Practical no -5

Aim: Write a program to search the element from array using linear search.

Linear Search:

Computer systems are often used to store large amounts of data from which individual records must be retrieved according to some search criterion. Thus the efficient storage of data to facilitate fast searching is an important issue.

The search proceeds by sequentially comparing the key with elements in the list, and continues until either we find a match or the end of the list is encountered. If we find a match, the search terminates successfully by returning the index of the element in the list which has matched. If the end of the list is encountered without a match, the search terminates unsuccessfully.

Linear search is the simplest method of searching. in this method the element to be found is sequentially searched in the list .this method can be applied in sorted an unsorted order of list .searching is a case of a sorted order list start from 0^{th} element and contains the element is found. Or an element whose value is greater than the value being searched is reached. as again this ,searching in case of unsorted list starts from 0^{th} element and contains until the element is found or the end of list is reached.

Explanation

1. In the best case, the search procedure terminates after one comparison only, whereas in the worst case, it will do n comparisons.
2. On average, it will do approximately $n/2$ comparisons, since the search time is proportional to the number of comparisons required to be performed.
3. The linear search requires an average time proportional to $O(n)$ to search one element. Therefore to search n elements, it requires a time proportional to $O(n^2)$.
4. We conclude that this searching technique is preferable when the value of n is small. The reason for this is the difference between n and n^2 is small for smaller values of n.

Algorithm:

Linear_search (l,n,elemnt)

Where l--→represent the list of element

n--→represent the no of element in the list.

Element---→ represent the value to be searched in the list

Steps:

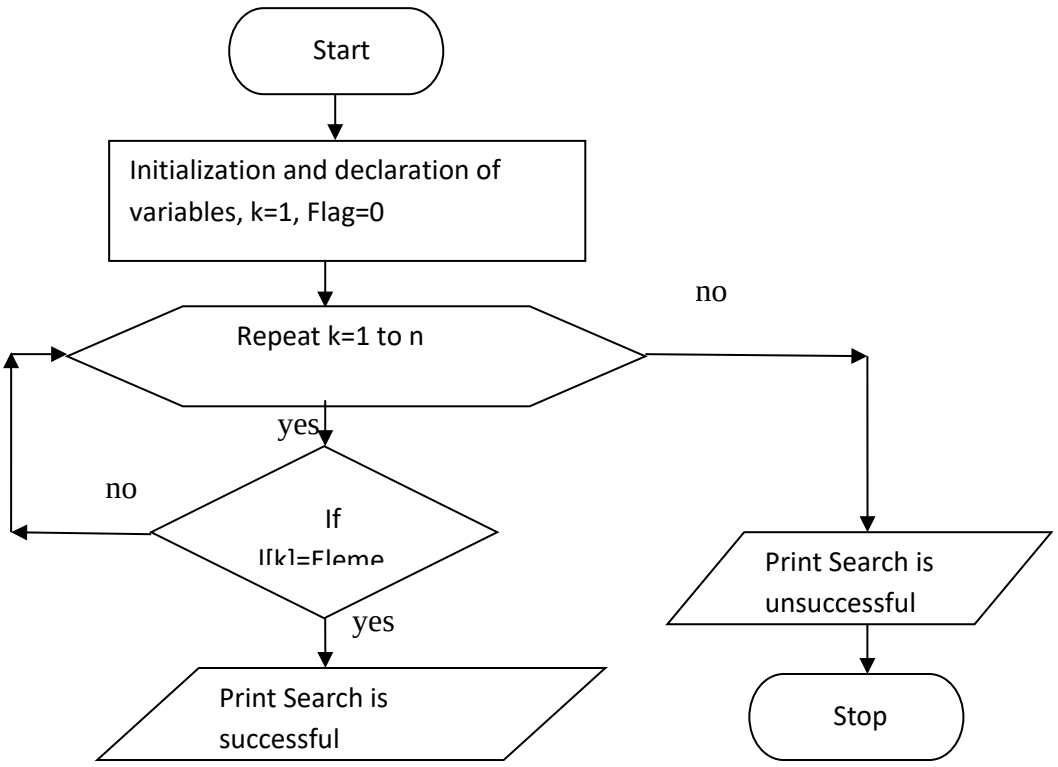
1. [initialize]

   K=1.

   Flag=1.

2. Repeat through step 3 for k=1,2,3,-------,n.
3. If l[k]=Element
   a. Flag =0
   b. Print" search is successful".
4. If flag print "Search is unsuccessful"
5. Exit.

Flowchart:



Program:

```
#include<stdio.h>

#include<conio.h>

void main()
```

```
{
int arr[100],i,found=0,num,dim;
clrscr();
printf("Enter the dimension");
scanf("%d",&dim);
printf(“Enter the array element”)
for(i=1;i<=dim;i++)
        {
        scanf("%d",&arr[i]);
        }
printf("\n Enter the element to search");
scanf("%d",&num);
for(i=1;i<=dim;i++)
        {
        if(num==arr[i])
        {
                found=1;
                break;
        }
}
if(found==1)
        printf("%d Number is present",num);
else
        printf("%d Number is not present",num);
getch();
}
```

Output:

Enter the dimension 5

Enter the array element:

8

9

5

4

3

Enter the element to search:

9

9 Number is present