

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void insertionSort(int arr[], int n);
5
6  void main()
7  {
8      int arr[100], i, n, x, choice, flag = 0;
9      printf("\t --- WELCOME TO IMPLEMENTATION OF BINARY
          SEARCH --- \n");
10     printf("\n Enter the number of elements of the array
          [maximum size = 100] : ");
11     scanf("%d", &n);
12     printf("\n Enter %d elements of the array : \n", n);
13     for (i = 0; i < n; i++)
14     {
15         scanf(" %d", &arr[i]);
16     }
17     insertionSort(arr, n);
```

```

18     do
19     {
20         printf("\n\n !! -- Operations available -- !!");
21         printf("\n 1. Display Sorted List \t 2. Search a
           particular value \t 3. Exit");
22         printf("\n Please Enter your choice : ");
23         scanf("%d", &choice);
24         switch (choice)
25         {
26         case 1:
27         {
28             printf("\n\n The sorted array is : \n");
29             for (i = 0; i < n; i++)
30             {
31                 printf(" %d \t", arr[i]);
32             }
33             break;
34         }
35         case 2:
36         {
37             printf("\n Enter the number to be searched : "
38                 );
39             scanf("%d", &x);
40             int beg = 0, end = n - 1, mid;
41             while (beg <= end)
42             {
43                 mid = (beg + end) / 2;
44                 if (arr[mid] == x)
45                 {
46                     printf("\n %d is present in the sorted
47                         array at index : %d", x, mid);
48                     flag = 1;
49                     break;
50                 }
51                 else if (arr[mid] > x)
52                 {
53                     end = mid - 1;
54                 }
55             }
56             if (flag == 0)
57                 printf("\n Element not found");
58         }
59         }
60     }
61     while (choice != 3);
62 }

```

```
53         else
54     ▾         {
55             beg = mid + 1;
56         }
57     }
58     if (beg > end || flag == 0)
59     ▾     {
60         printf("\n %d does not exist int the array"
61             , x);
62     }
63     break;
64 case 3:
65     ▾     {
66         printf("\n Program Finished !! Thank You");
67         break;
68     }
69 default:
70     ▾     {
```

```
70
71     printf("\n Please enter a valid choice 1, 2, 3\n");
72 }
73 }
74 } while (choice != 3);
75 }
76
77 void insertionSort(int arr[], int n)
78 {
79     int i, j, temp;
80     for (i = 1; i < n; i++)
81     {
82         temp = arr[i];
83         j = i - 1;
84         while ((temp < arr[j]) && (j >= 0))
85         {
86             arr[j + 1] = arr[j];
87             j--;
88         }
```

```
88     }
89     arr[j + 1] = temp;
90 }
91 }
92
```

Output

[Clear](#)

```
/tmp/RtKJN8qRVU.o
```

```
--- WELCOME TO IMPLEMENTATION OF BINARY SEARCH ---
```

```
Enter the number of elements of the array [maximum size = 100] :  
6
```

```
Enter 6 elements of the array :
```

```
12 75 9 44 89 34
```

```
!! -- Operations available -- !!
```

```
1. Display Sorted List      2. Search a particular value    3.  
Exit
```

```
Please Enter your choice : 1
```

```
The sorted array is :
```

```
9    12    34    44    75    89
```

```
!! -- Operations available -- !!
```

```
1. Display Sorted List      2. Search a particular value    3.  
Exit
```

```
Please Enter your choice : 2
```

```
Enter the number to be searched : 44
```

```
Please Enter your choice : 2
```

```
Enter the number to be searched : 44
```

```
44 is present in the sorted array at index : 3
```

```
!! -- Operations available -- !!
```

```
1. Display Sorted List      2. Search a particular value    3.  
Exit
```

```
Please Enter your choice : 2
```

```
Enter the number to be searched : 80
```

```
80 does not exist int the array
```

```
!! -- Operations available -- !!
```

```
1. Display Sorted List      2. Search a particular value    3.  
Exit
```

```
Please Enter your choice : 3
```

```
Program Finished !! Thank You
```