



The image shows a screenshot of a C code editor window. The editor has a dark theme and a toolbar at the top with buttons for Run, Debug, Stop, Share, Save, and Beautify. The code is written in C and implements a search program. It includes a header file <stdio.h> and defines three functions: binarysearch, sort, and linear. The main function prompts the user to enter the number of elements, the array elements, and the elements to be searched. It then asks the user to choose between linear search, binary search, or exit. A while loop handles the user's choice, and a switch statement implements the selected search algorithm. The linear search function is partially visible at the bottom of the code.

```
9 #include <stdio.h>
10 int binarysearch(int a[], int, int, int);
11 void sort(int [], int);
12 int linear(int a[], int, int);
13 int main()
14 {
15     int choice;
16     int n, i, a[10], key, pos = 0;
17     printf("enter the no. of elements\n");
18     scanf("%d",&n);
19     printf("enter the array elements\n");
20     for(i=0;i<n;i++)
21     {
22         scanf("%d",&a[i]);
23     }
24     printf("enter the elements to be searched\n");
25     scanf("%d",&key);
26     printf("\nChoose one:\n1.linear search\n2.binary search\n3.Exit\n");
27     printf("Your choice:");
28     scanf("%d",&choice);
29     while(choice!=3){
30         switch(choice){
31             case 1:pos = linear(a, n, key);
32                 if(pos>0)
33                 {
34                     printf("element %d is found at %dth position",key, pos);
35                 }
36                 else
37                 {
```

```
main.c
37 {
38     printf("element not found\n");
39 }
40 break;
41 case 2: sort(a, n);
42     pos = binarysearch(a, 0, n-1, key);
43     if (pos < 0 )
44         printf("Cannot find the element %d in the array.\n", key);
45
46     else
47         printf("The position of %d in the array is %d.\n", key, pos+1);
48     break;
49 default: printf("Invalid choice!");
50 }
51 printf("\nChoose one:\n1.linear search\n2.binary search\n3.Exit\n");
52 printf("Your choice:");
53 scanf("%d",&choice);
54 }
55 return 0;
56
57 }
58 void sort(int list[], int size)
59 {
60     int temp, i, j;
61     for (i = 0; i < size; i++)
62     {
63         for (j = i; j < size; j++)
64         {
65             if (list[i] > list[j])
66                 swap(&list[i], &list[j]);
67         }
68     }
69 }
```



```
main.c
63     for (j = i; j < size; j++)
64     {
65         if (list[i] > list[j])
66         {
67             temp = list[i];
68             list[i] = list[j];
69             list[j] = temp;
70         }
71     }
72 }
73 printf("Sorted Array:\n");
74 for(i=0;i<size;i++)
75     printf("%d\n",list[i]);
76 }
77
78 int binarysearch(int a[], int low, int high, int x) {
79     int mid = (low + high) / 2;
80     if (low > high) return -1;
81     if (a[mid] == x) return mid;
82
83     if (a[mid] < x)
84         return binarysearch(a, mid + 1, high, x);
85     else
86         return binarysearch(a, low, mid-1, x);
87 }
88 int linear(int a[], int n, int key)
89 {
90     if(n>=0)
91     {
```

```
main.c
75     printf("%d\n", list[1]);
76
77 }
78 int binarysearch(int a[], int low, int high, int x) {
79     int mid = (low + high) / 2;
80     if (low > high) return -1;
81     if (a[mid] == x) return mid;
82
83     if (a[mid] < x)
84         return binarysearch(a, mid + 1, high, x);
85     else
86         return binarysearch(a, low, mid - 1, x);
87 }
88 int linear(int a[], int n, int key)
89 {
90     if(n >= 0)
91     {
92         if(a[n-1] == key)
93         {
94             return n;
95         }
96         else
97         {
98             return linear(a, n-1, key);
99             n--;
100         }
101     }
102 }
103
```

input

```
enter the no. of elements
5
enter the array elements
20
10
50
30
40
enter the elements to be searched
40

Choose one:
1.linear search
2.binary search
3.Exit
Your choice:1
element 40 is found at 5th position
Choose one:
1.linear search
2.binary search
3.Exit
Your choice:2
Sorted Array:
10
20
30
40
50
```



```
input
enter the array elements
20
10
50
30
40
enter the elements to be searched
40

Choose one:
1.linear search
2.binary search
3.Exit
Your choice:1
element 40 is found at 5th position
Choose one:
1.linear search
2.binary search
3.Exit
Your choice:2
Sorted Array:
10
20
30
40
50
The position of 40 in the array is 4.
```