**DATA SCIENCE & MACHINE LEARNING-LAB CYCLE 1**

**Anunadh Rajesh**

**20MCA015**

**S3MCA**

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

n1=int(input("enter the first interval:"))

n2=int(input("enter the second interval:"))

for x in range(n1,n2):

if x>1:

for m in range(2,x):

if(x % m)==0:

print(x)

break

**OUTPUT**

Enter the first interval: 1

Enter the second interval: 10

4

6

8

9

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

n\_terms = int(input("How many terms the user wants to print? "))

n\_1 = 0

n\_2 = 1

count = 0

if n\_terms <= 0:

print("Please enter a positive integer, the given number is not valid")

elif n\_terms == 1:

print("The Fibonacci sequence of the numbers up to", n\_terms, ": ")

print(n\_1)

else:

print("The fibonacci sequence of the numbers is:")

while count < n\_terms:

print(n\_1)

nth = n\_1 + n\_2

n\_1 = n\_2

n\_2 = nth

**OUTPUT**

How many terms the user wants to print? 6

The fibonacci sequence of the numbers is:

0

1

1

2

3

5

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

**isosceles, equilateral or scalene**.

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**

Input lengths of the triangle sides:

x: 4

y: 4

z: 5

isosceles triangle

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

def are\_coprime(a, b):

hcf = 1

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

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

hcf = i

return hcf == 1

first = int(input('Enter first number: '))

second = int(input('Enter second number: '))

if are\_coprime(first, second):

print('%d and %d are CO-PRIME' % (first, second))

else:

print('%d and %d are NOT CO-PRIME' % (first, second))

**OUTPUT**

Enter first number: 13

Enter second number: 28

13 and 28 are CO-PRIME

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

import cmath

a = float(input('Enter a: '))

b = float(input('Enter b: '))

c = float(input('Enter c: '))

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

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

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

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

**OUTPUT**

Enter a: 2

Enter b: 3

Enter c: 4

The solution are (-0.75-1.1989578808281798j) and (-0.75+1.1989578808281798j)

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

**=number)**

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: 3

The number is not a Perfect number!

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

print("Armstrong Numbers Between 1 and 1000:")

first = 1

last = 1000

while first<=last:

res = 0

temp = first

noOfDigit = 0

while temp>0:

temp = int(temp/10)

noOfDigit = noOfDigit + 1

num = first

while num>0:

rem = num%10

pow = 1

i = 0

while i<noOfDigit:

pow = pow\*rem

i = i+1

res = res+pow

num = int(num/10)

if res == first:

print(res)

first = first+1

**OUTPUT**

Armstrong Numbers Between 1 and 1000:

1

2

3

4

5

6

7

8

9

153

370

371

407

**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.**

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'>

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

<class 'dict'>

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

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

List1 = [7, 5, 21, 18, 8]

List2 = [9, 15, 6, 3,11]

print ("list1 : " + str(List1))

print ("list2 : " + str(List2))

newList = []

for n in range(0, len(List1)):

newList.append(List1[n] + List2[n])

print(newList)

**OUTPUT**

list1 : [7, 5, 21, 18, 8]

list2 : [9, 15, 6, 3, 11]

[16, 20, 27, 21, 19]

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

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]

for r in result:

print(r)

**OUTPUT**

Enter the Number of rows : 2

Enter the Number of Columns: 2

Enter the elements of First Matrix:

2

3

4

5

First Matrix is:

[2, 3]

[4, 5]

Enter the elements of Second Matrix:

2

3

4

5

[2, 3]

[4, 5]

[4, 6]

[8, 10]

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

a = []

number = int(input("Please Enter the Total Number of Elements : "))

for i in range(number):

value = int(input("Please enter the %d th Element of List1 : " %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 Sorted List in Ascending Order : ", a)

**OUTPUT**

Please Enter the Total Number of Elements : 5

Please enter the 0 th Element of List1 : 2

Please enter the 1 th Element of List1 : 3

Please enter the 2 th Element of List1 : 58

Please enter the 3 th Element of List1 : 2

Please enter the 4 th Element of List1 : 3

The Sorted List in Ascending Order : [2, 2, 3, 3, 58]

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

print("Enter the String:")

text = input()

vowela = ['a', 'A']

vowele = ['e', 'E']

voweli = ['i', 'I']

vowelo = ['o', 'O']

vowelu = ['u', 'U']

ca = 0

ce = 0

ci = 0

co = 0

cu = 0

for x in text:

if x in vowela:

ca = ca+1

elif x in vowele:

ce = ce+1

elif x in voweli:

ci = ci+1

elif x in vowelo:

co = co+1

elif x in vowelu:

cu = cu+1

print("\n'a' occurs ", ca)

print("'e' occurs ", ce)

print("'i' occurs ", ci)

print("'o' occurs ", co)

print("'u' occurs ", cu)

**OUTPUT**

Enter the String:

ANUNADH RAJESH

'a' occurs 3

'e' occurs 1

'i' occurs 0

'o' occurs 0

'u' occurs 1

**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**

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

3

**14. Write a Python program that accepts a 10 digit mobile number, and find the digits**

**which are absent in a given mobile number**

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([8,8,8,2,2,0,8,7,6,8]))

**OUTPUT**

[1, 3, 4, 5, 9]