**General category programs**

1. Write a program to print all the Non-Prime numbers between A and B? Sample Input: A = 12 B = 19

Sample Output:

14, 15, 16, 18

a = int(input())

b = int(input())

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

if x > 1:

for i in range (2, x):

if (x%i)== 0:

break

else:

print (x)

1. Find the year of the given Anniversary is leap year or not. If leap year then print the next Anniversary, if not leap year then print the previous Anniversary.

Sample Input:

Enter Date: 04/11/1947 Sample Output:

Given Anniversary Year: Non Leap Year. Anniversary Date: 04/11/1946

date = input("Enter the date to be checked: ")

c=date.split("/")

b = list(map(int,c))

input\_year=(b[2])

if(input\_year%4 == 0):

if(input\_year%100 == 0):

if(input\_year%400 == 0):

print("%d is Leap Year" %input\_year)

else:

print("%d is not the Leap Year" %input\_year)

else:

print("%d is the Leap Year" %input\_year)

else:

print("%d is not the Leap Year" %input\_year)

x=input\_year%4

if x!=0:

print("Previous Leap year:", input\_year-x)

else:

print("Next leap year:", input\_year+4)

1. Write a program to print the given number is Perfect number or not?

Sample Input: Given Number: 6

Sample Output: Its a Perfect Number

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

Sum = 0

for i in range(1, Number):

if(Number % i == 0):

Sum = Sum + i

if (Sum == Number):

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

else:

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

1. Write a program to generate Pythagorean Triplets for the given limit.

Enter upper limit: 10

3 4 5

8 6 10

A=input("Enter upper limit:")

c=0

m=2

if A.isnumeric():

x=int(A)

while(c<x):

for n in range(1,m+1):

a=m\*m-n\*n

b=2\*m\*n

c=m\*m+n\*n

if(c>x):

break

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

break

print(a,b,c)

m=m+1

else:

print("invalid input")

1. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input: Enter N value : 3 Enter 3 digit number: 143

Sample Output: Sum of 3 digit number: 8

num=int(input("Enter the number:"))

Sum=0

temp=num

while temp>0:

digit=temp%10

Sum+=digit

temp=temp//10

print("Sum of Digits:", Sum)

1. Program to find whether the given number is Armstrong number or not

Sample Input: Enter number: 153

Sample Output: Given number is Armstrong number

num=int(input("Enter the number:"))

Sum=0

temp=num

while temp>0:

digit=temp%10

Sum+=digit\*\*3

temp=temp//10

if Sum==num:

print("Armstrong Number")

else:

print("Not a Armstrong Number")

1. Program to find whether the given number is Harshad number or not

Sample Input: Enter number: 21

Sample Output: Given number is Harshad number

num=int(input("Enter the number:"))

Sum=0

temp=num

while temp>0:

digit=temp%10

Sum+=digit

temp=temp//10

if num%Sum==0:

print("Harshad Number")

else:

print("Not a Harshad Number")

1. Program to find whether the given number is Happy number or not

Sample Input: Enter number: 19

Sample Output: Given number is happy number

def happy(n):

temp=n

sum=0

while temp>0:

digit=temp%10

sum=digit\*\*2+sum

temp=temp//10

return sum

# Main Program

num=int(input("Enter the number:"))

result=num

while result!=1 and result!=4:

result=(happy(result))

if result==1:

print("True")

elif result==4:

print("False")

1. Program to find whether the given number is Tech number or not

Sample Input: Enter number: 3025

Sample Output: Given number is Tech number

n = 3025

m = str(n)

a = m[:len(m)//2]

b = m[len(m)//2:]

c = int(a)+int(b)

d = c\*\*2

if(d==n):

print("Tech number")

else:

print("Not a Tech number")

1. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. She is being offered 15 percent rate of interest; he is being offered 12 percent rate of interest for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000 Enter the no of years: 3

Gender (m/f): m

Is customer senior citizen (y/n): n Sample Output:

Interest: 60000

p=int(input("Enter the Principle amount:"))

n=int(input("Enter the number of years:"))

SC=input("Senior Citizen Yes/No:")

G=input("Male/Female:")

if SC=='Y' and G=='M':

print("SI=",(p\*n\*12)/100)

elif SC=='Y' and G=='F':

print("SI=",(p\*n\*15)/100)

else:

print("SI=",(p\*n\*10)/100)

1. Find the number of factors for the given number and print the 1st N factors of the given number.

Sample Input: Given number: 100

N: 4

Sample Output: Number of factors = 9

1st 4 factors are: 1, 2, 4, 5

x=int(input("Enter the number:"))

y=[]

print("The factors of",x,"are:")

for i in range(1, x):

if x % i == 0:

y.append(i)

print(y)

print("Number of factors:", len(y))

n=int(input("Enter N value:"))

if n>len(y):

print("Invalid")

else:

print("First", n, "factors:")

for k in range(0,n):

print(y[k], end=' ')

1. Write a program to print number of factors and to print nth factor of the given number.

Sample Input: Given Number: 100

N = 4

Sample Output:

Number of factors = 9 4th factor of 100 = 5

x=int(input("Enter the number:"))

y=[]

print("The factors of",x,"are:")

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

if x % i == 0:

y.append(i)

print(y)

print("Number of factors:", len(y))

n=int(input("Enter N value:"))

print(n, "th factor is:",y[n-1])

1. Write a program to print unique permutations of a given number Sample Input:

Given Number: 143 Sample Output:

Permutations are:

134

143

314

341

413

431

import itertools

n=input("Enter the number")

P=list(itertools.permutations(n))

print(\*[''.join(p) for p in P])

1. Write a program to find the square, cube of the given decimal number Sample Input:

Given Number: 0.6

Sample Output: Square Number: 0.36 Cube Number:0.216

import math

num=float(input("Enter the number:"))

print("Square number=",math.pow(num,2))

print("Cube number=",round(math.pow(num,3),3))

1. Write a program to convert the Binary to Decimal, Octal Sample Input:

Given Number: 1101 Sample Output:

Decimal Number: 13 Octal: 15

num=input("Enter the binary number:")

bin\_num="01"

for i in range(len(num)):

binary=True

if num[i] not in bin\_num:

print("Invalid input")

binary=False

break

if binary:

dec\_number=int(num,2)

oct\_number=oct(dec\_number)

hexa=hex(dec\_number)

print("Decimal Equivalent=",dec\_number)

print("Octal Equivalent=",oct\_number)

print("Hexa Equivalent=",hexa)

1. Add Binary

Given two binary strings a and b, return their sum as a binary string.

a and b consist only of '0' or '1' characters.

Each string does not contain leading zeros except for the zero itself.

**Test cases:**

1.Input: a = "11", b = "1"

1. Output: "100"

num1=input("Enter the binary number 1=")

num2=input("Enter the binary number 2=")

sum=bin(int(num1,2)+int(num2,2))

print("Sum of binary numbers: ",sum[2:])

17.Python program to find the greatest of three binary numbers

a='1101'

b='1110'

c='1111'

bina=int(a,2)

binb=int(b,2)

binc=int(c,2)

if bina>binb and bina>binc:

print("Greatest is", a)

elif binb>bina and binb>binc:

print("Greatest is", b)

else:

print("Greatest is", c)

18.Write a program for matrix multiplication?

Sample Input:

Mat1 =

Mat2 =

Sample Output:

Mat Sum =

X=[[1,2],

[5,3]]

Y=[[2,3],

[4,1]]

result=[[0,0],

[0,0]]

# iterate through rows of X

for i in range(len(X)):

# iterate through columns of Y

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

# iterate through rows of Y

for k in range(len(Y)):

result[i][j] += X[i][k] \* Y[k][j]

for r in result:

print(r)

1. Write a program for matrix addition?

Sample Input:

Mat1 =

Mat2 =

Sample Output:

Mat Sum =

a=[[1,2],

[5,3]]

b=[[2,3],

[4,1]]

c=[[0,0],

[0,0]]

for i in range(len(a)):

for j in range(len(b)):

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

for i in c:

print(i)

1. Find the LCM and GCD of n numbers

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output: LCM = 80 GCD = 4

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

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

x = n1

y = n2

while(n2!=0):

t = n2

n2 = n1 % n2

n1 = t

gcd = n1

print("GCD of {0} and {1} = {2}".format(x,y,gcd))

lcm = (x\*y)/gcd

print("LCM of {0} and {1} = {2}".format(x,y,lcm))

1. Transpose of a matrix

matrix = [[4, 6, 7, 8],

           [3, 7, 2, 7],

           [7, 3, 7, 5]]

a=[[1,2],

[3,2]]

c=[[0,0],

[0,0]]

for i in range(len(a)):

for j in range(len(a)):

c[i][j]=a[j][i]

for i in c:

print(i)

1. Program to find row, column and diagonal sum in Matrix

a = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

o/p:

Sum of 1 row: 6

Sum of 2 row: 15

Sum of 3 row: 24

Sum of 1 column: 12

Sum of 2 column: 15

Sum of 3 column: 18

Diagonal sum 15

#Initialize matrix a

a = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

#Calculates number of rows and columns present in given matrix

rows = len(a);

cols = len(a[0]);

#Calculates sum of each row of given matrix

for i in range(0, rows):

sumRow = 0;

for j in range(0, cols):

sumRow = sumRow + a[i][j];

print("Sum of " + str(i+1) +" row: " + str(sumRow));

#Calculates sum of each column of given matrix

for i in range(0, rows):

sumCol = 0;

for j in range(0, cols):

sumCol = sumCol + a[j][i];

print("Sum of " + str(i+1) +" column: " + str(sumCol));

#Calculates sum of diagonal

diagonal=0

for k in range(0,rows):

diagonal=diagonal+a[k][k]

print("Diagonal sum",diagonal)

1. Given three integers **M, N**and**K**. Consider a grid of **M \* N**, where **mat[i][j] = i \* j** (1 based index). The task is to return the **Kth** smallest element in the **M \* N** multiplication table.

def findKthNumber(m, n, k):

low = 1

high = n\*m

while low < high:

mid = (low + high) // 2

count = 0

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

count += min(n, mid//i)

if count < k:

low = mid + 1

else:

high = mid

return low

#Driver Program

m=3

n=3

k=5

print(findKthNumber(m,n,k))

1. Print the sum of boundary elements of a matrix

def printBoundary(a, m, n):

for i in range(m):

for j in range(n):

if (i == 0):

print a[i][j],

elif (i == m-1):

print a[i][j],

elif (j == 0):

print a[i][j],

elif (j == n-1):

print a[i][j],

else:

print " ",

print

# Driver code

if \_name\_ == "\_main\_":

a = [[1, 2, 3, 4], [5, 6, 7, 8],

[1, 2, 3, 4], [5, 6, 7, 8]]

1. Print the given matrix in spiral order

a=[[2,5,3],

[6,4,1],

[9,7,8]]

l=[]

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

l.append(a[0][i])

for j in range(1,len(a)-1):

l.append(a[j][-1])

for k in range(1,len(a[-1])+1):

l.append(a[-1][-k])

for m in range(len(a[0])-1):

l.append(a[1][m])

print(l)

1. Write a python program to find the sum of N numbers

Sample input: N=10

Sample output: Sum=55

N=int(input("Enter the limit:"))

count=0

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

count+=i

print("Sum of N natural numbers",count)

25.Write a python program to find the sum of 12+22+.......N2 numbers

Sample input: N=6

Sample output: Sum=91

N=int(input("Enter the limit:"))

count=0

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

count+=i\*i

print("Sum of square of N natural numbers",count)

26.Find the factorial of the number.

Sample input: N=5

Sample output: Sum=120

def fact(n):

if n==0 or n==1:

return 1

if n>1:

return n\*fact(n-1)

# Main program

num=int(input("Enter the number: "))

print(fact(num))

27.Write a python program to find the sum of 1!+2!+.......N! numbers

Sample input: N=4

Sample output: Sum=33

def fact(n):

if n==0 or n==1:

return 1

if n>1:

return n\*fact(n-1)

# Main program

num=int(input("Enter the number: "))

sum=0

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

sum+=fact(i)

print(sum)

28.Write a python program to find the sum of 1!/1+2!/2+.......N!/N numbers

Sample input: N=5

Sample output: Sum=34

def fact(n):

if n==0 or n==1:

return 1

if n>1:

return n\*fact(n-1)

# Main program

num=int(input("Enter the number: "))

sum=0

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

sum+=fact(i)/i

print(sum)

29.Write a python program to find the difference between sum of square and square of sum N numbers

Sample input: N=5

Sample output: Diff=170

n=20

x=(n\*(n+1)\*(2\*n+1))/6

y=((n\*(n+1))/2)\*\*2

print("Difference:",y-x)

30.Write a python program to find the sum of all digits in a triangle

def digits\_sum():

for i in reversed(range(len(triangle\_num) - 1)):

for j in range(len(triangle\_num[i])):

triangle\_num[i][j] += max(triangle\_num[i + 1][j], triangle\_num[i + 1][j + 1])

return str(triangle\_num[0][0])

#Main Program

triangle\_num =

[[3],

[4,6],

[2,7,6],

[8,5,9,3]]

print(digits\_sum())

31. Fibonacci series

def Fibonacci(n):

if n < 0:

print("Incorrect input")

elif n == 0:

return 0

elif n == 1 or n == 2:

return 1

else:

return Fibonacci(n-1) + Fibonacci(n-2)

# Driver Program

num=int(input("Enter the number of terms="))

for i in range(num):

print(Fibonacci(i))

32.You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Sol:

def fib(n):

if n <= 1:

return n

return fib(n-1) + fib(n-2)

s = int(input("Enter the value of n: "))

print ("Number of ways = ", end="")

print (fib(s+1))

Output:

Enter the value of n: 5

Number of ways = 8

33. Vehicles and children program

M=int(input("Enter the number of vehicles:"))

N=int(input("Enter number of children: "))

x=M%N

if x==0:

print("You are so lucky")

elif x!=0 and x%2!=0:

print("Mr.Peter gets", x, "Vehicles")

elif x!=0 and x%2==0:

print("Mr.Peter gets", x, "Vehicles. He is lucky")

34. Find the difference between two dates.

from datetime import datetime

from dateutil import relativedelta

# get two dates

d1 = '17/7/1980'

d2 = '16/3/2007'

# convert string to date object

start\_date = datetime.strptime(d1, "%d/%m/%Y")

end\_date = datetime.strptime(d2, "%d/%m/%Y")

# Get the relativedelta between two dates

delta = relativedelta.relativedelta(end\_date, start\_date)

print('Years, Months, Days between two dates is')

print(delta.years, 'Years,', delta.months, 'months,', delta.days, 'days')

delta.years

d3=d1.split('/')

d4=d2.split('/')

BY=int(d3[2])

JY=int(d4[2])

if(delta.years>=19 and BY%4==0):

print("I m a lucky adult")

elif delta.years<19:

print("I m aspiring to become adult")

if BY%4==0:

print("Birth year is leap Year")

else:

print("Birth year is not a leap Year")

if JY%4==0:

print("Joining year is leap Year")

else:

print("Joining year is not a leap Year")

35. Calendar Programs

# Current time

from datetime import datetime

now=datetime.now()

print(now)

# Current date

from datetime import datetime

now=datetime.today()

print(now)

# Entire month in a year

import calendar

y = int(input("Enter the Year :"))

print(calendar.prcal(y))

# Particular month in a year

import calendar

y = int(input("Enter the Year :"))

m=int(input("Enter the month :"))

print(calendar.month(y,m))

#Program to find number of weekdays in(mm/yyyy)

import numpy as np

# Number of weekdays in March 2017

print("Number of weekdays in March 2017:",

np.busday\_count('2017-03', '2017-04'))

# Number of sundays in Nov 2020

print("Number of Sunday in november 2020:",

np.busday\_count('2020-11', '2020-12',weekmask='Sun'))

# input year and month

yearMonth = '2017-05'

# getting date of first monday

date = np.busday\_offset(yearMonth, 0, roll='forward',weekmask='Mon')

# display date

print(date)