Create a C++ program that receives as input an array of integers, sorts them, and allows the user to query the sorted array by index.

**ChatGPT said:**

ChatGPT

Here is a C++ program that:

1. Takes an array of integers as input.
2. Sorts the array in ascending order.
3. Allows the user to query the array by index to retrieve the sorted elements.

cpp

Copy code

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int n;

    // Ask the user for the number of elements in the array

    cout << "Enter the number of elements: ";

    cin >> n;

    vector<int> arr(n);

    // Input the array elements

    cout << "Enter the elements:" << endl;

    for (int i = 0; i < n; ++i) {

        cin >> arr[i];

    }

    // Sort the array in ascending order

    sort(arr.begin(), arr.end());

    // Output the sorted array

    cout << "Sorted array: ";

    for (int i = 0; i < n; ++i) {

        cout << arr[i] << " ";

    }

    cout << endl;

    // Allow the user to query the array by index

    while (true) {

        int index;

        cout << "Enter an index to query (0 to " << n - 1 << "), or -1 to exit: ";

        cin >> index;

        if (index == -1) {

            break;  // Exit the loop

        }

        // Check if the index is valid

        if (index >= 0 && index < n) {

            cout << "Element at index " << index << ": " << arr[index] << endl;

        } else {

            cout << "Invalid index. Please try again." << endl;

        }

    }

    return 0;

}

**Explanation:**

1. **Input Section:**
   * The program first asks the user for the number of elements (n), followed by reading the elements into a vector arr.
2. **Sorting:**
   * The program uses std::sort to sort the elements of the vector in ascending order.
3. **Query Section:**
   * After sorting, the user is prompted to input an index to retrieve the value at that index in the sorted array.
   * The user can keep querying until they input -1 to exit the program.
4. **Error Handling:**
   * The program checks whether the entered index is within the valid range. If it is, the corresponding value is displayed; otherwise, it prompts the user to try again.

**Example Run:**

vbnet

Copy code

Enter the number of elements: 5

Enter the elements:

3 1 4 1 5

Sorted array: 1 1 3 4 5

Enter an index to query (0 to 4), or -1 to exit: 2

Element at index 2: 3

Enter an index to query (0 to 4), or -1 to exit: 4

Element at index 4: 5

Enter an index to query (0 to 4), or -1 to exit: 10

Invalid index. Please try again.

Enter an index to query (0 to 4), or -1 to exit: -1

This program is a simple and interactive way to handle sorting and indexing queries.