# implementation of linear search

def linearsearch(arr, x):

   for i in range(len(arr)):

      if arr[i] == x:

         return i

   return -1

arr = []                                                                    #for list

size=int(input("The number of element in list :"))

for k in range(size):

    a=int(input("the numbers are :"))

    arr.append(a)

print("\*\*THE GIVEN LIST IS\*\*\n",arr)

x =int(input("The number that you want to search in list :"))

print("element found at index,",(linearsearch(arr,x)))

# imlpimentation of sentinal search

def sentinel\_search (arr, x):

    size = len(arr)

    arr.append(x)

    i = 0

    while(arr[i] != x):

        i += 1

    if(i == size):

        return None

    else:

        return i

arr = []                                                                    #for list

size = int(input("Size of list is :"))

for k in range(size):

    a=int(input("the numbers are :"))

    arr.append(a)

print("\*\*THE GIVEN LIST IS\*\*\n",arr)

x =int(input("The number that you want to search in list :"))

if sentinel\_search(arr,x)==None:

    print("element ",x ," not found in list")

else:

    print("element found in list at index,",sentinel\_search (arr, x))

# implimentation of binary search

def binary\_search(arr, low, high, x):

    if high >= low:

        mid =(high+low) // 2

        if arr[mid] == x:

            return mid

        elif arr[mid] > x:

            return binary\_search(arr, low, mid - 1, x)

        else:

            return binary\_search(arr, mid + 1, high, x)

    else:

        return -1

arr = []                                                                    #for list

size = int(input("Size of list is :"))

for k in range(size):

    a=int(input("the numbers are :"))

    arr.append(a)

print("\*\*THE GIVEN LIST IS\*\*\n",arr)

x =int(input("The number that you want to search in list :"))

if binary\_search(arr, 0, len(arr), x) != -1:

    print("Element is present at index,", binary\_search(arr, 0, len(arr), x))

else:

    print("element ",x ," not found in list")

# implementatiion of Fibonacci search.

from bisect import bisect\_left

def fibMonaccian\_Search(arr, x, n):

    fibMMm2 = 0                                     # Initialize fibonacci numbers

    fibMMm1 = 1

    fibM = fibMMm2 + fibMMm1

    while (fibM < n):

        fibMMm2 = fibMMm1

        fibMMm1 = fibM

        fibM = fibMMm2 + fibMMm1

    offset = -1

    while (fibM > 1):

        i = min(offset+fibMMm2, n-1)

        if (arr[i] < x):

            fibM = fibMMm1

            fibMMm1 = fibMMm2

            fibMMm2 = fibM - fibMMm1

            offset = i

        elif (arr[i] > x):

            fibM = fibMMm2

            fibMMm1 = fibMMm1 - fibMMm2

            fibMMm2 = fibM - fibMMm1

        else:

            return i

    if(fibMMm1 and arr[n-1] == x):

        return n-1

    return -1

arr = []                                                                    #for list

n=int(input("The number of element in list :"))

for k in range(n):

    a=int(input("the numbers are :"))

    arr.append(a)

print("\*\*THE GIVEN LIST IS\*\*\n",arr)

x =int(input("The number that you want to search in list :"))

if fibMonaccian\_Search(arr, x, n)>=0:

    print("Found at index:",fibMonaccian\_Search(arr, x, n))

else:

    print("element ",x ," not found in list");