main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int linearSearch(const int arr[], int size, int k) {
    for (int i = 0; i < size; ++i) {
        if (k == arr[i]) {
            return i;
        }
    }
    return -1;
}

int main() {
    const int arr[] = {1, 2, 3, 4, 5};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer: " << endl;
    cin >> k;

    int result = linearSearch(arr, arraySize, k);
    if (result >= 0) {
        cout << "The number " << arr[result] << " was found at index "
             << result << " of the array." << endl;
    } else {
        cout << "The number " << k << " was not found." << endl;
    }

    return 0;
}
```

Output

Clear

```
/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

```cpp
#include <iostream>
using namespace std;

int linearSearch(const int arr[], int size, int k) {
    for (int i = 0; i < size; ++i) {
        if (k == arr[i]) {
            return i;
        }
    }
    return -1;
}

int main() {
    const int arr[] = {10, 20, 30};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer: " << endl;
    cin >> k;

    int result = linearSearch(arr, arraySize, k);
    if (result >= 0) {
        cout << "The number " << arr[result] << " was found at index "
             << result << " of the array." << endl;
    } else {
        cout << "The number " << k << " was not found." << endl;
    }

    return 0;
}

```

Output

Clear

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


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

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int linearSearch(const int arr[], int size, int k) {
    for (int i = 0; i < size; ++i) {
        if (k == arr[i]) {
            return i;
        }
    }
    return -1;
}

int main() {
    const int arr[] = {};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer: " << endl;
    cin >> k;

    int result = linearSearch(arr, arraySize, k);
    if (result >= 0) {
        cout << "The number " << arr[result] << " was found at index "
             << result << " of the array." << endl;
    } else {
        cout << "Since array is empty the value " << k << " cannot be found." <<
            endl;
    }

    return 0;
}
```

Output

Clear

```
/tmp/VrOF9pp9am.o
Enter an integer:
10
Since array is empty the value 10 cannot be found.


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

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int linearSearch(const int arr[], int size, int k) {
    for (int i = 0; i < size; ++i) {
        if (k == arr[i]) {
            return i;
        }
    }
    return -1;
}

int main() {
    const int arr[] = {12, 13, 14, 15};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer: " << endl;
    cin >> k;

    int result = linearSearch(arr, arraySize, k);
    if (result >= 0) {
        cout << "The number " << arr[result] << " was found at index "
             << result << " of the array." << endl;
    } else {
        cout << "The given value " << k << " does not exist in the given array
            hence cant be found." << endl;
    }

    return 0;
}
```

Output

Clear

```
/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

```cpp
#include <iostream>
using namespace std;

int linearSearch(const int arr[], int size, int k) {
    for (int i = 0; i < size; ++i) {
        if (k == arr[i]) {
            return i;
        }
    }
    return -1;
}

int main() {
    const int arr[] = {12, 13, 14, 15};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer: " << endl;
    cin >> k;

    int result = linearSearch(arr, arraySize, k);
    if (result >= 0) {
        cout << "The number " << arr[result] << " was found at index "
             << result << " of the array." << endl;
    } else {
        cout << "You have inputted a string and array is of integer hence
            invalid data-type has been submitted" << endl;
    }

    return 0;
}
```

Output

Clear

```
/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

Share Run

```cpp
#include <iostream>
using namespace std;

int binarySearch(const int arr[], int low, int high, int k) {
    if (low > high) {
        return -1;
    }

    int mid = low + (high - low) / 2;

    if (arr[mid] == k) {
        return mid;
    }

    if (k < arr[mid]) {
        return binarySearch(arr, low, mid - 1, k);
    } else {
        return binarySearch(arr, mid + 1, high, k);
    }
}

int main() {
    const int arr[] = {1, 2, 3, 4, 5};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer to search for: " << endl;
    cin >> k;

    int result = binarySearch(arr, 0, arraySize - 1, k);

    if (result >= 0) {
        cout << "The number " << k << " was found at index " << result << " of the array." << endl;
    } else {
        cout << "The number " << k << " was not found in the array." << endl;
    }

    return 0;
}
```

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 ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int binarySearch(const int arr[], int low, int high, int k) {
    if (low > high) {
        return -1;
    }

    int mid = low + (high - low) / 2;

    if (arr[mid] == k) {
        return mid;
    }

    if (k < arr[mid]) {
        return binarySearch(arr, low, mid - 1, k);
    } else {
        return binarySearch(arr, mid + 1, high, k);
    }
}

int main() {
    const int arr[] = {10, 20, 30};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer to search for: " << endl;
    cin >> k;

    int result = binarySearch(arr, 0, arraySize - 1, k);

    if (result >= 0) {
        cout << "The number " << k << " was found at index " << result << " of the array." << endl;
    } else {
        cout << "The number " << k << " was not found in the array." << endl;
    }

    return 0;
}
```

Output

Clear

```
/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

```cpp
#include <iostream>
using namespace std;

int binarySearch(const int arr[], int low, int high, int k) {
    if (low > high) {
        return -1;
    }

    int mid = low + (high - low) / 2;

    if (arr[mid] == k) {
        return mid;
    }

    if (k < arr[mid]) {
        return binarySearch(arr, low, mid - 1, k);
    } else {
        return binarySearch(arr, mid + 1, high, k);
    }
}

int main() {
    const int arr[] = {};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer to search for: " << endl;
    cin >> k;

    int result = binarySearch(arr, 0, arraySize - 1, k);

    if (result >= 0) {
        cout << "The number " << k << " was found at index " << result << " of the array." << endl;
    } else {
        cout << "Since array is empty the value " << k << " cannot be found." << endl;
    }

    return 0;
}
```

Output

Clear

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


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

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

int binarySearch(const int arr[], int low, int high, int k) {
    if (low > high) {
        return -1;
    }

    int mid = low + (high - low) / 2;

    if (arr[mid] == k) {
        return mid;
    }

    if (k < arr[mid]) {
        return binarySearch(arr, low, mid - 1, k);
    } else {
        return binarySearch(arr, mid + 1, high, k);
    }
}

int main() {
    const int arr[] = {12, 13, 14, 15};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer to search for: " << endl;
    cin >> k;

    int result = binarySearch(arr, 0, arraySize - 1, k);

    if (result >= 0) {
        cout << "The number " << k << " was found at index " << result << " of the array." << endl;
    } else {
        cout << "The given value " << k << " does not exist in the given array hence cant be found." << endl;
    }

    return 0;
}
```

Output Clear

```
/tmp/gjfSlIrRJt.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 ===
```

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;

bool isSorted(const int arr[], int size) {
    for (int i = 1; i < size; ++i) {
        if (arr[i] < arr[i - 1]) {
            return false;
        }
    }
    return true;
}

int binarySearch(const int arr[], int low, int high, int k) {
    if (low > high) {
        return -1;
    }

    int mid = low + (high - low) / 2;

    if (arr[mid] == k) {
        return mid;
    }

    if (k < arr[mid]) {
        return binarySearch(arr, low, mid - 1, k);
    } else {
        return binarySearch(arr, mid + 1, high, k);
    }
}

int main() {
    const int arr[] = {10, 5, 2, 8};
    const int arraySize = sizeof(arr) / sizeof(arr[0]);
    int k;

    cout << "Enter an integer to search for: " << endl;
    cin >> k;

    if (isSorted(arr, arraySize)) {
        int result = binarySearch(arr, 0, arraySize - 1, k);

        if (result >= 0) {
            cout << "The number " << k << " was found at index " << result << " of the array." << endl;
        } else {
            cout << "The given value " << k << " does not exist in the given array." << endl;
        }
    } else {
        cout << "The array is unsorted. Binary search requires a sorted array." << endl;
    }

    return 0;
}
```

Output

Clear

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


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