ASSIGNMENT NO 7

PARTH JOHRI

2K20/B17/33

THEORY

Linear search in C

Linear search in C to find whether a number is present in an array.

If it's present, then at what **location** it occurs.

It is also known as a **sequential search.**

It is **straightforward** and works as follows:

We compare each element with the element to search until we find it or the list ends.

The time required to search an element using the algorithm depends on the size of the list.

In the best case, it's present at the **beginning** of the list, in the **worst**-case, element is present at the end. Its time complexity is **O(n).**

Binary search in C

Binary search in **C** language to find an element in a **sorted array**.

If the array isn't sorted, you must **sort** it using a **sorting** technique such as **merge sort**.

If the element to search is present in the list, then we print its **location**. The program assumes that the input numbers are in ascending order.

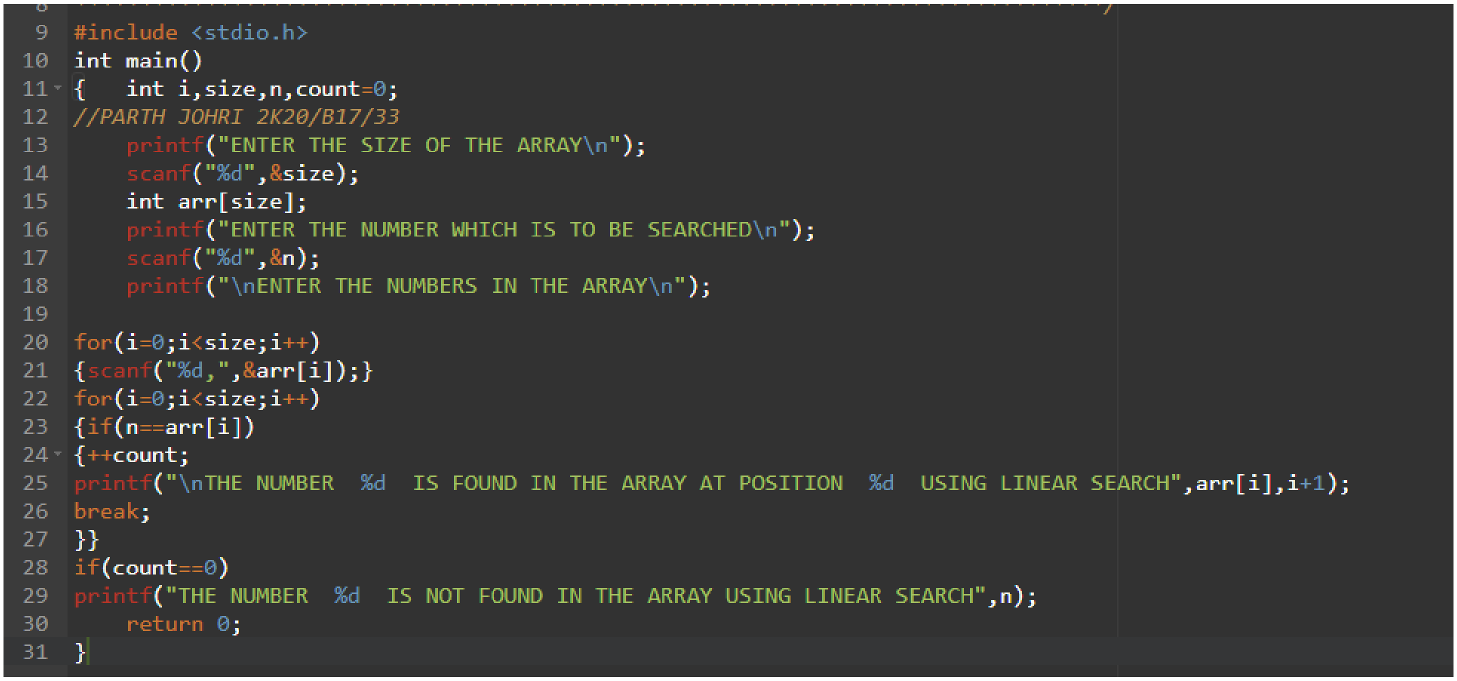
Binary search is **faster** than the **linear search**.

Its time complexity is **O(log(n)),** while that of the linear search is **O(n).**

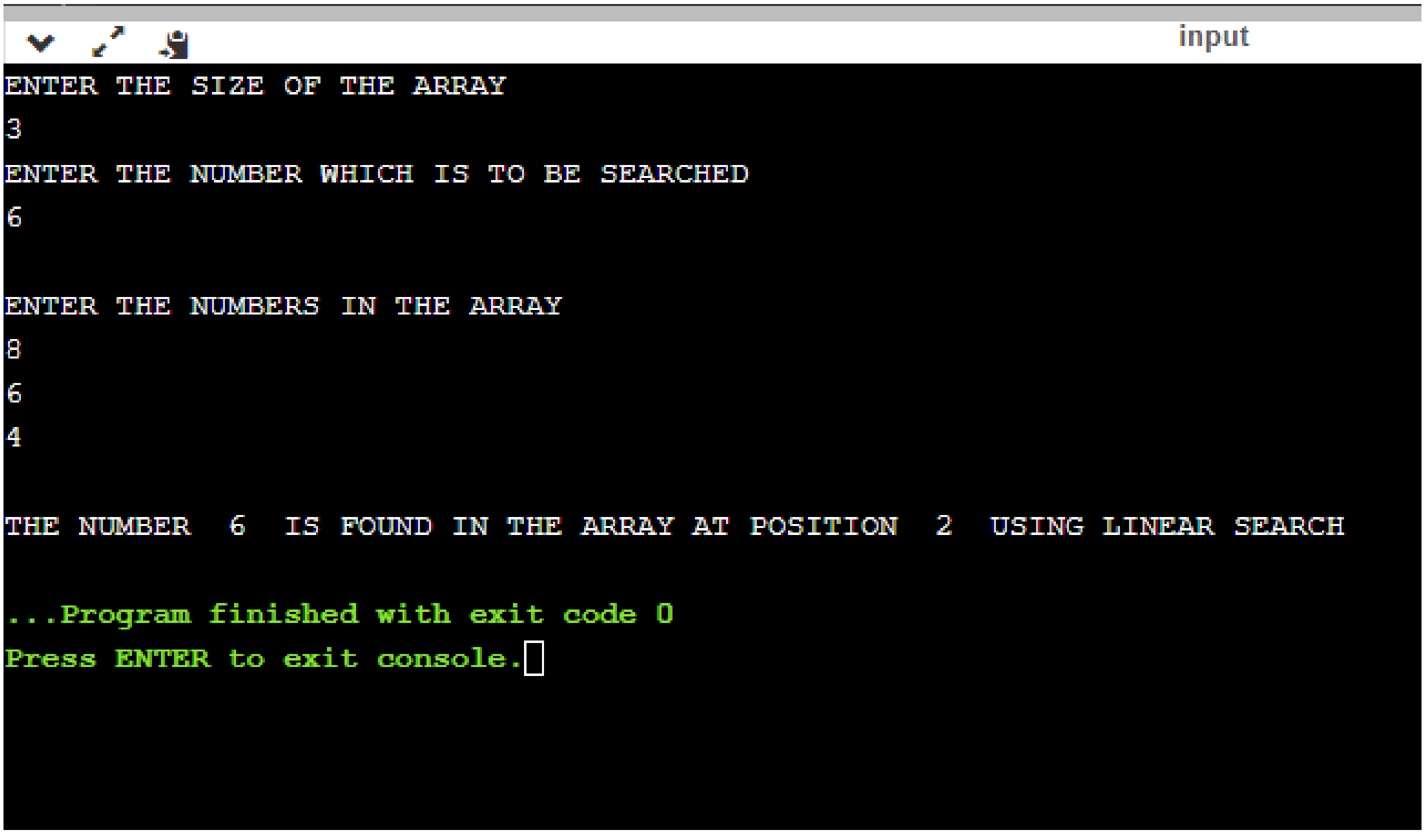
However, the list should be in **ascending/descending** order, **hashing** is **rapid** than **binary search** and perform searches in constant time.

Q1

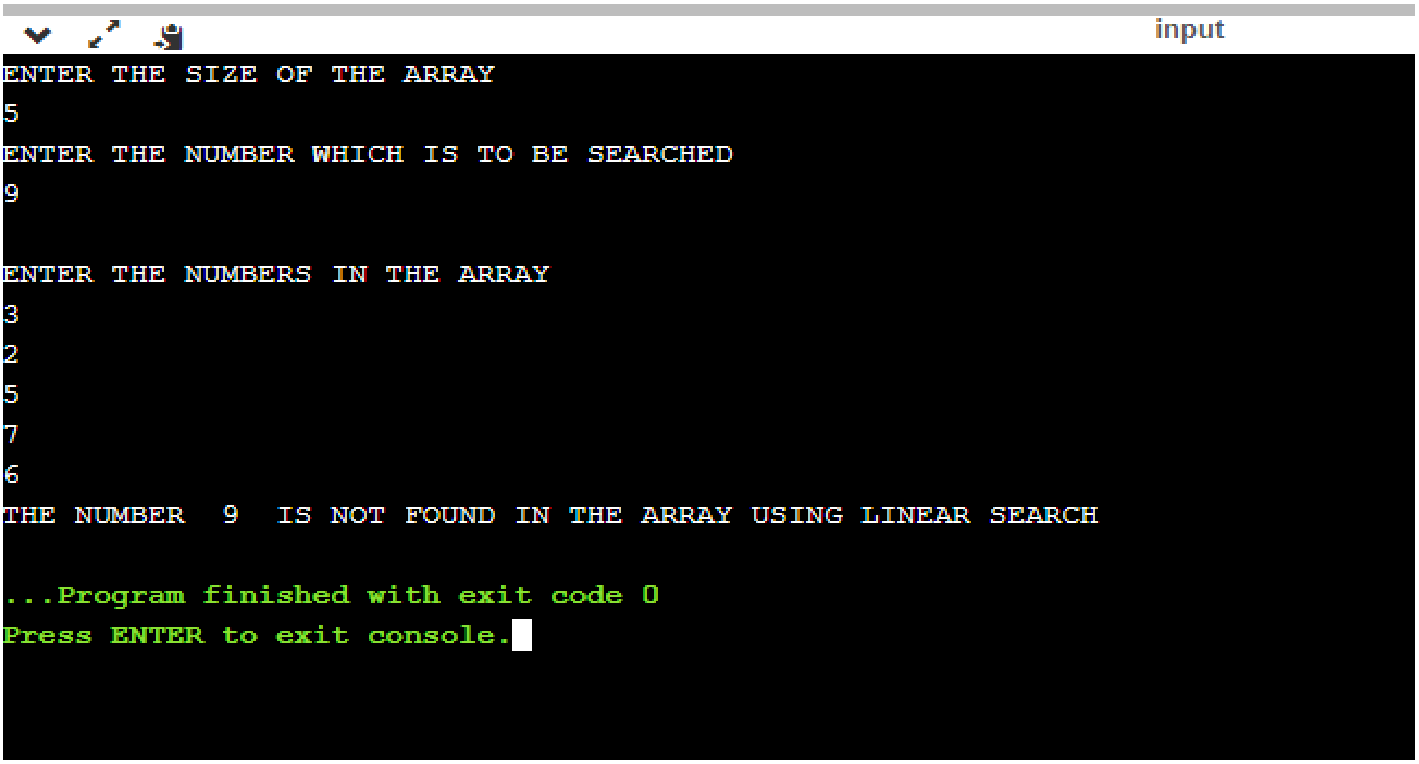
```
#include <stdio.h>

int main()

{   int i,size,n,count=0;

//PARTH JOHRI 2K20/B17/33

   printf("ENTER THE SIZE OF THE ARRAY\n");

   scanf("%d",&size);

   int arr[size];

   printf("ENTER THE NUMBER WHICH IS TO BE SEARCHED\n");

   scanf("%d",&n);

   printf("\nENTER THE NUMBERS IN THE ARRAY\n");


for(i=0;i<size;i++)

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

for(i=0;i<size;i++)

{if(n==arr[i])

{++count;

printf("\nTHE NUMBER  %d  IS FOUND IN THE ARRAY AT POSITION  %d  USING LINEAR SEARCH",arr[i],i+1);

break;

}}

if(count==0)

printf("THE NUMBER  %d  IS NOT FOUND IN THE ARRAY USING LINEAR SEARCH",n);

   return 0;

}
```



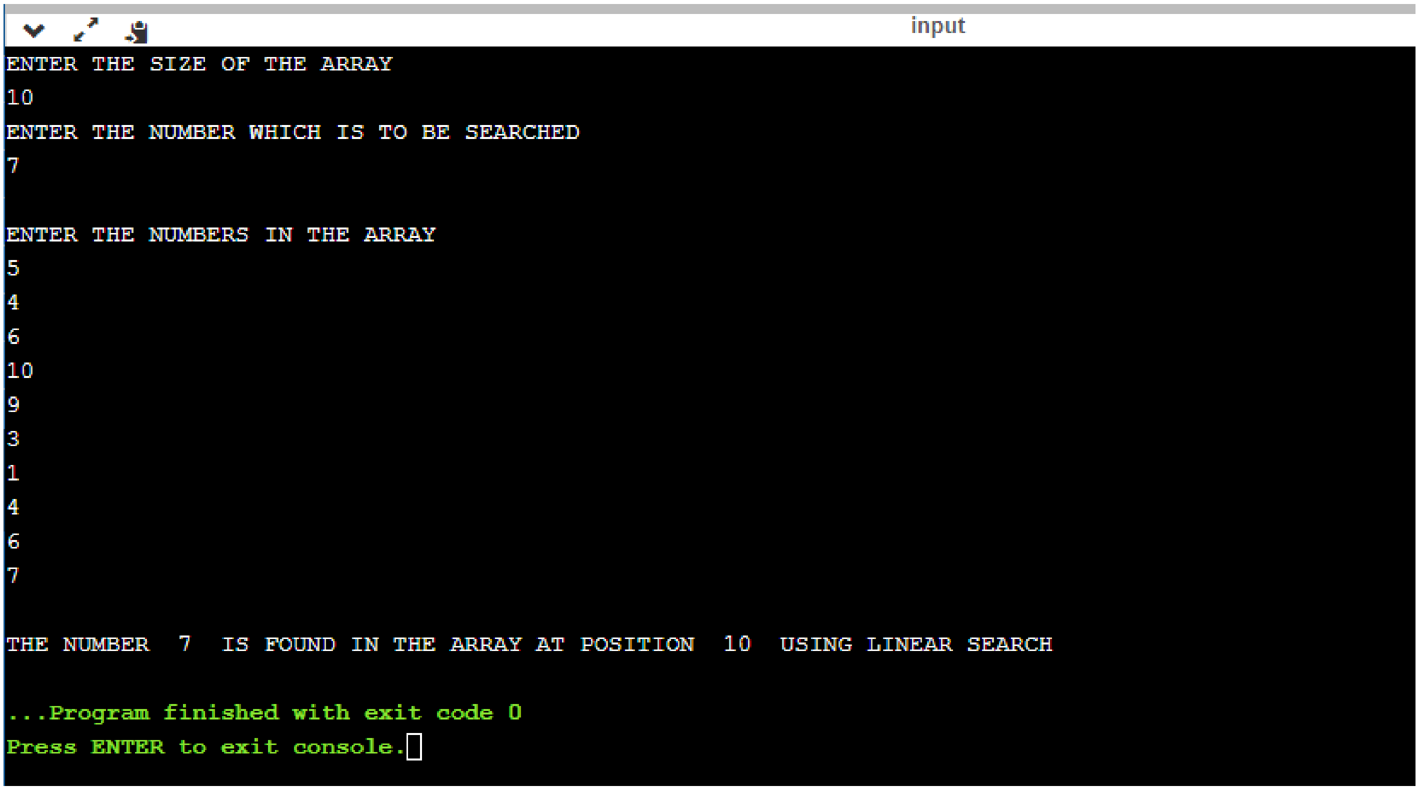
```c
#include <stdio.h>
int main()
{   int i,size,n,count=0;
//PARTH JOHRI 2K20/B17/33
    printf("ENTER THE SIZE OF THE ARRAY\n");
    scanf("%d",&size);
    int arr[size];
    printf("ENTER THE NUMBER WHICH IS TO BE SEARCHED\n");
    scanf("%d",&n);
    printf("\nENTER THE NUMBERS IN THE ARRAY\n");

for(i=0;i<size;i++)
{scanf("%d,",&arr[i]);}
for(i=0;i<size;i++)
{if(n==arr[i])
{++count;
printf("\nTHE NUMBER  %d  IS FOUND IN THE ARRAY AT POSITION  %d  USING LINEAR SEARCH",arr[i],i+1);
break;
}}
if(count==0)
printf("THE NUMBER  %d  IS NOT FOUND IN THE ARRAY USING LINEAR SEARCH",n);
    return 0;
}
```



```
ENTER THE SIZE OF THE ARRAY
3
ENTER THE NUMBER WHICH IS TO BE SEARCHED
6

ENTER THE NUMBERS IN THE ARRAY
8
6
4

THE NUMBER  6  IS FOUND IN THE ARRAY AT POSITION  2  USING LINEAR SEARCH

...Program finished with exit code 0
Press ENTER to exit console.
```



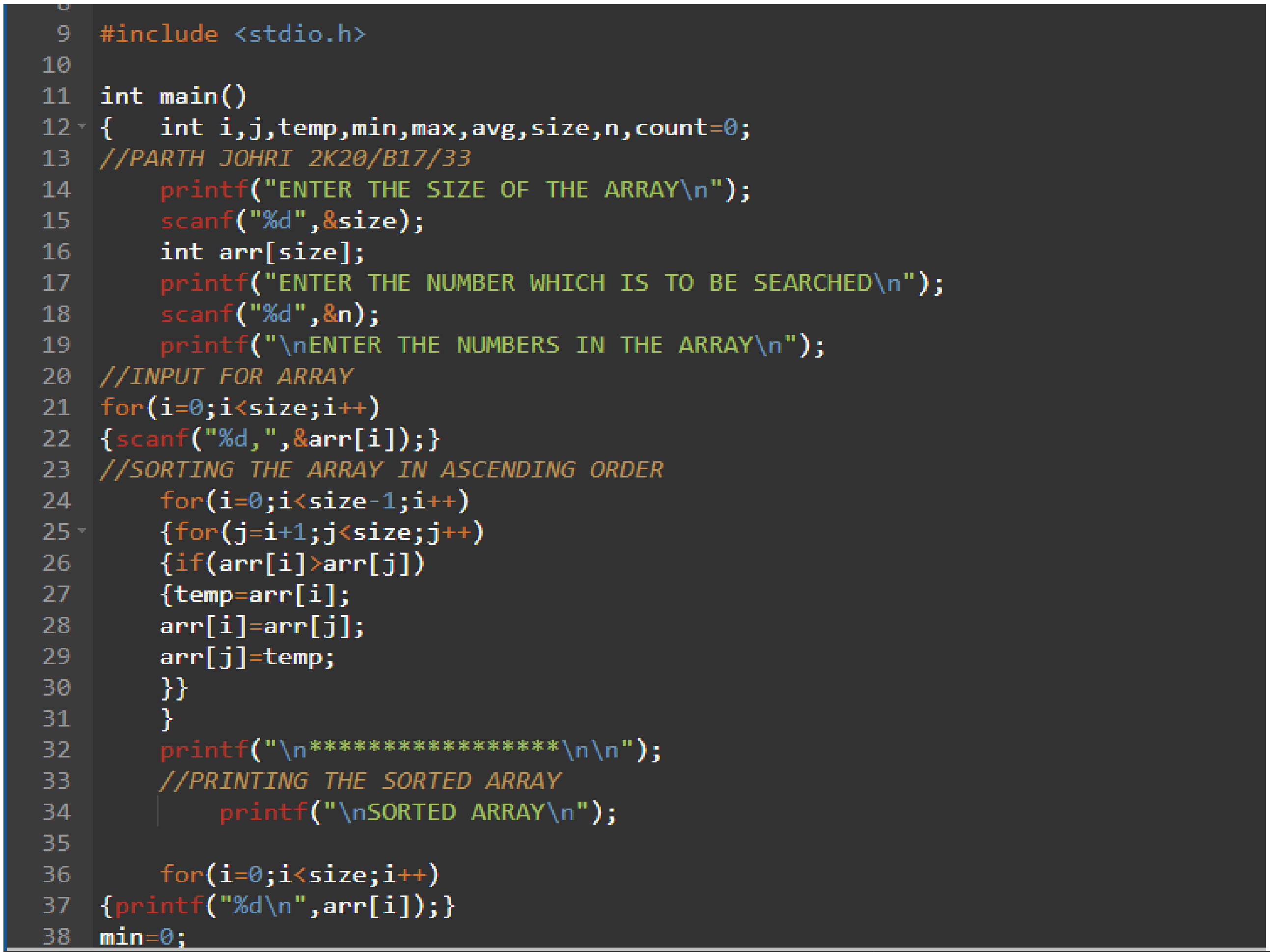
```
ENTER THE SIZE OF THE ARRAY
5
ENTER THE NUMBER WHICH IS TO BE SEARCHED
9

ENTER THE NUMBERS IN THE ARRAY
3
2
5
7
6
THE NUMBER  9  IS NOT FOUND IN THE ARRAY USING LINEAR SEARCH

...Program finished with exit code 0
Press ENTER to exit console.
```



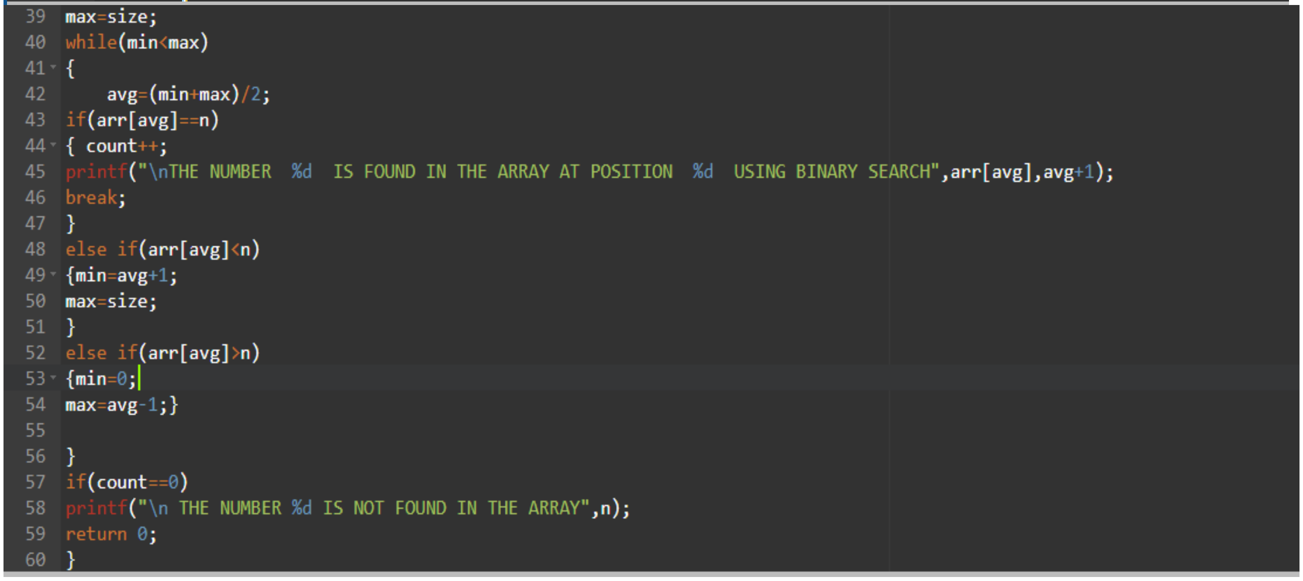
```
ENTER THE SIZE OF THE ARRAY
10
ENTER THE NUMBER WHICH IS TO BE SEARCHED
7

ENTER THE NUMBERS IN THE ARRAY
5
4
6
10
9
3
1
4
6
7

THE NUMBER  7  IS FOUND IN THE ARRAY AT POSITION  10  USING LINEAR SEARCH

...Program finished with exit code 0
Press ENTER to exit console.
```

Q2

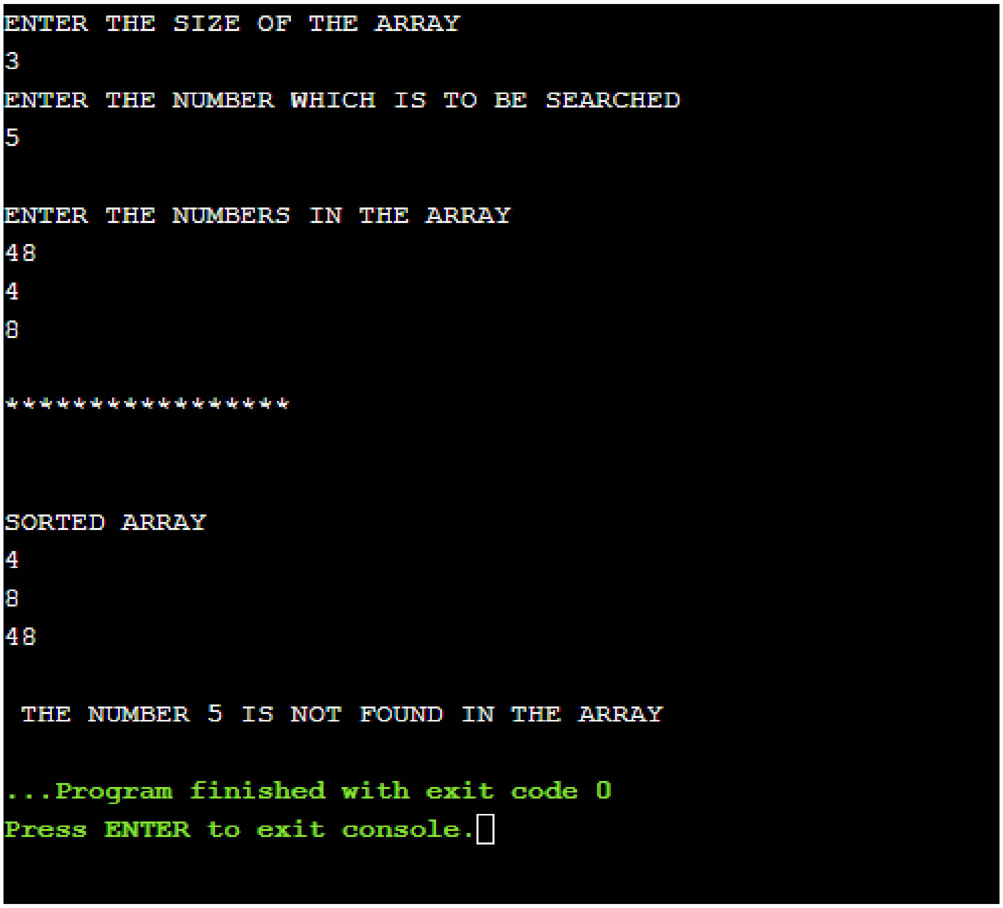
```
#include <stdio.h>

int main()

{   int i,j,temp,min,max,avg,size,n,count=0;

//PARTH JOHRI 2K20/B17/33

   printf("ENTER THE SIZE OF THE ARRAY\n");

   scanf("%d",&size);

   int arr[size];

   printf("ENTER THE NUMBER WHICH IS TO BE SEARCHED\n");

   scanf("%d",&n);

   printf("\nENTER THE NUMBERS IN THE ARRAY\n");

//INPUT FOR ARRAY

for(i=0;i<size;i++)

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

//SORTING THE ARRAY IN ASCENDING ORDER

   for(i=0;i<size-1;i++)

   {for(j=i+1;j<size;j++)

   {if(arr[i]>arr[j])

   {temp=arr[i];

   arr[i]=arr[j];

   arr[j]=temp;

   }}

   }

   printf("\n*****************\n\n");

   //PRINTING THE SORTED ARRAY

      printf("\nSORTED ARRAY\n");


   for(i=0;i<size;i++)

{printf("%d\n",arr[i]);}

min=0;
max=size;
while(min<max)
{
   avg=(min+max)/2;
if(arr[avg]==n)
{ count++;
printf("\nTHE NUMBER  %d  IS FOUND IN THE ARRAY AT POSITION  %d  USING BINARY SEARCH",arr[avg],avg+1);
break;
}
else if(arr[avg]<n)
{min=avg+1;
max=size;
}
else if(arr[avg]>n)
{min=0;
max=avg-1;}


}
if(count==0)
printf("\n THE NUMBER %d IS NOT FOUND IN THE ARRAY",n);
return 0;
}
```



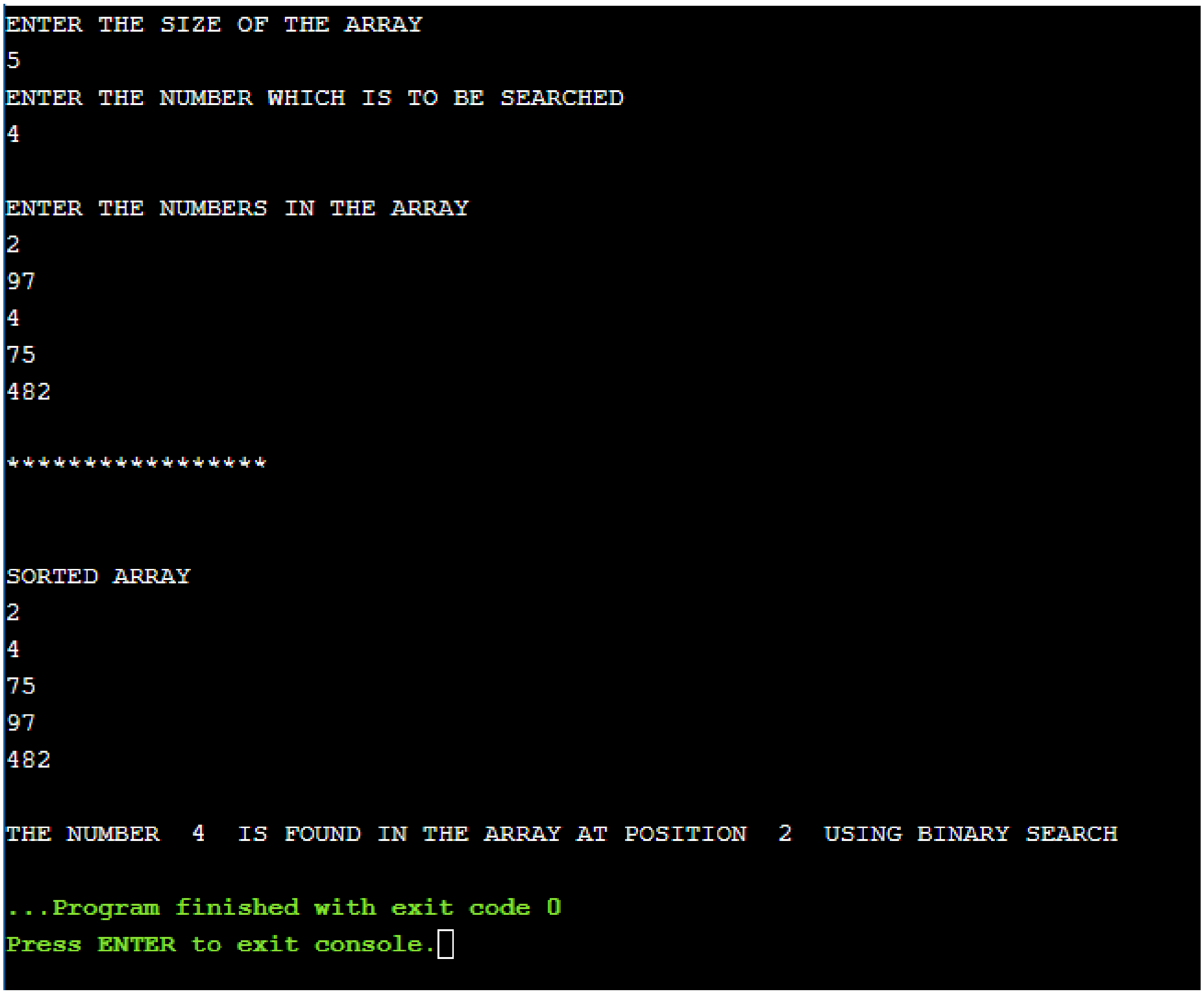
```
#include <stdio.h>

int main()
{   int i,j,temp,min,max,avg,size,n,count=0;
//PARTH JOHRI 2K20/B17/33
    printf("ENTER THE SIZE OF THE ARRAY\n");
    scanf("%d",&size);
    int arr[size];
    printf("ENTER THE NUMBER WHICH IS TO BE SEARCHED\n");
    scanf("%d",&n);
    printf("\nENTER THE NUMBERS IN THE ARRAY\n");
//INPUT FOR ARRAY
for(i=0;i<size;i++)
{scanf("%d,",&arr[i]);}
//SORTING THE ARRAY IN ASCENDING ORDER
    for(i=0;i<size-1;i++)
    {for(j=i+1;j<size;j++)
    {if(arr[i]>arr[j])
    {temp=arr[i];
    arr[i]=arr[j];
    arr[j]=temp;
    }}
    }
    printf("\n*****************\n\n");
    //PRINTING THE SORTED ARRAY
        printf("\nSORTED ARRAY\n");

    for(i=0;i<size;i++)
{printf("%d\n",arr[i]);}
min=0;
```



```
max=size;
while(min<max)
{
    avg=(min+max)/2;
if(arr[avg]==n)
{ count++;
printf("\nTHE NUMBER  %d  IS FOUND IN THE ARRAY AT POSITION  %d  USING BINARY SEARCH",arr[avg],avg+1);
break;
}
else if(arr[avg]<n)
{min=avg+1;
max=size;
}
else if(arr[avg]>n)
{min=0;
max=avg-1;}

}
if(count==0)
printf("\n THE NUMBER %d IS NOT FOUND IN THE ARRAY",n);
return 0;
}
```



```
ENTER THE SIZE OF THE ARRAY
3
ENTER THE NUMBER WHICH IS TO BE SEARCHED
5

ENTER THE NUMBERS IN THE ARRAY
48
4
8

*****************


SORTED ARRAY
4
8
48

 THE NUMBER 5 IS NOT FOUND IN THE ARRAY

...Program finished with exit code 0
Press ENTER to exit console.
```



```
ENTER THE SIZE OF THE ARRAY
5
ENTER THE NUMBER WHICH IS TO BE SEARCHED
4

ENTER THE NUMBERS IN THE ARRAY
2
97
4
75
482

*****************


SORTED ARRAY
2
4
75
97
482

THE NUMBER  4  IS FOUND IN THE ARRAY AT POSITION  2  USING BINARY SEARCH

...Program finished with exit code 0
Press ENTER to exit console.
```