**Linear Search**

**#include<stdio.h>**

**int main(){**

**int a[10],i,n,m,c=0;**

**printf("Enter the size of an array: ");**

**scanf("%d",&n);**

**printf("Enter the elements of the array: ");**

**for(i=0;i<=n-1;i++){**

**scanf("%d",&a[i]);**

**}**

**printf("Enter the number to be search: ");**

**scanf("%d",&m);**

**for(i=0;i<=n-1;i++){**

**if(a[i]==m){**

**c=1;**

**break;**

**}**

**}**

**if(c==0)**

**printf("The number is not in the** list");

**else**

**printf("The number is found");**

**return 0;**

**}**

**Output:**

**Enter the size of an array: 7**

**Enter the elements of the array: 5 0 7 1 13 2 9**

**Enter the number to be search: 13**

**The number is found**

**Binary Search**

**#include<stdio.h>**

**int main()**

**{**

**int a[10],i,n,m,c=0,l,u,mid;**

**printf("Enter the size of an array: ");**

**scanf("%d",&n);**

**printf("Enter the elements in ascending order: ");**

**for(i=0;i<n;i++){**

**scanf("%d",&a[i]);**

**}**

**printf("Enter the number to be search: ");**

**scanf("%d",&m);**

**l=0,u=n-1;**

**while(l<=u){**

**mid=(int)(l+u)/2;**

**if(m==a[mid]){**

**c=1;**

**break;**

**}**

         else if(m<a[mid]){

             u=mid-1;

         }

         else

             l=mid+1;

    }

    if(c==0)

         printf("The number is not found.");

    else

         printf("The number is found.");

    return 0;

}

Output of the program:

Enter the size of an array: 8

Enter the elements in ascending order: 4 7 8 11 21 23 34 35

Enter the number to be search: 11

The number is found.

**The binary search algorithm** begins by comparing the target value to the value of the middle element of the sorted array. If the target value is equal to the middle element's value, then the position is returned and the search is finished. If the target value is less than the middle element's value, then the search continues on the lower half of the array; or if the target value is greater than the middle element's value, then the search continues on the upper half of the array. This process continues, eliminating half of the elements, and comparing the target value to the value of the middle element of the remaining elements - until the target value is either found (and its associated element position is returned), or until the entire array has been searched (and "not found" is returned).