

## EX-1.9

### Title :

Check if a given number  $x$  exists in a sorted array `arr` using binary search.

### Aim:

To design and implement a Python program to search for a given key in a sorted array using binary search.

### Procedure:

1. Read input size  $n$ .
2. Read  $n$  elements into an array.
3. Read the key to search for.
4. Implement binary search on the sorted array:
  - Initialize  $low = 0, high = n - 1$ .
  - While  $low \leq high$ :
    - Calculate  $mid = (low + high) // 2$ .
    - If  $arr[mid] == key$ , return the position.
    - If  $arr[mid] < key$ , search in the right half ( $low = mid + 1$ ).
    - Else, search in the left half ( $high = mid - 1$ ).
5. If the key is found, print its position (1-indexed).
6. If not found, print not found message.

**Algorithm:**

1. Start
2. Read n
3. Read array arr
4. Sort the array (since binary search requires sorted array)
5. Read key k
6. Set low = 0, high = n-1
7. While low <= high
  - mid = (low + high) // 2
  - If arr[mid] == k, return mid + 1
  - Else if arr[mid] < k, set low = mid + 1
  - Else, set high = mid - 1
8. If not found, print not found message
9. Stop

**Input:**

8

3 4 6 -9 10 8 9 30

10

**Output:**

Element 10 is found at position 5

**Program :**

```
def binarySearch(arr, key):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == key:
            return mid + 1 # 1-indexed position
        elif arr[mid] < key:
            low = mid + 1
        else:
            high = mid - 1
    return -1

n = int(input("Enter size of array: "))
arr = list(map(int, input("Enter array elements: ").split()))
arr.sort() # Sorting before binary search, as binary search requires sorted array
key = int(input("Enter key to search: "))

pos = binarySearch(arr, key)
if pos != -1:
    print(f"Element {key} is found at position {pos}")
else:
    print(f"Element {key} is not found")
```

## Performance Analysis:

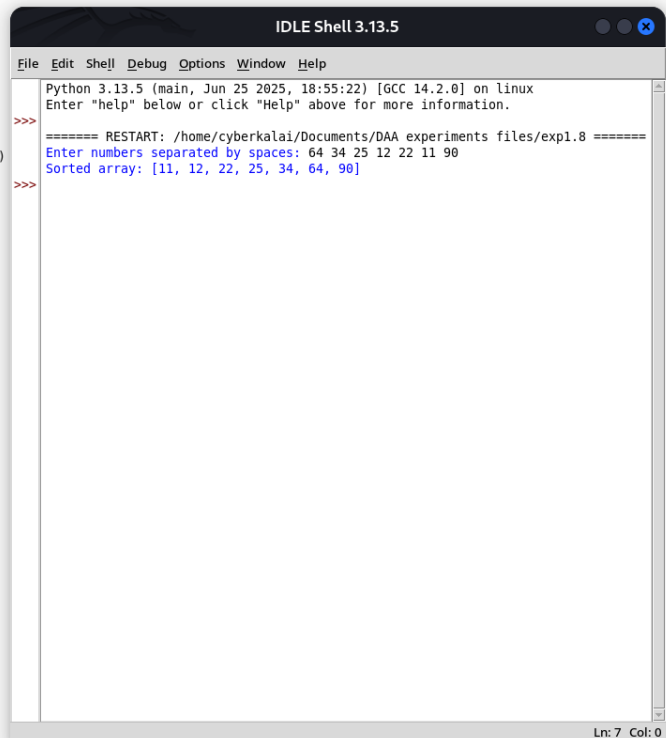
**Time Complexity:**  $O(\log n)$

**Space Complexity:**  $O(1)$

## program output:

```
File Edit Format Run Options Window Help
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        swapped = False
        for j in range(n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
        if not swapped:
            break
    return arr

nums = list(map(int, input("Enter numbers separated by spaces: ").split()))
sorted_nums = bubble_sort(nums)
print("Sorted array:", sorted_nums)
```



```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (main, Jun 25 2025, 18:55:22) [GCC 14.2.0] on linux
Enter "help" below or click "Help" above for more information.
>>>
===== RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.8 =====
Enter numbers separated by spaces: 64 34 25 12 22 11 90
Sorted array: [11, 12, 22, 25, 34, 64, 90]
>>>
```

## Result :

Thus the given program Binary Search is executed and got output successfully.