

main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  clock_t start, end;
6  double cpu_time;
7
8  void swap(int *xp, int *yp)
9  {
10     int temp = *xp;
11     *xp = *yp;
12     *yp = temp;
13 }
14
15 void selectionSort(int arr[], int n)
16 {
17     int i, j, min_idx;
18
19     for (i = 0; i < n-1; i++)
20     {
21         min_idx = i;
22         for (j = i+1; j < n; j++)
23             if (arr[j] < arr[min_idx])
24                 min_idx = j;
25
26         swap(&arr[min_idx], &arr[i]);
27     }
28 }
29
30 int linear_search(int arr[],int key,int i,int n)
31 {
32     if(arr[i]==key)
33         return i+1;
34     else if(i>=n)
35         return -1;
36     else
37         return linear_search(arr,key,i+1,n);
38 }
```

main.c

```
37 return linear_search(arr,key);  
38 }  
39  
40 void binary_search(int arr[],int l,int r,int key)  
41 {  
42     if(l>r)  
43     {  
44         printf("Element not present");  
45         return;  
46     }  
47     int mid=l + (r - l) / 2;  
48     if(arr[mid]==key)  
49         printf("Element found at index:%d \n",mid +1);  
50     else if(key<arr[mid])  
51         return binary_search(arr,l,mid-1,key);  
52     else if(key>arr[mid])  
53         return binary_search(arr,mid+1,r,key);  
54 }  
55  
56 int main()  
57 {  
58     int r, k, n, ch, flag, key, l, mid, arr[10000];  
59     printf("Enter the number of elements in array \n");  
60     scanf("%d", &n);  
61  
62     for(k=0;k<n;k++)  
63         arr[k]=rand()%n+1;  
64  
65     printf("Elements of the array are:\n");  
66  
67     for(k=0;k<n;k++){  
68         printf("%d ",arr[k]);  
69     }  
70     printf("\n");  
71  
72     printf("RECURSIVE SEARCH functions:\n");  
73     printf("1.LINEAR SEARCH:\n");  
74     printf("2.BINARY SEARCH:\n");  
75     printf("Enter your choice:\n");
```

main.c

```
70 printf("\n");
71
72 printf("RECURSIVE SEARCH functions:\n");
73 printf("1.LINEAR SEARCH:\n");
74 printf("2.BINARY SEARCH:\n");
75 printf("Enter your choice:");
76 scanf("%d",&ch);
77
78 switch(ch)
79 {
80 case 1:
81     printf("Enter the element to be searched:");
82     scanf("%d",&key);
83
84     start = clock();
85     flag=linear_search(arr,key,0,n);
86     for (int c = 1; c <= 5000; c++)
87         for (int d = 1; d <= 5000; d++) { }
88     end = clock();
89     cpu_time = (double)(end-start)/CLOCKS_PER_SEC;
90
91     if(flag== -1)
92         printf("Element not found \n");
93     else
94         printf("Element found at position:%d \n",flag);
95
96     printf("Execution time for linear_search = %f seconds\n", cpu_time);
97     break;
98 case 2:
99     printf("Enter the element to be searched:");
100     scanf("%d",&key);
101
102     start = clock();
103     // sorting array for binary search
104     selectionSort(arr, n);
105     printf("Elements of the sorted array are:\n");
106     for(k=0;k<n;k++){
107         printf("%d ",arr[k]);
```

main.c

```
84 start = clock();
85 flag=linear_search(arr,key,0,n);
86 for (int c = 1; c <= 5000; c++)
87     for (int d = 1; d <= 5000; d++) { }
88 end = clock();
89 cpu_time = (double)(end-start)/CLOCKS_PER_SEC;
90
91 if(flag==-1)
92     printf("Element not found \n");
93 else
94     printf("Element found at position:%d \n",flag);
95
96 printf("Execution time for linear_search = %f seconds\n", cpu_time);
97 break;
98 case 2:
99     printf("Enter the element to be searched:");
100     scanf("%d",&key);
101
102 start = clock();
103 // sorting array for binary search
104 selectionSort(arr, n);
105 printf("Elements of the sorted array are:\n");
106 for(k=0;k<n;k++){
107     printf("%d ",arr[k]);
108 }
109 printf("\n");
110 binary_search(arr,0,n,key);
111
112 for (int c = 1; c <= 5000; c++)
113     for (int d = 1; d <= 5000; d++) { }
114 end = clock();
115 cpu_time = (double)(end-start)/CLOCKS_PER_SEC;
116 printf("Execution time for binary_search = %f seconds\n", cpu_time);
117
118 break;
119 }
120 return 0;
121 }
```

input

Enter the number of elements in array

50

Elements of the array are:

34 37 28 16 44 36 37 43 50 22 13 28 41 10 14 27 41 27 23 37 12 19 18 30 33 31 13 24 18 36 30 3 23 9 20 18 44 7 12 43 30 24 22 20 35 38 49 25 16 21

RECURSIVE SEARCH functions:

1.LINEAR SEARCH:

2.BINARY SEARCH:

Enter your choice:1

Enter the element to be searched:43

Element found at position:8

Execution time for linear_search = 0.055775 seconds

...Program finished with exit code 0

Press ENTER to exit console.

input

```
Enter the number of elements in array
50
Elements of the array are:
34 37 28 16 44 36 37 43 50 22 13 28 41 10 14 27 41 27 23 37 12 19 18 30 33 31 13 24 18 36 30 3 23 9 20 18 44 7 12 43 30 24 22 20 35 38 49 25 16 21
RECURSIVE SEARCH functions:
1.LINEAR SEARCH:
2.BINARY SEARCH:
Enter your choice:2
Enter the element to be searched:37
Elements of the sorted array are:
3 7 9 10 12 12 13 13 14 16 16 18 18 18 19 20 20 21 22 22 23 23 24 24 25 27 27 28 28 30 30 30 31 33 34 35 36 36 37 37 37 38 41 41 43 43 44 44 49 50
Element found at index:39
Execution time for binary_search = 0.056555 seconds
```

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