

main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int linearSearch(const int arr[], int size, int k) {
5      for (int i = 0; i < size; ++i) {
6          if (k == arr[i]) {
7              return i;
8          }
9      }
10     return -1;
11 }
12
13 int main() {
14     const int arr[] = {1, 2, 3, 4, 5};
15     const int arraySize = sizeof(arr) / sizeof(arr[0]);
16     int k;
17
18     cout << "Enter an integer: " << endl;
19     cin >> k;
20
21     int result = linearSearch(arr, arraySize, k);
22     if (result >= 0) {
23         cout << "The number " << arr[result] << " was found at index "
24             << result << " of the array." << endl;
25     } else {
26         cout << "The number " << k << " was not found." << endl;
27     }
28
29     return 0;
30 }
31
```

/tmp/yNYtMLRHjS.o

Enter an integer:

2

The number 2 was found at index 1 of the array.

=== Code Execution Successful ===



main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int linearSearch(const int arr[], int size, int k) {
5      for (int i = 0; i < size; ++i) {
6          if (k == arr[i]) {
7              return i;
8          }
9      }
10     return -1;
11 }
12
13 int main() {
14     const int arr[] = {10, 20, 30};
15     const int arraySize = sizeof(arr) / sizeof(arr[0]);
16     int k;
17
18     cout << "Enter an integer: " << endl;
19     cin >> k;
20
21     int result = linearSearch(arr, arraySize, k);
22     if (result >= 0) {
23         cout << "The number " << arr[result] << " was found at index "
24             << result << " of the array." << endl;
25     } else {
26         cout << "The number " << k << " was not found." << endl;
27     }
28
29     return 0;
30 }
31
```

/tmp/NFTAlzH0XC.o

Enter an integer:

30

The number 30 was found at index 2 of the array.

=== Code Execution Successful ===



main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int linearSearch(const int arr[], int size, int k) {
5      for (int i = 0; i < size; ++i) {
6          if (k == arr[i]) {
7              return i;
8          }
9      }
10     return -1;
11 }
12
13 int main() {
14     const int arr[] = {};
15     const int arraySize = sizeof(arr) / sizeof(arr[0]);
16     int k;
17
18     cout << "Enter an integer: " << endl;
19     cin >> k;
20
21     int result = linearSearch(arr, arraySize, k);
22     if (result >= 0) {
23         cout << "The number " << arr[result] << " was found at index "
24         << result << " of the array." << endl;
25     } else {
26         cout << "Since array is empty the value " << k << " cannot be found." <<
            endl;
27     }
28
29     return 0;
30 }
31
```

/tmp/Vr0F9pp9am.o

Enter an integer:

10

Since array is empty the value 10 cannot be found.

=== Code Execution Successful ===



main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int linearSearch(const int arr[], int size, int k) {
5      for (int i = 0; i < size; ++i) {
6          if (k == arr[i]) {
7              return i;
8          }
9      }
10     return -1;
11 }
12
13 int main() {
14     const int arr[] = {12, 13, 14, 15};
15     const int arraySize = sizeof(arr) / sizeof(arr[0]);
16     int k;
17
18     cout << "Enter an integer: " << endl;
19     cin >> k;
20
21     int result = linearSearch(arr, arraySize, k);
22     if (result >= 0) {
23         cout << "The number " << arr[result] << " was found at index "
24             << result << " of the array." << endl;
25     } else {
26         cout << "The given value " << k << " does not exist in the given array
                hence cant be found." << endl;
27     }
28
29     return 0;
30 }
31
```

/tmp/l5UwDhXVIk.o

Enter an integer:

18

The given value 18 does not exist in the given array hence cant be found.

=== Code Execution Successful ===



main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int linearSearch(const int arr[], int size, int k) {
5      for (int i = 0; i < size; ++i) {
6          if (k == arr[i]) {
7              return i;
8          }
9      }
10     return -1;
11 }
12
13 int main() {
14     const int arr[] = {12, 13, 14, 15};
15     const int arraySize = sizeof(arr) / sizeof(arr[0]);
16     int k;
17
18     cout << "Enter an integer: " << endl;
19     cin >> k;
20
21     int result = linearSearch(arr, arraySize, k);
22     if (result >= 0) {
23         cout << "The number " << arr[result] << " was found at index "
24             << result << " of the array." << endl;
25     } else {
26         cout << "You have inputted a string and array is of integer hence
27             invalid data-type has been submitted" << endl;
28     }
29     return 0;
30 }
31
```

/tmp/nYz1xHr2hb.o

Enter an integer:

mihir

You have inputted a string and array is of integer hence invalid data-type has been submitted

=== Code Execution Successful ===



main.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  int binarySearch(const int arr[], int low, int high, int k) {
5      if (low > high) {
6          return -1;
7      }
8
9      int mid = low + (high - low) / 2;
10
11     if (arr[mid] == k) {
12         return mid;
13     }
14
15     if (k < arr[mid]) {
16         return binarySearch(arr, low, mid - 1, k);
17     } else {
18         return binarySearch(arr, mid + 1, high, k);
19     }
20 }
21
22 int main() {
23     const int arr[] = {1, 2, 3, 4, 5};
24     const int arraySize = sizeof(arr) / sizeof(arr[0]);
25     int k;
26
27     cout << "Enter an integer to search for: " << endl;
28     cin >> k;
29
30     int result = binarySearch(arr, 0, arraySize - 1, k);
31
32     if (result >= 0) {
33         cout << "The number " << k << " was found at index " << result << " of the array." << endl;
34     } else {
35         cout << "The number " << k << " was not found in the array." << endl;
36     }
37
38     return 0;
39 }
40
41
42
```

Output

Clear

/tmp/URtTLaoNzh.o  
Enter an integer to search for:  
2  
The number 2 was found at index 1 of the array.  
  
=== Code Execution Successful ===





```
1  #include <iostream>
2  using namespace std;
3
4  int binarySearch(const int arr[], int low, int high, int k) {
5      if (low > high) {
6          return -1;
7      }
8
9      int mid = low + (high - low) / 2;
10
11     if (arr[mid] == k) {
12         return mid;
13     }
14
15     if (k < arr[mid]) {
16         return binarySearch(arr, low, mid - 1, k);
17     } else {
18         return binarySearch(arr, mid + 1, high, k);
19     }
20 }
21
22 int main() {
23     const int arr[] = {10, 20, 30};
24     const int arraySize = sizeof(arr) / sizeof(arr[0]);
25     int k;
26
27     cout << "Enter an integer to search for: " << endl;
28     cin >> k;
29
30     int result = binarySearch(arr, 0, arraySize - 1, k);
31
32     if (result >= 0) {
33         cout << "The number " << k << " was found at index " << result << " of the array." << endl;
34     } else {
35         cout << "The number " << k << " was not found in the array." << endl;
36     }
37
38     return 0;
39 }
40
41
42
```

```
/tmp/EtVG9auzaU.o
Enter an integer to search for:
30
The number 30 was found at index 2 of the array.

=== Code Execution Successful ===
```



main.cpp



Share

Run

Output

Clear

```
1  #include <iostream>
2  using namespace std;
3
4  int binarySearch(const int arr[], int low, int high, int k) {
5      if (low > high) {
6          return -1;
7      }
8
9      int mid = low + (high - low) / 2;
10
11     if (arr[mid] == k) {
12         return mid;
13     }
14
15     if (k < arr[mid]) {
16         return binarySearch(arr, low, mid - 1, k);
17     } else {
18         return binarySearch(arr, mid + 1, high, k);
19     }
20 }
21
22 int main() {
23     const int arr[] = {};
24     const int arraySize = sizeof(arr) / sizeof(arr[0]);
25     int k;
26
27     cout << "Enter an integer to search for: " << endl;
28     cin >> k;
29
30     int result = binarySearch(arr, 0, arraySize - 1, k);
31
32     if (result >= 0) {
33         cout << "The number " << k << " was found at index " << result << " of the array." << endl;
34     } else {
35         cout << "Since array is empty the value " << k << " cannot be found." << endl;
36     }
37
38     return 0;
39 }
40
41
42
```

/tmp/ixnyM7pEQX.o  
Enter an integer to search for:  
10  
Since array is empty the value 10 cannot be found.

=== Code Execution Successful ===





```
1  #include <iostream>
2  using namespace std;
3
4  int binarySearch(const int arr[], int low, int high, int k) {
5      if (low > high) {
6          return -1;
7      }
8
9      int mid = low + (high - low) / 2;
10
11     if (arr[mid] == k) {
12         return mid;
13     }
14
15     if (k < arr[mid]) {
16         return binarySearch(arr, low, mid - 1, k);
17     } else {
18         return binarySearch(arr, mid + 1, high, k);
19     }
20 }
21
22 int main() {
23     const int arr[] = {12, 13, 14, 15};
24     const int arraySize = sizeof(arr) / sizeof(arr[0]);
25     int k;
26
27     cout << "Enter an integer to search for: " << endl;
28     cin >> k;
29
30     int result = binarySearch(arr, 0, arraySize - 1, k);
31
32     if (result >= 0) {
33         cout << "The number " << k << " was found at index " << result << " of the array." << endl;
34     } else {
35         cout << "The given value " << k << " does not exist in the given array hence cant be found." << endl;
36     }
37
38     return 0;
39 }
40
41
42
```

```
/tmp/gjfS1IrRJt.o
Enter an integer to search for:
18
The given value 18 does not exist in the given array hence cant be found.

=== Code Execution Successful ===
```





```
1  #include <iostream>
2  using namespace std;
3
4  bool isSorted(const int arr[], int size) {
5      for (int i = 1; i < size; ++i) {
6          if (arr[i] < arr[i - 1]) {
7              return false;
8          }
9      }
10     return true;
11 }
12
13 int binarySearch(const int arr[], int low, int high, int k) {
14     if (low > high) {
15         return -1;
16     }
17
18     int mid = low + (high - low) / 2;
19
20     if (arr[mid] == k) {
21         return mid;
22     }
23
24     if (k < arr[mid]) {
25         return binarySearch(arr, low, mid - 1, k);
26     } else {
27         return binarySearch(arr, mid + 1, high, k);
28     }
29 }
30
31 int main() {
32     const int arr[] = {10, 5, 2, 8};
33     const int arraySize = sizeof(arr) / sizeof(arr[0]);
34     int k;
35
36     cout << "Enter an integer to search for: " << endl;
37     cin >> k;
38
39     if (isSorted(arr, arraySize)) {
40         int result = binarySearch(arr, 0, arraySize - 1, k);
41
42         if (result >= 0) {
43             cout << "The number " << k << " was found at index " << result << " of the array." << endl;
44         } else {
45             cout << "The given value " << k << " does not exist in the given array." << endl;
46         }
47     } else {
48         cout << "The array is unsorted. Binary search requires a sorted array." << endl;
49     }
50
51     return 0;
52 }
53
54
55
56
```

/tmp/kJJfLrH7Q1.o  
Enter an integer to search for:  
2  
The array is unsorted. Binary search requires a sorted array.

=== Code Execution Successful ===