

C linear_binary.c ×

C linear_binary.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  clock_t start, end;
5  double cpu_time;
6  int linear_search(int arr[], int high, int low, int key)
7  {
8      if (low < high)
9          return -1;
10     if (arr[high] == key)
11         return high;
12     if (arr[low] == key)
13         return low;
14     return linear_search(arr, high+1, low-1, key);
15 }
16 int binary_search(int arr[], int high, int low, int key)
17 {
18     if (low >= high)
19     {
20         int mid = (high+low)/2;
21         if (arr[mid] == key)
22         {
23             return mid;
24         }
25         if (arr[mid] > key)
26         {
27             return binary_search(arr, high, mid-1, key);
28         }
29         return binary_search(arr, mid + 1, low, key);
30     }
31     return -1;
32 }
33 int main()
34 {
35
36     int k, pos, c, d, i, n, temp, choice, key, j, flag = 1, arr[10000];
37     srand(time(0));
38     while (flag == 1)
```

C linear_binary.c X

C linear_binary.c

```
37 srand(time(0));
38 while (flag==1)
39 {
40     printf("1:Linear_Search\n2:Binary_Search\n3:Exit\n");
41     printf("Enter your choice\n");
42     scanf("%d", &choice);
43     switch(choice)
44     {
45         case 1:
46             printf("Enter the number of elements:\n");
47             scanf("%d", &n);
48             for (k = 1; k <= n; k++)
49             {
50                 arr[k]=rand()%100;
51                 printf("%d ",arr[k]);
52             }
53             printf("\nEnter the Element to be Searched : \n");
54             scanf("%d", &key);
55             start = clock();
56             pos = linear_search(arr, 0, n-1, key);
57             for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }
58             end = clock();
59             cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
60             if(pos == -1)
61             {
62                 printf("Element is not present in the Array\n");
63             }
64             else
65             {
66                 printf("Element is present at the Position %d\n", pos);
67             }
68             printf("Execution time for linear_search = %f ms\n", cpu_time*1000);
69             break;
70         case 2:
71             printf("Enter the number of elements:");
72             scanf("%d", &n);
73             for (int k =1; k<=n; k++)
```

C linear_binary.c X

C linear_binary.c

```
73     for (int k =1; k<=n; k++)
74     {
75         arr[k]=rand()%100;
76     }
77     for (i=1; i <=n;i++)
78     {
79         for (j = i + 1; j <= n; ++j)
80         {
81             if (arr[i] >arr[j])
82             {
83                 temp =arr[i];
84                 arr[i] = arr[j];
85                 arr[j] = temp;
86             }
87         }
88     }
89     for (int k =1; k <=n; k++)
90     {
91         printf("%d ",arr[k]);
92     }
93     printf("\nEnter the element to be Searched :\n");
94     scanf("%d", &flag);
95     start = clock();
96     for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }
97     pos = binary_search(arr, 0, n - 1, flag);
98     end = clock();
99     cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
100     if(pos == -1)
101     {
102         printf("Element is not present in array\n");
103     }
104     else
105     {
106         printf("Element is present at the Position %d\n", pos);
107     }
108     printf("Execution time for binary_search = %f ms\n", cpu_time*1000);
109     break;
```

C linear_binary.c X

C linear_binary.c

```
82         {
83             temp = arr[i];
84             arr[i] = arr[j];
85             arr[j] = temp;
86         }
87     }
88 }
89 for (int k = 1; k <= n; k++)
90 {
91     printf("%d ", arr[k]);
92 }
93 printf("\nEnter the element to be Searched :\n");
94 scanf("%d", &flag);
95 start = clock();
96 for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }
97 pos = binary_search(arr, 0, n - 1, flag);
98 end = clock();
99 cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
100 if(pos == -1)
101 {
102     printf("Element is not present in array\n");
103 }
104 else
105 {
106     printf("Element is present at the Position %d\n", pos);
107 }
108 printf("Execution time for binary_search = %f ms\n", cpu_time*1000);
109 break;
110 default: flag=0;
111 }
112 }
113 return 0;
114 }
```

Enter the number of elements:

400

89 55 54 32 35 78 33 99 5 10 28 80 17 27 35 56 11 74 94 91 68 70 4 87 11 23 67 68 66 90 47 2 10 86 56 17 15 61 32 8 10 10 12 46 3 82 10 38 13 20 11 22 16 46 82 18 11 80 24 60 81 31 13 63 88 34 83 40 0
68 93 68 26 34 19 83 93 14 37 88 33 41 53 74 53 85 69 35 80 66 20 51 48 85 55 3 83 73 82 15 35 98 9 24 27 79 58 54 87 8 16 65 88 12 84 15 96 15 66 61 64 64 30 5 62 17 29 40 98 82 3 31 84 79 86 83 36 16
71 61 12 11 31 15 1 85 65 48 92 29 21 10 19 2 32 58 44 22 43 44 78 90 21 44 38 67 81 0 39 93 31 70 17 0 10 0 38 97 58 82 42 82 67 80 40 20 97 78 42 56 85 8 6 0 48 45 41 15 99 14 73 75 47 90 9 46 69 73
67 87 25 94 69 54 82 25 64 35 65 8 55 71 95 76 43 96 28 32 34 66 94 90 73 33 21 61 85 45 4 38 69 47 88 65 94 22 96 47 99 10 8 94 11 80 25 80 52 38 1 35 9 96 72 76 56 19 27 31 68 25 92 86 32 42 9 46 53
25 51 38 28 18 62 78 28 90 12 35 42 99 43 59 21 74 15 4 41 62 32 79 71 43 59 27 79 51 42 24 13 27 77 44 43 9 64 81 43 37 74 47 24 30 27 40 43 27 61 65 89 5 39 27 61 46 82 91 47 62 45 54 26 25 80 64 78
18 80 54 59 18 77 70 55 60 23 54 11 83 57 64 3 64 18 61 68 78 54 63 75 80 73 65 6 38 22 13 34 14 79 23 61 24 71 17 95 59 72 12 48 27 38 66 92 40 32 6 5 48 81 11

Enter the Element to be Searched :

18

Element is present at the Position 363

Execution time for linear_search = 56.000000 ms

1:Linear_Search

2:Binary_Search

3:Exit

Enter your choice

2

Enter the number of elements:200

0 0 1 2 2 2 2 3 4 6 6 6 7 7 7 8 8 9 9 9 10 11 11 11 11 13 13 13 14 15 15 15 17 17 17 18 18 18 20 20 21 21 21 22 22 23 23 24 24 24 24 24 25 25 27 28 28 29 29 29 29 30 31 31 31 32 32 32 33 33 33 34 34 34
34 36 37 37 38 38 38 38 38 39 39 40 40 40 41 41 42 42 43 43 44 44 44 44 44 44 45 45 45 46 46 47 47 49 49 49 49 50 50 51 51 51 52 52 52 53 54 54 54 55 55 57 59 59 59 59 59 60 60 61 62 62 62 62 63 64 64
64 65 65 65 66 66 66 66 67 67 68 68 69 69 70 71 72 72 72 72 72 74 75 76 78 78 80 80 80 80 80 80 81 82 84 84 85 85 87 87 88 88 88 88 89 90 90 92 93 94 94 94 95 95 96 97 97 97

Enter the element to be Searched :

10

Element is present at the Position 21

Execution time for binary_search = 78.000000 ms

PS C:\Users\muska\OneDrive\Desktop\C programs> █