[5, 6, 7]

resultant matrix is

[5, 7, 9]

[11, 13, 15]

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

**PROGRAM**

def bubble\_sort(list1):  
 for i in range(0, len(list1) - 1):  
 for j in range(len(list1) - 1):  
 if (list1[j] > list1[j + 1]):  
 temp = list1[j]  
 list1[j] = list1[j + 1]  
 list1[j + 1] = temp  
 return list1  
list1 = [4, 9, 8, 6, 1, 2]  
print("The unsorted list is: ", list1)  
print("The sorted list is: ", bubble\_sort(list1))

**OUTPUT**

The unsorted list is: [4, 9, 8, 6, 1, 2]

The sorted list is: [1, 2, 4, 6, 8, 9]

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:",vowels)

**OUTPUT**

Enter string:malayalam

Number of vowels: 4

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(6))  
print(repeat\_times(18))

**OUTPUT**

1

2

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([7,3,5,6,1,6,0,1,0,0]))

**OUTPUT**

[2, 4, 8, 9]