Started on	Wednesday, 26 March 2025, 1:34 PM
State	Finished
Completed on	Wednesday, 26 March 2025, 2:39 PM
Time taken	1 hour 5 mins
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 2.00 out of 2.00

Write a python program to search an element in the given sorted using iterative binary search.

For example:

Test	Input	Result
`binarySearch(arr, 0, len(arr)-1, x)`	5 2 3 4 10 40 10	`Element is present at index  3`
`binarySearch(arr, 0, len(arr)-1, x)`	4 2 5 9 17 3	`Element is not present in array`

Answer: (penalty regime: 0 %)

```
def binarySearch(arr,low,high,x):
    if low<=high:
        mid=(low+high)//2
        if arr[mid]==x:
            return mid
        elif arr[mid]>x:
            return binarySearch(arr,low,mid-1,x)
        else:
            return binarySearch(arr,mid+1,high,x)
    else:
        return -1
arr=[]
n=int(input())
for i in range(0,n):
    ele= int(input())
    arr.append(ele)
x=int(input())
res= binarySearch(arr,0,len(arr)-1,x)
if res==-1:
    print("Element is not present in array")
else:
    print("Element is present at index ",res)
```

	Test	Input	Expected	Got	
✔	`binarySearch(arr, 0, len(arr)-1, x)`	5 2 3 4 10 40 10	`Element is present at index  3`	`Element is present at index  3`	✔
✔	`binarySearch(arr, 0, len(arr)-1, x)`	4 2 5 9 17 3	`Element is not present in array`	`Element is not present in array`	✔

	Test	Input	Expected	Got	
✔	`binarySearch(arr, 0, len(arr)-1, x)`	6 10 20 30 45 63 79 63	`Element is present at index  4`	`Element is present at index  4`	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **2**

Correct

Mark 2.00 out of 2.00

Write a python program to implement binary search on the given list of characters using iterative method

For example:

Test	Input	Result
binarySearchAppr(arr, 0, len(arr)-1, x)	5 m a n g o n	Element is present at index 3
binarySearchAppr(arr, 0, len(arr)-1, x)	6 S w e e t y s	Element is not present in array

Answer: (penalty regime: 0 %)

```
def binarySearchAppr(arr,low,high,x):
    for i in range(0,len(arr)):
        if arr[i]==x:
            return i
    return -1
n=int(input())
arr=[]
for i in range(0,n):
    arr.append(input())
x=input()
res=binarySearchAppr(arr,0,len(arr)-1,x)
if(res==-1):
    print("Element is not present in array")
else:
    print("Element is present at index",n-res)
```

	Test	Input	Expected	Got	
✔	binarySearchAppr(arr, 0, len(arr)-1, x)	5 m a n g o n	Element is present at index 3	Element is present at index 3	✔
✔	binarySearchAppr(arr, 0, len(arr)-1, x)	6 S w e e t y s	Element is not present in array	Element is not present in array	✔

	Test	Input	Expected	Got	

✔	`binarySearchAppr(arr, 0, len(arr)-1, x)`	`7`
		`ba\x08`
		`\x08`
		`a`
		`n`
		`a`
		`n`
		`a`
		`b`
		`s`
			`Element is not present in array`	`Element is not present in array`	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **3**

Correct

Mark 2.00 out of 2.00

Write a python program to implement binary search on the given list of float values using iterative method

For example:

Test	Input	Result
`binarySearchAppr(arr, 0, len(arr)-1, x)`	5 3.2 6.1 4.5 9.6 8.3 6.1	`Element is present at index 2`
`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	`Element is present at index 3`

Answer: (penalty regime: 0 %)

```
def binarySearchAppr(arr, left, right, target):
    while left <= right:
        mid = (left + right) // 2

        if arr[mid] == target:
            return f"Element is present at index {mid}"
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return f"Element is not present in array"

# Example usage:
if __name__ == "__main__":
    arr = []
    n=int(input())
    for i in range(n):
        arr.append(float(input()))
    arr.sort()
    x=float(input())
    result = binarySearchAppr(arr, 0, len(arr) - 1, x)
```

	Test	Input	Expected	Got	
✔	`binarySearchAppr(arr, 0, len(arr)-1, x)`	5 3.2 6.1 4.5 9.6 8.3 6.1	`Element is present at index 2`	`Element is present at index 2`	✔
✔	`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	`Element is present at index 3`	`Element is present at index 3`	✔

	Test	Input	Expected	Got	
✔	binarySearchAppr(arr, 0, len(arr)-1, x)	8 2.1 6.3 5.2 4.2 9.3 6.7 5.6 9.8 7.2	Element is not present in array	Element is not present in array	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **4**

Correct

Mark 2.00 out of 2.00

Write a python program to implement linear search on the given tuple of string values.

note: As the tuple is immutable convert the list to tuple to perform search

For example:

Input	Result
5 ram john akbar seetha oviya john	Tuple: john found
4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found

Answer: (penalty regime: 0 %)

```
def LS(list,n,key):
    for i in range(0,n):
        if(list[i]==key):
            return i
    return -1
list=[]
n=int(input())
for i in range(0,n):
    temp=input()
    list.append(temp)
key=input()
res=LS(list,n,key)
if(res==-1):
    print("Tuple: %s not found"%key)
else:
    print("Tuple: %s found"%key)
```

	Input	Expected	Got	
✔	5 ram john akbar seetha oviya john	Tuple: john found	Tuple: john found	✔
✔	4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found	Tuple: rakesh not found	✔

	Input	Expected	Got	
✔	6 rose jasmine tulips marigold hibiscus lotus lilly	Tuple: lilly not found	Tuple: lilly not found	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **5**

Correct

Mark 2.00 out of 2.00

Write a python program to implement linear search on the given integer tuple.

For example:

Input	Result
5 10 26 48 96 35 26	26 Found

Answer: (penalty regime: 0 %)

```
1
2
3  def search(tuple1,x):
4      for value in tuple1:
5          if(value==x):
6              print("%d Found"%x)
7              return 0
8      print("Tuple: %d not found"%x)
9
10 List=[]
11 n=int(input())
12 for i in range(n):
13     List.append(int(input()))
14 tuple1=tuple(List)
15 x=float(input())
16 search(tuple1,x)
17
```

	Input	Expected	Got	
✔	5 10 26 48 96 35 26	26 Found	26 Found	✔
✔	6 5 6 2 4 7 9 2	2 Found	2 Found	✔
✔	4 12 20 31 40 20	20 Found	20 Found	✔

Passed all tests! ✔

Correct!

Marks for this submission: 2.00/2.00.