

main.c

```
1 #include <stdio.h>
2
3 void binary_search(int [], int, int, int);
4 void bubble_sort(int [], int);
5
6 int main()
7 {
8     int key, size, i;
9     int list[25];
10
11     printf("Enter size of a list: ");
12     scanf("%d", &size);
13     printf("Enter elements\n");
14     for(i = 0; i < size; i++)
15     {
16         scanf("%d", &list[i]);
17     }
18     bubble_sort(list, size);
19     printf("\n");
20     printf("Enter key to search: ");
21     scanf("%d", &key);
22     binary_search(list, 0, size, key);
23
24 }
25
26 void bubble_sort(int list[], int size)
27 {
28     int temp, i, j;
29     for (i = 0; i < size; i++)
```

main.c

```
27  {
28      int temp, i, j;
29      for (i = 0; i < size; i++)
30      {
31          for (j = i; j < size; j++)
32          {
33              if (list[i] > list[j])
34              {
35                  temp = list[i];
36                  list[i] = list[j];
37                  list[j] = temp;
38              }
39          }
40      }
41  }
42
43 void binary_search(int list[], int lo, int hi, int key)
44 {
45     int mid;
46
47     if (lo > hi)
48     {
49         printf("Key not found . . .");
50         return;
51     }
52     mid = (lo + hi) / 2;
53     if (list[mid] == key)
54     {
55         printf("Key found . . .");
56     }
57 }
```

main.c

```
..  
46  
47     if (lo > hi)  
48     {  
49         printf("Key not found\n");  
50         return;  
51     }  
52     mid = (lo + hi) / 2;  
53     if (list[mid] == key)  
54     {  
55         printf("Key found\n");  
56     }  
57     else if (list[mid] > key)  
58     {  
59         binary_search(list, lo, mid - 1, key);  
60     }  
61     else if (list[mid] < key)  
62     {  
63         binary_search(list, mid + 1, hi, key);  
64     }  
65 }  
66
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
Enter size of a list: 6
Enter elements
6
1
2
3
5
6

Enter key to search
3
Key found
>
```

Binary Search

#include <stdio.h>

Void binary - Search (int [], int, int, int);

Void bubble - sort (int [], int);

int main()

{

int Key, size; i;

int list [25];

printf ("Enter size of a list : ");

scanf ("%d", &size);

printf ("Enter elements \n");

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

{

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

}

bubble - sort (list, size);

printf ("\n");

printf ("Enter key to search in ");

scanf ("%d", &key);

binary - search (list, 0, size, key);

}

Void bubble - sort (int list[], int size)

{

int temp, i, j;

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

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

if (list[i] > list[j])

}

temp = list[i];

list[i] = list[j];

list[j] = temp;

}

}

}

}

void binarySearch(int lis[], int lo, int hi,
int key)

}

int mid;

if (lo > hi)

{ printf("Key not found\n");
return;

}

mid = (lo + hi) / 2;

if (list[mid] == key)

{

printf("Key found\n");

if (lo > hi)

{ printf("Key not found\n");
return;

}

mid = (lo + hi) / 2;

if (list[mid] == key)

{

printf("Key found\n");

{

else if (list[mid] > key)
{

 binary - Search (list, lo, mid - 1, key);

else if (list[mid] < key)
{

 binary - Search (list, mid + 1, hi, key);

{

 return -1;

```
#include <stdio.h>
int RecursiveLS(int arr[], int value, int index, int n)
{
    int pos = 0;

    if(index >= n)
    {
        return 0;
    }

    else if (arr[index] == value)
    {
        pos = index + 1;
        return pos;
    }

    else
    {
        return RecursiveLS(arr, value, index+1, n);
    }
    return pos;
}

int main()
{
    int n, value, pos, m = 0, arr[100];
    printf("Enter the total elements in the array: ");
    scanf("%d", &n);
```

```
24 int main()
25 {
26     int n, value, pos, m = 0, arr[100];
27     printf("Enter the total elements in the array: ");
28     scanf("%d", &n);
29
30     printf("Enter the array elements: \n");
31     for (int i = 0; i < n; i++)
32     {
33         scanf("%d", &arr[i]);
34     }
35
36     printf("Enter the element to search: ");
37     scanf("%d", &value);
38
39     pos = RecursiveLS(arr, value, 0, n); [I]
40     if (pos != 0)
41     {
42         printf("Element found at pos %d\n", pos);
43     }
44     else
45     {
46         printf("Element not found\n");
47     }
48     return 0;
49 }
```

```
* clang-7 -pthread -lm -o main main.c
```

```
* ./main
```

```
Enter the total elements in the array: 5
```

```
Enter the array elements:
```

```
1
```

```
2
```

```
3
```

```
4
```

```
6
```

```
Enter the element to search: 4
```

```
Element found at pos 4
```

```
*
```

Linear Search

```
# include <stdio.h>
int RecursiveLS( int arr[], int m, int l, int r, int n )
```

{

```
int pos = 0;
```

```
if ( index >= n )
    }
```

return -1;

}

```
else if ( arr[index] == value )
    }
```

```
pos = index + 1,
```

return pos;

}

else

{

return RecursiveLS(arr, value, index + 1, n);

}

return pos;

}

```
int main()
```

{

```
int a[100], value, pos, m = 0, arr[100];
```

```
printf("Enter the total element in the array: ");
```

```
scanf("%d", &n);
```

```
printf("Enter the array elements: \n");
```

```
for ( int i = 0; i < n; i++ )
```

{

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

}

print ("Enter the elements to Search : ");
Scanning C (" / d ", & value);

pos = Reversesearch (arr, value, 0, n),
if (pos != 0)
 {

 print ("Element found at pos : %d \n", pos);

}

else

{

 print ("Element not found in ");

 return 0;

}