```cpp
/*
Allocate an array dynamically using malloc(), initialize it with values, and traverse it using pointer arithmetic.
Use realloc() to resize the array, then free the allocated memory.
*/
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    int no_of_elements;
    cout << "Enter the number of elements: ";
    cin >> no_of_elements;

    // Dynamically allocate memory using malloc()
    int* ptr = (int*)malloc(no_of_elements * sizeof(int));
    if (ptr == nullptr) {
        cout << "Memory allocation failed!" << endl;
        return 1;
    }

    // Initialize the array with values
    cout << "Enter " << no_of_elements << " elements:" << endl;
    for (int i = 0; i < no_of_elements; i++) {
        cout << "Element " << i + 1 << ": ";
        cin >> *(ptr + i);
    }

    // Traverse the array using pointer arithmetic
    cout << "Traversing the array:" << endl;
    for (int i = 0; i < no_of_elements; i++) {
        cout << "Element " << i + 1 << ": " << *(ptr + i) << endl;
    }

    // Resize the array using realloc()
    cout << "Enter the new number of elements: ";
    int new_no_of_elements;
    cin >> new_no_of_elements;

    ptr = (int*)realloc(ptr, new_no_of_elements * sizeof(int));
    if (ptr == nullptr) {
        cout << "Memory reallocation failed!" << endl;
        return 1;
    }

    // Initialize new elements if the array has grown
    if (new_no_of_elements > no_of_elements) {
        cout << "Enter " << new_no_of_elements - no_of_elements << " new elements:" << endl;
        for (int i = no_of_elements; i < new_no_of_elements; i++) {
            cout << "Element " << i + 1 << ": ";
            cin >> *(ptr + i);
        }
    }

    // Traverse the resized array using pointer arithmetic
    cout << "Traversing the resized array:" << endl;
    for (int i = 0; i < new_no_of_elements; i++) {
        cout << "Element " << i + 1 << ": " << *(ptr + i) << endl;
    }

    // Free the allocated memory
    free(ptr);

    return 0;
}

```