**LAB CYCLE – 3**

**PROGRAM NO : 01**

**Aim:**

**Write a program to find the factorial of a number.**

**Source Code:**

**n=int(input("Enter a number : "))**

**if n<0:**

**print("Factorial does not exist for negative numbers")**

**else:**

**f=1**

**for i in range(1,n+1):**

**f=f\*i**

**print(f"Factorial : {f}")**

**Output:**

****

****

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 02**

**Aim:**

**Generate Fibonacci series of N terms.**

**Source Code:**

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

**a,b=0,1**

**print(a,b,end=" ")**

**for i in range(2,n):**

**c=a+b**

**print(c,end=" ")**

**a,b=b,c**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 03**

**Aim:**

**Write a program to find the sum of all items in a list. [Using for loop]**

**Source Code:**

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

**l=[]**

**for i in range(n):**

**a=int(input("Enter the element:"))**

**l.append(a)**

**print("List :",l)**

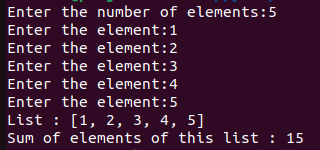
**s=0**

**for i in l:**

**s=s+i**

**print("Sum of elements of this list :",s)**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 04**

**Aim:**

**Generate a list of four digit numbers in a given range with all their digits even and**

**the number is a perfect square.**

**Source Code:**

**from math import \***

**squares=[]**

**ll=int(input("Enter the lower limit:"))**

**ul=int(input("Enter the upper limit:"))**

**for i in range(ll,ul):**

**for j in range(1,i):**

**if sqrt(i)==j:**

**squares.append(i)**

**print("Perfect squares between this limit:",squares)**

**even=[]**

**for i in squares:**

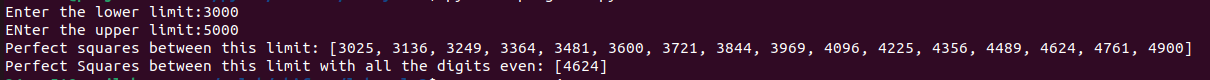
**square\_str=str(i)**

**if int(square\_str[0])%2==0 and int(square\_str[1])%2==0 and int(square\_str[2])%2==0 and int(square\_str[3])%2==0:**

**even.append(int(square\_str))**

**print("Perfect Squares between this limit with all the digits even:",even)**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO :05**

**Aim:**

**Write a program using a for loop to print the multiplication table of n, where n is**

**entered by the user.**

**Source Code:**

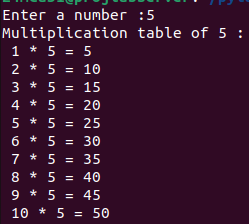
**n=int(input("Enter a number :"))**

**print(f"Multiplication table of {n} :")**

**for i in range(1,11):**

**print(f" {i} \* {n} = {i\*n}")**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 06**

**Aim:**

**Write a program to display alternate prime numbers till N (obtain N from the user).**

**Source Code:**

**from math import \***

**n=int(input("Enter the limit:"))**

**prime=[]**

**for i in range(2,n+1):**

**is\_prime=True**

**for j in range(2,int(sqrt(i))+1):**

**if i%j==0:**

**is\_prime=False**

**if is\_prime:**

**prime.append(i)**

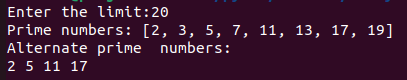
**print("Prime numbers:",prime)**

**print("Alternate prime numbers:")**

**for i in range(0,len(prime),2):**

**print(prime[i],end=" ")**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 07**

**Aim:**

**Write a program to compute and display the sum of all integers that are divisible**

**by 6 but not by 4, and that lie below a user-given upper limit.**

**Source Code:**

**n=int(input("Enter a limit:"))**

**l=[]**

**for i in range(1,n+1):**

**if i%6==0 and i%4!=0:**

**l.append(i)**

**print("List of integers divisible by 6 and not divisible by 4 upto this limit :",l)**

**s=0**

**for i in l:**

**s=s+i;**

**print("Sum of these integers :",s)**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 08**

**Aim:**

**Calculate the sum of the digits of each number within a specified range (from 1 to**

**a user-defined upper limit). Print the sum only if it is prime.**

**Source Code:**

**from math import \***

**n=int(input("Enter the upper limit:"))**

**l=[]**

**for i in range(1, n + 1):**

**a = str(i)**

**s = 0**

**for j in a:**

**b = int(j)**

**s += b**

**l.append(s)**

**print("\nSum of digits of numbers within this limit:",l)**

**prime=[]**

**for i in l:**

**is\_prime=True**

**for j in range(2,int(sqrt(i))+1):**

**if i%j==0:**

**is\_prime=False**

**break**

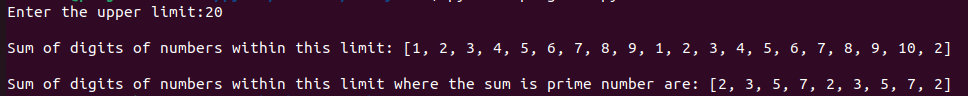
**if is\_prime:**

**if i!=1:**

**prime.append(i)**

**print("\nSum of digits of numbers within this limit where the sum is prime number are:",prime)**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 09**

**Aim:**

**A number is input through the keyboard. Write a program to determine if it’s**

**palindromic.**

**Source Code:**

**n=int(input("Enter a number:"))**

**s=str(n)**

**if s==s[::-1]:**

**print("This Number is Palindrome")**

**else:**

**print("This Number is not a Palindrome")**

**Output:**

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO :10**

**Aim:**

**Write a program to generate all factors of a number. [use while loop]**

**Source Code:**

**n=int(input("Enter a number:"))**

**l=[]**

**i=1**

**while(i<=n):**

**if n%i==0:**

**l.append(i)**

**i=i+1**

**print(f"Factors of {n} are : {l}")**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 11**

**Aim:**

**Write a program to find whether the given number is an Armstrong number or**

**not. [use while loop]**

**Source Code:**

**num=int(input("Enter a number:"))**

**num\_str=str(num)**

**noofdigit=len(num\_str)**

**sum=0**

**i=0**

**while i<noofdigit:**

**n=int(num\_str[i])**

**p=n\*\*noofdigit**

**sum=sum+p**

**i=i+1**

**if num==sum:**

**print(f"{num} is an Armstrong number")**

**else:**

**print(f"{num} is not an Armstrong number")**

**Output:**

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO : 12**

**Aim:**

**Display the given pyramid with the step number accepted from the user. Eg: N=4**

**1**

**24**

**369**

**4 8 12 16**

**Source Code:**

**n=int(input("Enter the step number of the pyramid:"))**

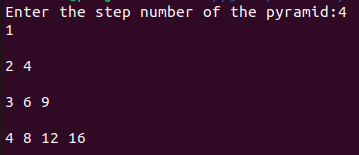
**for i in range(1,n+1):**

**for j in range(1,i+1):**

**print(j\*i,end=" ")**

**print("\n")**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROGRAM NO :13**

**Aim:**

**Construct following pattern using nested loop**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**Source Code:**

**for i in range(1,6):**

**for j in range(1,i+1):**

**print("\*",end=" ")**

**print("\n")**

**for l in range(1,5):**

**k=5**

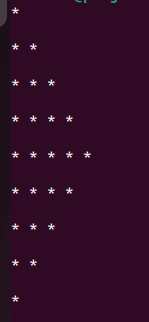
**while k>l:**

**print("\*",end=" ")**

**k=k-1**

**print("\n")**

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**