The third greatest number.

Below code is use to find third greatest number from thes the set of numbers entered by users.

```
#include <iostream>

#include <vector>

#include <algorithm>

#include <limits> // For numeric_limits



using namespace std;



int main() {

    int size;

    bool validSize = false;



    // 1. Ask the user to enter the size of the element and ensure it's an integer greater than 3.

    while (!validSize) {

        cout << "Enter the size of the element (must be an integer greater than 3): ";



        // Input validation for size

        if (cin >> size) {

            if (size > 3) {

                validSize = true;

            } else {

                cout << "Size must be greater than 3. Please enter again." << endl;

            }

        } else {
```

```
        cout << "Invalid input. Please enter an integer." << endl;

        cin.clear(); // Clear error flags

        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Discard invalid input

    }

}



// 2. Ask the user to enter elements, ensuring each input is an integer and avoiding duplicates.

vector<int> elements;

cout << "Enter " << size << " unique elements:" << endl;



for (int i = 0; i < size; ++i) {

    int element;

    bool validElement = false;



    while (!validElement) {

        cout << "Enter element " << (i + 1) << ": ";



        // Input validation for each element

        if (cin >> element) {

            // Check for duplicates

            if (find(elements.begin(), elements.end(), element) == elements.end()) {

                elements.push_back(element);

                validElement = true;

            } else {

                cout << "Duplicate element. Please enter a unique value." << endl;

            }

        } else {

            cout << "Invalid input. Please enter an integer." << endl;
```

```
                cin.clear(); // Clear error flags

                cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Discard invalid input

            }

        }

    }



    // 3. Order entered numbers by ascending order.

    sort(elements.begin(), elements.end());



    // 4. Print out the third largest element.

    cout << "The third largest number is: " << elements[size - 3] << endl;



    return 0;

}
```

Description

1. Headers and Namespaces

- `#include <iostream>`: Used for input and output operations.
- `#include <vector>`: Allows dynamic arrays to store user inputs.
- `#include <algorithm>`: Provides functions like `sort()` to arrange elements.
- `#include <limits>`: Used for handling invalid inputs by clearing and ignoring input streams.
- `using namespace std;`: Avoids the need to prefix standard functions and objects like `cout`, `cin`, etc.

2. Step 1: Validating the Size Input

- **Purpose:** The program asks the user to input the number of elements (`size`) but ensures:
  - It's an **integer**.
  - It's greater than `3`.
- **How It Works:**
  - Inside the `while` loop, `cin >> size` is used to take input.
  - If invalid input (like a string) is detected, the error is cleared using `cin.clear()` and invalid data is discarded using `cin.ignore()`.
  - If the input is valid but not greater than `3`, the user is prompted to re-enter.

3. Step 2: Collecting Unique Elements

- **Purpose:** Asks the user to enter `size` integers while ensuring:
  - Each input is a valid integer.
  - There are **no duplicate values** in the input.
- **How It Works:**
  - A loop runs `size` times to collect inputs.
  - For each input:
    - Checks if it's a valid integer using `cin >> element`.
    - Uses `find()` to see if the entered element already exists in the `elements` vector. If not, it adds the element.
    - For invalid input or duplicates, prompts the user to re-enter a valid, unique number.

4. Step 3: Sorting Elements

- **Purpose:** Arranges the collected numbers in ascending order.
- **How It Works:**
  - `sort(elements.begin(), elements.end())`: Sorts the `elements` vector from the smallest to the largest number.

5. Step 4: Printing the Third Largest Element

- **Purpose:** Finds and outputs the third largest number from the sorted list.
- **How It Works:**
  - Since the list is sorted in ascending order, the third largest element is at the index `size - 3` in the sorted vector (`elements[size - 3]`).

Example Run:

Input:

```
Enter the size of the element (must be an integer greater than 3): 5
Enter 5 unique elements:
Enter element 1: 10
Enter element 2: 20
Enter element 3: 30
Enter element 4: 40
Enter element 5: 50
```

Output:

```
The third largest number is: 30
```