```
1
    Allocate an array dynamically using malloc(), initialize it with values, and
 2
     traverse it using pointer arithmetic.
     Use realloc() to resize the array, then free the allocated memory.
 3
    */
    #include <iostream>
 5
     #include <cstdlib>
 6
     using namespace std;
 8
 9
     int main() {
         int no_of_elements;
10
         cout << "Enter the number of elements: ";</pre>
11
         cin >> no_of_elements;
12
13
14
         // Dynamically allocate memory using malloc()
         int* ptr = (int*)malloc(no_of_elements * sizeof(int));
15
         if (ptr = nullptr) {
16
             cout << "Memory allocation failed!" << endl;</pre>
17
18
             return 1;
19
         }
20
21
         cout << "Enter " << no_of_elements << " elements:" << endl;</pre>
22
23
         for (int i = 0; i < no of elements; i++) {</pre>
             cout << "Element " << i + 1 << ": ";</pre>
24
             cin \gg *(ptr + i);
25
         }
26
27
28
         // Traverse the array using pointer arithmetic
29
         cout << "Traversing the array:" << endl;</pre>
         for (int i = 0; i < no_of_elements; i++) {</pre>
30
31
             cout << "Element " << i + 1 << ": " << *(ptr + i) << endl;
         }
32
33
         // Resize the array using realloc()
34
         cout << "Enter the new number of elements: ";</pre>
35
         int new_no_of_elements;
36
         cin >> new_no_of_elements;
37
38
         ptr = (int*)realloc(ptr, new_no_of_elements * sizeof(int));
39
40
         if (ptr = nullptr) {
41
             cout << "Memory reallocation failed!" << endl;</pre>
42
             return 1;
43
         }
44
45
         // Initialize new elements if the array has grown
46
         if (new_no_of_elements > no_of_elements) {
47
             cout << "Enter " << new_no_of_elements - no_of_elements << " new</pre>
     elements:" << endl;</pre>
48
             for (int i = no_of_elements; i < new_no_of_elements; i++) {</pre>
```

```
cout << "Element " << i + 1 << ": ";</pre>
49
                 cin \gg *(ptr + i);
50
51
52
53
         // Traverse the resized array using pointer arithmetic
54
         cout << "Traversing the resized array:" << endl;</pre>
55
         for (int i = 0; i < new_no_of_elements; i++) {</pre>
56
             cout << "Element " << i + 1 << ": " << *(ptr + i) << endl;
57
58
59
60
         // Free the allocated memory
         free(ptr);
61
62
63
         return 0;
64
65
```