PRACTICAL-5

AIM:-Implement a function of binary search and count the steps executed by function on various inputs for best case and worst case. Also write complexity in each case and draw a comparative chart.

Iterative method-

Code:

```
 #include <stdio.h>
//Iteration method
int step=0; int
compare=0;
int binarysearch(int arr[], int low, int high,int x) {
for(int i=0; low<=high ;i++,step++){
compare++;
int mid = (low+high)/2;
 if(arr[mid] == x){
 return mid;
          compare++;
       }
       else if(arr[mid]>x){
high = mid-1;
          compare++;
       }
       else{
low=mid+1;
          compare++;
       }     }
return -1;
} int main()
{
          int n;
          printf("Enter no of element:");
          scanf("%d",&n);
          int arr[n];
          printf("Enter elements:");
          for(int i=0;i<n;i++){
            scanf("%d",&arr[i]);
          }
          int sr;
          printf("Enter element to search:");
          scanf("%d",&sr);
          int result=binarysearch(arr,0,n-1,sr);
          if (result==-1){
              printf("!Element not found!");
```

```
        }
        else{
           printf("\n\nElement found at position : %d",result+1);
printf("\nNo of steps: %d",step);
           printf("\nNo of comparision: %d",compare);
        }
        return 0;
}
```

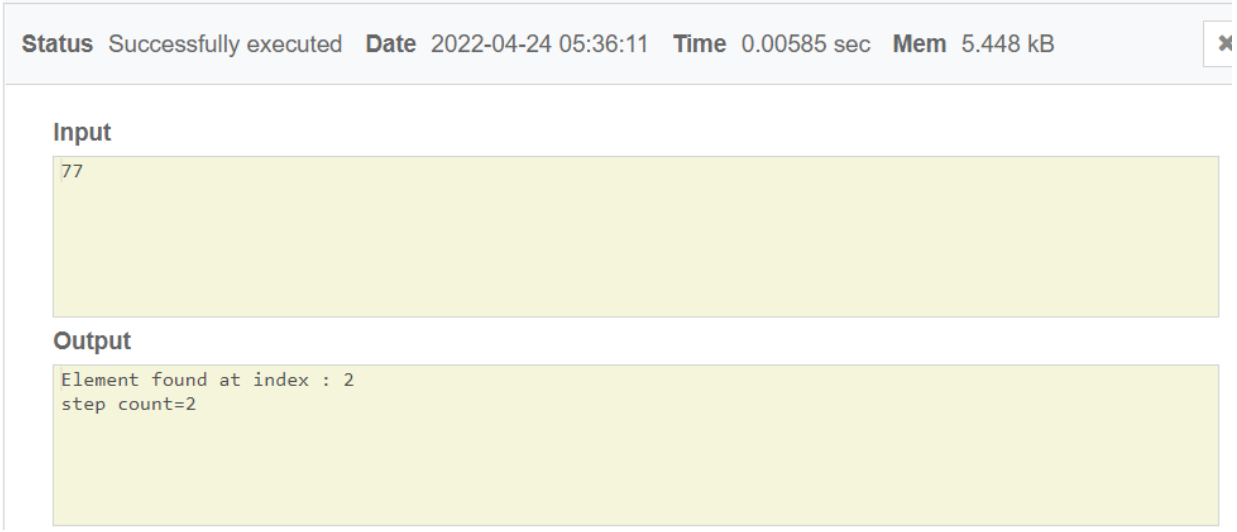
Output-

Recursive method-

Code:

```
#include <stdio.h>
//Recursive method
int count=0;
int binarysearch(int arr[], int low, int high,int x) {
   while(low<=high){
int mid=(low+high)/2;
if(arr[mid]==x){
count++;
return mid;
      }
      else if(arr[mid]<x){
low = mid+1;
         count++;
}
 else{
high=mid+1;
count++;
      }
 }
return -1;
}
 int main()
{
 int
arr[]={85,77,86,113,102,84,109,95};
int n = 8;
int element=86;
 int found_index=binarysearch(arr,0,n-1,element);
if(found_index==-1){
   printf("Element not found in the array");
 }
 else{
("Element found at index : %d",found_index);
 }
 printf("\nstep count=%d",count);
 return 0;
}
```

Output:


Status Successfully executed Date 2022-04-24 05:36:11 Time 0.00585 sec Mem 5.448 kB
Input
77
Output
Element found at index : 2
step count=2