1. Linear Search

CODE:

```
#include<stdio.h>
int main()
{
int n,key,i,f=0;

printf("Enter number of elements: \n");
scanf("%d",&n);
 int arr[n];

printf("Enter the elements: \n");
for(i=0;i<n;i++){
scanf("%d",&arr[i]);
}
printf("Enter the element to be searched: \n");
scanf("%d",&key);
for (i=0;i<n;i++)
    {
if (arr[i]==key)
      {
printf("The number is found at index %d",i);
      f=1;
      break;
  }
    }
```

```
    if(f==0){
printf("the number is not found");
}

return 0;
}
```

```
Enter number of elements:
5
Enter the elements:
5
4
3
7
8
Enter the element to be searched:
7
The number is found at index 3
```

```
Enter number of elements:
5
Enter the elements:
1
2
7
5
8
Enter the element to be searched:
3
the number is not found
```

2. Binary Search

CODE:

```
#include <stdio.h>
int main() {

   int n, key, i, low, high, mid, j, temp;
  printf("Enter number of elements : ");

   scanf("%d", &n);
   int arr[n];
    printf("Enter elements: \n");

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

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

   }

   for (i = 0; i < n - 1; i++) {
      for (j = 0; j < n - i - 1; j++) {
         if (arr[j] > arr[j + 1]) {
            // Swap
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
         }
      }
   }
```

```
 printf("Sorted array in ascending order:\n");
 for (i = 0; i < n; i++) {
    printf("%d ", arr[i]);
 }

printf("Enter the element to search: ");

 scanf("%d", &key);
  low = 0;

 high = n - 1;

 int f = 0;
  while (low <= high) {

    mid = (low + high) / 2;
     if (arr[mid] == key) {

       printf("Element found at index %d\n", mid + 1);

       f = 1;

       break;

    }
```

```
        else if (arr[mid] < key) {

            low = mid + 1;

        }

        else {

            high = mid - 1;

        }

    }
    if (f==0) {

        printf("Element not found.\n");

    }
    return 0;

}
```

```
Enter number of elements : 4
Enter elements:
2
1
3
4
Sorted array in ascending order:
1 2 3 4 Enter the element to search: 3
Element found at index 3


=== Code Execution Successful ===
```

```
Enter number of elements : 5
Enter elements:
34
2
1
56
78
Sorted array in ascending order:
1 2 34 56 78 Enter the element to search: 3
Element not found.
```