```
'''
//===============================================================================
// Name        : 21465_Pract4.py
// Author      : Chaitanya Paraskar
// Roll No.    : 21465
// Aim         : a) Write a Python program to store roll numbers of student in
array
                    who attended training program in random order.
                    Write function for searching whether particular student
attended
                    training program or not, using Linear search and Sentinel
search.
                 b) Write a Python program to store roll numbers of student
array
                    who attended training program in sorted order.
                    Write function for searching whether particular student
attended
                    training program or not, using Binary search and Fibonacci
search.
//===============================================================================

'''


class Sentinal:
    def __init__(self):
        n = int(input("Enter No. of elements in Array : "))
        self.arr = [None] * (n+1)
        self.n = n

        for i in range(n):
            print("i = ", i)
            e = int(input(f"Enter {i+1}th element: "))
            self.arr[i] = e

    def search(self, x):
        self.arr[self.n] = x
        i = 0

        while self.arr[i] != x:
            i = i+1

        if i != self.n:
            return i
        return -1

#
---------------------------------------------------------------------------------
-----------------------


class Sequential:
    def __init__(self):
        n = int(input("Enter No. of elements in Array : "))
        self.arr = [None] * n
        self.n = n
        for i in range(n):
            e = int(input(f"Enter {i+1}th element: "))
            self.arr[i] = e

    def search(self, x):
        i = 0
        for i in range(self.n):
            if self.arr[i] == x:
                return i
        return -1

#
--------------------------------------------------------------------------------
----------------------


class Binary:
    def __init__(self):
        n = int(input("Enter No. of elements in Array : "))
        self.arr = [None] * n
        self.n = n
        for i in range(n):
            e = int(input(f"Enter {i+1}th element: "))
            self.arr[i] = e
        self.sort()

    def sort(self):
        swapped = False

        for i in range(self.n-1):
            for j in range(0, self.n-i-1):
                if self.arr[j] > self.arr[j + 1]:
                    swapped = True
                    self.arr[j], self.arr[j + 1] = self.arr[j + 1], self.arr[j]

            if not swapped:
                return

    def search(self, x):
        min = 0
        max = self.n-1
        mid = (min + max)//2
        found = False

        while min <= max:
            mid = (min + max)//2

            if self.arr[mid] == x:
                found = True
                break
            elif self.arr[mid] > x:
                max = mid - 1
            elif self.arr[mid] < x:
                min = mid + 1

        if found:
            return mid

        return -1

#
--------------------------------------------------------------------------------
-----------------------


class Fibonacci:
    def __init__(self):
        n = int(input("Enter No. of elements in Array : "))
        self.arr = [None] * n
        self.n = n
        for i in range(n):
            e = int(input(f"Enter {i+1}th element: "))
            self.arr[i] = e
        print("arr = ", self.arr)
        self.sort()
        print("arr = ", self.arr)

    def sort(self):
        swapped = False

        for i in range(self.n-1):
            for j in range(0, self.n-i-1):
                if self.arr[j] > self.arr[j + 1]:
                    swapped = True
                    self.arr[j], self.arr[j + 1] = self.arr[j + 1], self.arr[j]

            if not swapped:
                return

    def fibo(self, m):
        a = 0
        b = 1
        c = -1

        if m < 0:
            return -1
        if m == 0:
            return a
        if m == 1:
            return b
        else:
            for i in range(2, m+1):
                temp = a + b
                a = b
                b = temp
            return b

    def search(self, x):

        m = 0
        # increment m till fibo(m) is less than n
        while self.fibo(m) < self.n:
            m = m+1

        # init offset to -1
        offset = -1

        # iterate till fibo(m) is greater than 1
        while self.fibo(m) > 1:

            # init mid to minimum of given 2 terms
            mid = self.min(offset+self.fibo(m - 2), self.n - 1)

            # if x is greater than arr[mid] set offset = mid and decrement m by
1
            if x > self.arr[mid]:
                offset = mid
                m = m - 1
            # if x is smaller than arr[mid] decrement m by 2
            elif x < self.arr[mid]:
                m = m - 2
            # if x is equal to arr[mid]
            # i.e. x is found
            elif x == self.arr[mid]:
                return mid

        if (not (self.fibo(m - 1) > 0)) and (self.arr[offset + 1] == x):
            return offset + 1
        return -1

    def min(self, a, b):
        if a > b:
            return a
        else:
            return b

#
--------------------------------------------------------------------------------
----------------------


def main():
    while True:
        print("1. Sequential Search")
        print("2. Sentinal Search")
        print("3. Binary Search")
        print("4. Fibonacci Search")
        print("5. Exit")
        ch = int(input("Enter choice"))

        if ch == 1:
            obj = Sequential()
            x = int(input("Enter Element to be Searched :"))
            e = obj.search(x)
            if e == -1:
                print("Element not Found !!")
            else:
                print(f"Element found at {e}th index")
        elif ch == 2:
            obj = Sentinal()
            x = int(input("Enter Element to be Searched :"))
            e = obj.search(x)
            if e == -1:
                print("Element not Found !!")
            else:
                print(f"Element found at {e}th index")
        elif ch == 3:
            obj = Binary()
            x = int(input("Enter Element to be Searched :"))
            e = obj.search(x)
            if e == -1:
                print("Element not Found !!")
            else:
                print(f"Element found at {e}th index")
        elif ch == 4:
            obj = Fibonacci()
            x = int(input("Enter Element to be Searched :"))
            e = obj.search(x)
            if e == -1:
                print("Element not Found !!")
            else:
                print(f"Element found at {e}th index")
        elif ch == 5:
            break
        else:
            print("Enter Valid Input")


main()

'''

OUTPUT :

$ python pract4.py
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
```

```
Enter choice1
Enter No. of elements in Array : 5
Enter 1th element: 1
Enter 2th element: 2
Enter 3th element: 3
Enter 4th element: 4
Enter 5th element: 5
Enter Element to be Searched :4
Element found at 3th index
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
Enter choice2
Enter No. of elements in Array : 5
Enter 1th element: 1
Enter 2th element: 2
Enter 3th element: 3
Enter 4th element: 4
Enter 5th element: 5
Enter Element to be Searched :2
Element found at 1th index
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
Enter choice3
Enter No. of elements in Array : 5
Enter 1th element: 1
Enter 2th element: 2
Enter 3th element: 3
Enter 4th element: 4
Enter 5th element: 5
Enter Element to be Searched :6
Element not Found !!
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
Enter choice4
Enter No. of elements in Array : 1
Enter 1th element: 4
arr =  [4]
Enter Element to be Searched :1
Element not Found !!
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
Enter choice4
Enter No. of elements in Array : 5
Enter 1th element: 1
Enter 2th element: 2
Enter 3th element: 3
Enter 4th element: 4
Enter 5th element: 5
arr =  [1, 2, 3, 4, 5]
arr =  [1, 2, 3, 4, 5]
Enter Element to be Searched :1
Element found at 0th index
1. Sequential Search
2. Sentinal Search
3. Binary Search
4. Fibonacci Search
5. Exit
Enter choice5

'''
```