

# Practical:-4

**Implement a function of sequential 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.**

## Input Code:-

```
#include<stdio.h>
#include <stdlib.h>
#include <time.h>

void linear_search(int list[],int n,int key);
void main()
{
    int list[200],n,i,key;
    printf("\nHow many value to enter:-");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("\nEnter values:-");
        scanf("%d",&list[i]);
    }
    printf("\nEnter the searching element:-");
    scanf("%d",&key);
    linear_search(list,n,key);

    for(i=0;i<n;i++)
    {
        printf("\n value is===%d",list[i]);
    }
}

void linear_search(int list[],int n,int key){
    int flag=0,i;
    for(i=0;i<n;i++){
        if(list[i]==key){
```

```

        printf("\n value %d is found %d location",list[i],i);
        flag=1;
        break;
    }
}
if(flag==0)
    printf("\n Value is NOT found");
}

```

**Output:-**

How many value to enter:-5

Enter values:-16

Enter values:-98

Enter values:-35

Enter values:-26

Enter values:-10

Enter the searching element:-35

value 35 is found 2 location

value is==16

value is==98

value is==35

value is==26

value is==10

Process returned 5 (0x5) execution time : 12.048 s

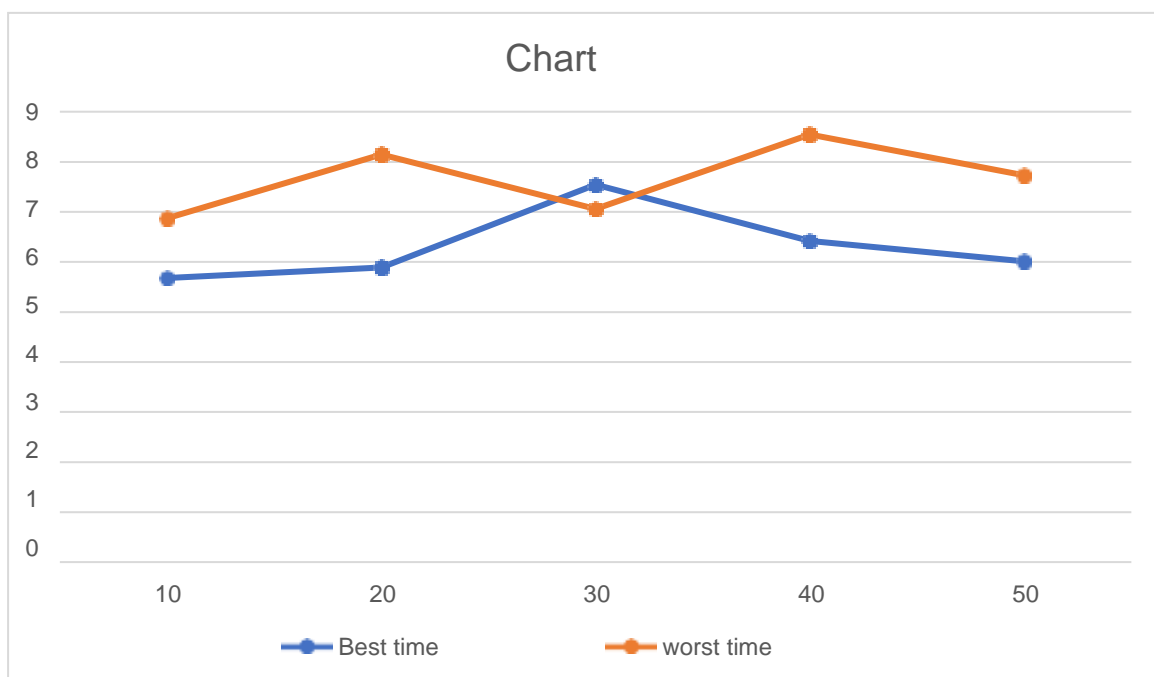
Press any key to continue.

**\* No. of Steps:**

values	Best time complexities	Worst time complexities
10	2	20
20	2	40
30	2	60
40	2	80
50	2	100

- **Time complexity**

values	Best time complexities	Worst time complexities
10	5.670	6.855
20	5.880	8.150
30	7.550	7.040
40	6.420	8.560
50	6.010	7.720



**Conclusion:-** we can find the best case and worst case time complexity of sequential search

**Best case time complexity :**

**$O(1)$  Worst case time complexity :  $O(n)$**