***DATA SCEINCE and MACHINE LEARNING:***

**LAB CYCLE 1**

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

**program**

start = 11

end = 25

for i in range(start, end + 1):

if i > 1:

for j in range(2, i):

if (i % j == 0):

break

else:

print(i)

OUTPUT

2. Program to print the first N Fibonacci numbers.

**Program**

def printFibonacciNumbers(n):

f1 = 0

f2 = 1

if (n < 1):

return

print(f1, end=" ")

for x in range(1, n):

print(f2, end=" ")

next = f1 + f2

f1 = f2

f2 = next

printFibonacciNumbers(7)

OUTPUT

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

isosceles, equilateral or scalene.

**Program**

def checkTriangle(x, y, z):

if x == y == z:

print("Equilateral Triangle")

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

print("Isosceles Triangle")

else:

print("Scalene Triangle")

x = input("enter the first side")

y = input("enter the second variable")

z = input("enter the third variable")

checkTriangle(x, y, z)

OUTPUT

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

**program**

def \_\_gcd(a, b):

if (a == 0 or b == 0): return 0

if (a == b): return a

if (a > b):

return \_\_gcd(a - b, b)

return \_\_gcd(a, b - a)

def coprime(a, b):

if (\_\_gcd(a, b) == 1):

print("Co-Prime")

else:

print("Not Co-Prime")

a = 5;

b = 6

coprime(a, b)

a = 7;

b = 8

coprime(a, b)

OUTPUT

5. 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:

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).

**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

7. 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

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.

**Program.**

l = []

l.append(5)

l.append(10)

print("Adding 5 and 10 in list", l)

# Set

s = set()

s.add(5)

s.add(10)

print("Adding 5 and 10 in set", s)

# Tuple

t = tuple(l)

print("Tuple", t)

print()

# Dictionary

d = {}

d[5] = "Five"

d[10] = "Ten"

print("Dictionary", d)

OUTPUT

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

**Program.**

l = [5,6,7,8]

k=[1,2,3,4]

print("list 1 element ", l)

print("list 2 element", k)

res\_list = [l[i] + k[i] for i in range(len(l))]

print("Resultant list is : " + str(res\_list))

OUTPUT

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

**Program.**

matOne = []

print("Enter Elements for First Matrix: ")

for i in range(3):

matOne.append([])

for j in range(3):

num = int(input())

matOne[i].append(num)

matTwo = []

print("Enter Elements for Second Matrix: ")

for i in range(3):

matTwo.append([])

for j in range(3):

num = int(input())

matTwo[i].append(num)

matThree = []

for i in range(3):

matThree.append([])

for j in range(3):

matThree[i].append(matOne[i][j]+matTwo[i][j])

print("\nAddition Result of Two Given Matrix is:")

for i in range(3):

for j in range(3):

print(matThree[i][j], end=" ")

print()

OUTPUT

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

**Program**

a=[1,2,3,4,10,3,8]

rint("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

12. 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)

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

**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))

14. 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([9,8,3,2,2,0,9,7,6,3]))