162) Implement the Binary Search algorithm in a programming language of your choice and test it on the array 5,10,15,20,25,30,35,40,45 to find the position of the element 20. Execute your code and provide the index of the element 20. Modify your implementation to count the number of comparisons made during the search process. Print this count along with the result.

Input : N= 9, a[] = {5,10,15,20,25,30,35,40,45}, search key = 20
Output : 4
Test cases
Input : N= 6, a[] = {10,20,30,40,50,60}, search key = 50
Output : 5
Input : N= 7, a[] = {21,32,40,54,65,76,87}, search key = 32
Output : 2

AIM : To find the position of the element 20. Execute your code and provide the index of the element 20. Modify your implementation to count the number of comparisons made during the search process. Print this count along with the result.

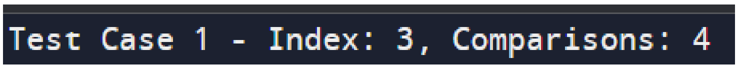
PROGRAM :

```
def binary_search(arr, search_key):
    left, right = 0, len(arr) - 1
    comparisons = 0

    while left <= right:
        mid = (left + right) // 2
        comparisons += 1
        if arr[mid] == search_key:
            return mid, comparisons
        elif arr[mid] < search_key:
            left = mid + 1
        else:
            right = mid - 1

    return -1, comparisons

arr1 = [5, 10, 15, 20, 25, 30, 35, 40, 45]
search_key1 = 20
index1, comparisons1 = binary_search(arr1, search_key1)
print(f"Test Case 1 - Index: {index1}, Comparisons: {comparisons1}")
```

OUTPUT: 

```
Test Case 1 - Index: 3, Comparisons: 4
```

TIME COMPLEXITY:O(log n)