|  |  |
| --- | --- |
| **Ex. 4** | **EXPLORING LOOPS** |
| **Date: 12 Feb 2024** | |
|  | |

**Aim:**

To explore the use of loops in Python by writing programs for the following and executing them:

1. Print the list of prime numbers between 1 and N.
2. Print the multiplication table up to M for a number N.
3. Print the following pattern for 2N-1 rows.   
                                 1  
                           2          2  
                     3          3           3  
                          2           2  
                                 1
4. Find the greatest common divisor of 2 numbers obtained from the user.
5. Find the sum of the series up to N terms.
6. Find the sum of the digits of a given integer, N.
7. Find the square root of an integer, N, using Newton's method. Obtain N and the limit, L, from the user.

**Algorithm:**

**( a )**

**Step – 1 :** Get the value of N form the user

**Step – 2 :** Iterate through each number between 1 and N and Check if it’s prime by iterrative modulas division by all the numbers between 2 and the number / 2 until a divisor is found then it is not prime.

**Step – 3 :** Print the Numbers that are not

**( b )**

**Step – 1:** Get the Inputs of N and M

**Step – 2 :** In a for loop of range 1 to N+1 print NxM

**Step – 3 :** Print the multiplicatives in a standard Tables format ( M x N = {M\*N})

**( c )**

**Step – 1 :** Get the input of N (the maximum number in pattern) from the user.

**Step – 2 :** With 2 for loops. With one for top half and the other for lower half, print the required spaces and print the row number,row number of times.

**Step – 3 :** The lower half starts form 1 – the end and lowers till it reaches 1 as row number.

**( d )**

**Step – 1 :** Get two numbers from the user

**Step – 2 :** Divide the larger number by the numbers between the smaller number and 1.

**Step – 3 :** Untill you reach a divisible number that is divisible by both the larger and smaller number print the Greatect common divisor.

**( e )**

**Step – 1 :** Obtain the number of terms from the user.

**Step – 2 :** Create a list of fibonacci series for the number of terms

**Step – 3 :** Add the elements by multiplying the element to the power of th term number and to the -1 if it is a even term.

**( f )**

**Step – 1 :** Get the Number form the user

**Step – 2:** Conver the number into a string and iterate throught it while converting each character into a integer and adding it to a common variable ‘sum’.

**Step – 3 :** Print the sum to the user.

**( g )**

**Step – 1 :** Get the number as input from the user.

**Step – 2 :** Assign X to be N,In a while loop assign root to be = 0.5 \* (X + (N/X)) and exit the while loop if the difference between X and root is less than the Tolerance level if not then assign X the value of root.

**Step – 3 :** Print the the value of root to the user.

**Program:**

**( a )**

# Printing list of prime numbers Between 1 and N

for i in range(2,int(input("Enter N : "))+1):

f=0

for j in range(2,(i//2)+1):

if(i%j == 0):

f = 1

break

if(f==0):

print(i,end = ", ")

**( b )**

# Multiplication Table

N = int(input("Enter the Starting Integer : "))

for j in range(1,int(input("Enter the number of Multiplicatives Needed : "))+1):

print(N," x ", j , " = ", N\*j)

**( c )**

# Print Diamond Pattern

N = int(input("Enter the maximum number in the pattern : "))

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

print(" "\*(N-i),(str(i)+' ')\*i)

for i in range(N-1,0,-1):

print(" "\*(N-i),(str(i)+' ')\*i)

**( d )**

# Greatest Common Divisor of two numbers

M = int(input("Enter Number - 1 : "))

n = int(input("Enter Number - 2 : "))

if(n>M):

n,M = M,n

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

if (M%i==0 and n%i==0):

print("Greatest Common Divisor is : ",i)

break

**( e )**

# Find the sum of series

sum = 0

N = int(input("Enter the number of terms in Series : "))

F\_Series = [1,1]

for i in range(2,N):

F\_Series.append(F\_Series[i-1] + F\_Series[i-2])

for i in range(N):

sum+= ((-1)\*\*i) \* (F\_Series[i]\*\*(i+1))

print("Sum of Series = ",sum)

**( f )**

# Sum Of digits of given Number

sum = 0

for i in [int(x) for x in input("Enter the number : ")]:

sum+=i

print(sum)

**( g )**

# Newton Method for finding Root

N = int(input("Enter the number to find root of : "))

l = float(input('Enter the tolerance level : '))

X = N

while(True):

root = 0.5 \* (X + (N/X))

if(root - X<0):

if(X-root< l):

break

else:

if(root-X<l):

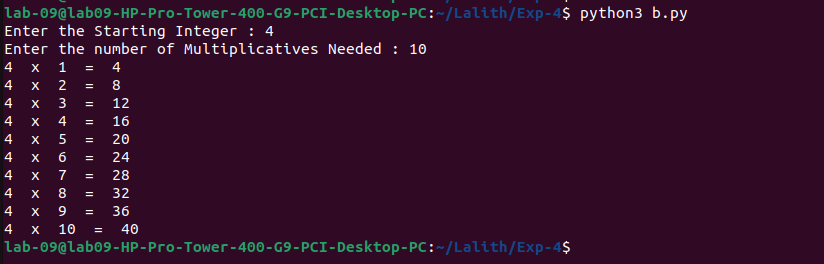
break

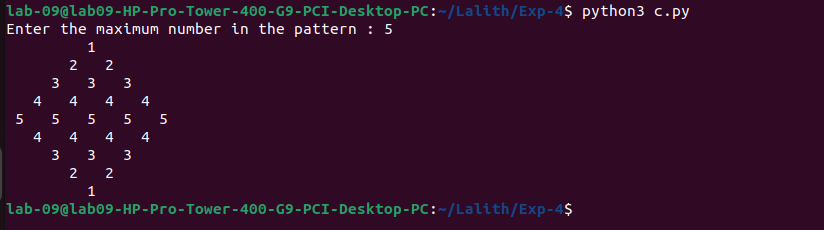
X = root

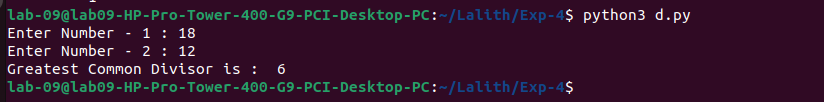
print(root)

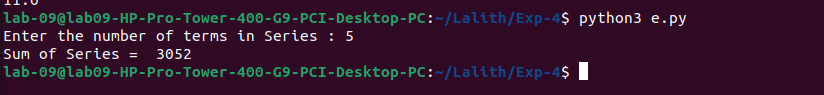
**Screenshot of Output:**

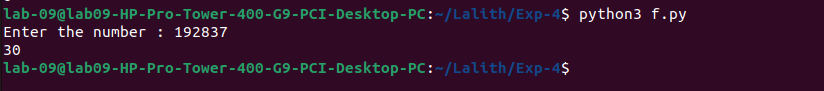
**( a )**

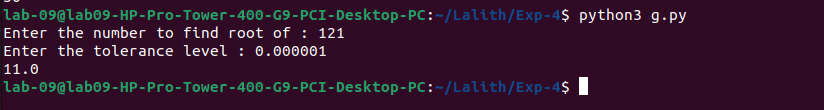
**( b )**

**( c )**

**( d )**

**( e )**

**( f )**

**( g )**

**Result:**

Thus, programs have been written and executed to explore the use of loops in Python.