

Data structures laboratory

- Binary search

Code:

```
1  #include <stdio.h>
2  int main()
3  {
4      int c, first, last, middle, n, search, array[100];
5
6      printf("Enter number of elements\n");
7      scanf("%d", &n);
8
9      printf("Enter %d integers\n", n);
10
11     for (c = 0; c < n; c++)
12         scanf("%d", &array[c]);
13
14     printf("Enter value to find\n");
15     scanf("%d", &search);
16
17     first = 0;
18     last = n - 1;
19     middle = (first+last)/2;
20
21     while (first <= last) {
22         if (array[middle] < search)
23             first = middle + 1;
24         else if (array[middle] == search) {
25             printf("%d found at location %d.\n", search, middle+1);
26             break;
27         }
28         else
29             last = middle - 1;
30
31         middle = (first + last)/2;
32     }
33     if (first > last)
34
35         printf("Not found! %d isn't present in the list.\n", search);
36
37     return 0;
38 }
```

Output:

Enter number of elements

4

Enter 4 integers

12

11

45

67

Enter value to find

45

45 found at location 3.

Process exited after 11.64 seconds with return value 0

Press any key to continue . . .

Data structures laboratory

1. Linear search

Code-

```
1  #include <stdio.h>
2  int main()
3  {
4      int array[100], search, c, n;
5
6      printf("Enter number of elements in array\n");
7      scanf("%d", &n);
8
9      printf("Enter %d integer(s)\n", n);
10
11     for (c = 0; c < n; c++)
12         scanf("%d", &array[c]);
13
14     printf("Enter a number to search\n");
15     scanf("%d", &search);
16
17     for (c = 0; c < n; c++)
18     {
19         if (array[c] == search)    /* If required element is found */
20         {
21             printf("%d is present at location %d.\n", search, c+1);
22             break;
23         }
24     }
25     if (c == n)
26         printf("%d isn't present in the array.\n", search);
27
28     return 0;
29 }
```

Output:

Enter number of elements in array

4

Enter 4 integer(s)

1

3

5

7

Enter a number to search

4

4 isn't present in the array.

Process exited after 10.31 seconds with return value 0

Press any key to continue . . . _

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int n, array[1000], c, d, t, flag = 0;
6
7      printf("Enter number of elements\n");
8      scanf("%d", &n);
9
10     printf("Enter %d integers\n", n);
11
12     for (c = 0; c < n; c++)
13         scanf("%d", &array[c]);
14
15     for (c = 1; c <= n - 1; c++) {
16         t = array[c];
17
18         for (d = c - 1; d >= 0; d--) {
19             if (array[d] > t) {
20                 array[d+1] = array[d];
21                 flag = 1;
22             }
23             else
24                 break;
25         }
26         if (flag)
27             array[d+1] = t;
28     }
29
30     printf("Sorted list in ascending order:\n");
31
32     for (c = 0; c <= n - 1; c++) {
33         printf("%d\n", array[c]);
34     }
35 }

```



```

int n, array[1000], c, d, t, flag = 0;

printf("Enter number of elements\n");
scanf("%d", &n);

printf("Enter %d integers\n", n);

for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

for (c = 1; c <= n - 1; c++) {
    t = array[c];

    for (d = c - 1; d >= 0; d--) {
        if (array[d] > t) {
            array[d+1] = array[d];
            flag = 1;
        }
        else
            break;
    }
    if (flag)
        array[d+1] = t;
}

printf("Sorted list in ascending order:\n");

for (c = 0; c <= n - 1; c++) {
    printf("%d\n", array[c]);
}

return 0;

```

Enter number of elements

4

Enter 4 integers

17

45

4

90

Sorted list in ascending order:

4

17

45

90

Process exited after 9.17 seconds with return value 0

Press any key to continue . . . ■

```

#include <stdio.h>
int main()
{
    int array[100], n, c, d, position, t;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    for (c = 0; c < (n - 1); c++)
    {
        position = c;

        for (d = c + 1; d < n; d++)
        {
            if (array[position] > array[d])
                position = d;
        }
        if (position != c)
        {
            t = array[c];
            array[c] = array[position];
            array[position] = t;
        }
    }

    printf("Sorted list in ascending order:\n");

    for (c = 0; c < n; c++)

```



```
printf("Enter number of elements\n");  
scanf("%d", &n);
```

```
printf("Enter %d integers\n", n);
```

```
for (c = 0; c < n; c++)  
    scanf("%d", &array[c]);
```

```
for (c = 0; c < (n - 1); c++)  
{  
    position = c;  
  
    for (d = c + 1; d < n; d++)  
    {  
        if (array[position] > array[d])  
            position = d;  
    }  
    if (position != c)  
    {  
        t = array[c];  
        array[c] = array[position];  
        array[position] = t;  
    }  
}
```

```
printf("Sorted list in ascending order:\n");
```

```
for (c = 0; c < n; c++)  
    printf("%d\n", array[c]);
```

```
return 0;
```

Enter number of elements

4

Enter 4 integers

56

8

9

76

Sorted list in ascending order:

8

9

56

76

Process exited after 7.398 seconds with return value 0

Press any key to continue . . . █