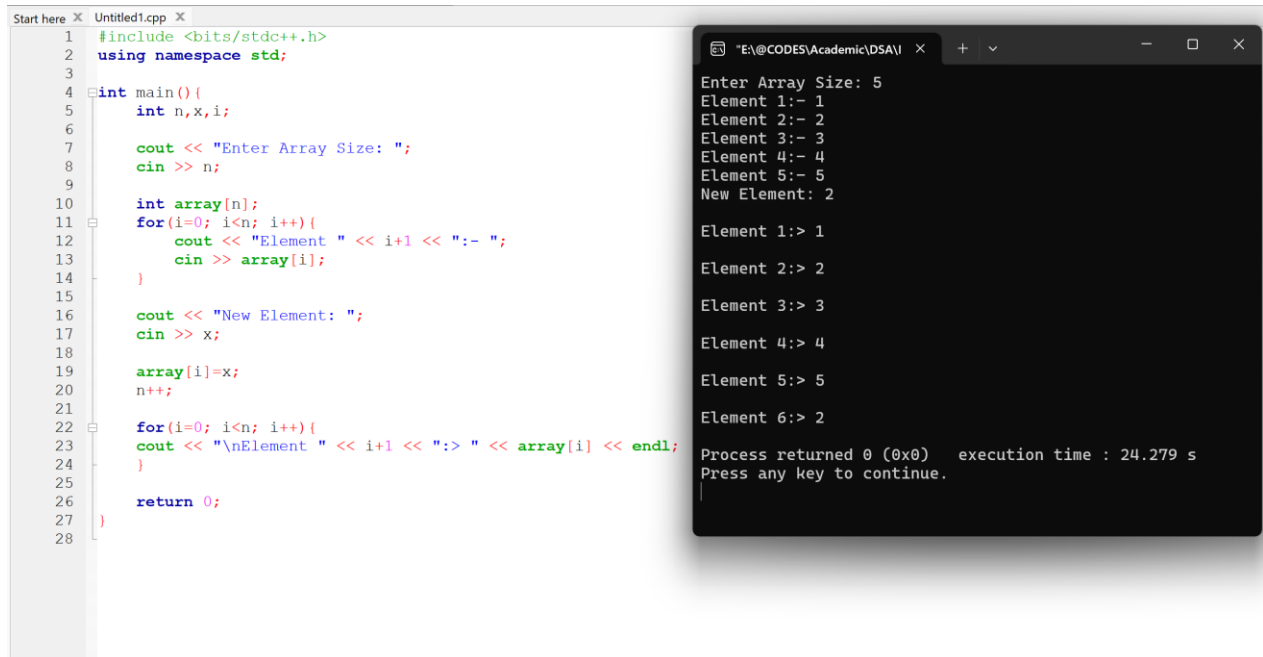


1. **Problem Name:** A program that attempts to add an element at the end of an array.

### Problem Code & Output:



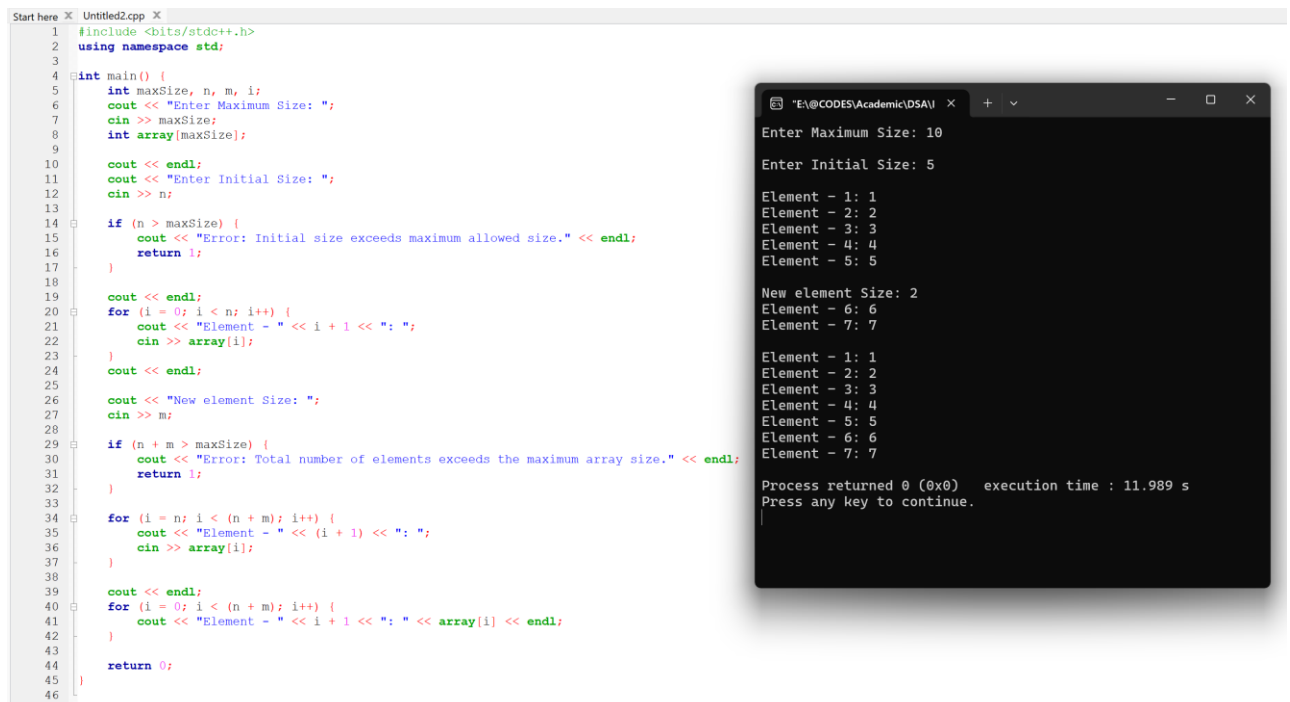
The screenshot shows a C++ program in a code editor and its execution output in a terminal window. The code defines a main function that takes an array size 'n' and an element 'x'. It creates an array of size 'n', fills it with values 1 through 'n', and then adds a new element 'x' at the end. The output shows the array elements and the new element being added.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n, x, i;
6
7     cout << "Enter Array Size: ";
8     cin >> n;
9
10    int array[n];
11    for(i=0; i<n; i++){
12        cout << "Element " << i+1 << ":- ";
13        cin >> array[i];
14    }
15
16    cout << "New Element: ";
17    cin >> x;
18
19    array[i]=x;
20    n++;
21
22    for(i=0; i<n; i++){
23        cout << "\nElement " << i+1 << ":- " << array[i] << endl;
24    }
25
26    return 0;
27 }
```

Enter Array Size: 5  
Element 1:- 1  
Element 2:- 2  
Element 3:- 3  
Element 4:- 4  
Element 5:- 5  
New Element: 2  
Element 1:-> 1  
Element 2:-> 2  
Element 3:-> 3  
Element 4:-> 4  
Element 5:-> 5  
Element 6:-> 2  
Process returned 0 (0x0) execution time : 24.279 s  
Press any key to continue.

2. **Problem Name:** A program that allows adding multiple elements to the end of an array after its initial creation.

### Problem Code & Output:



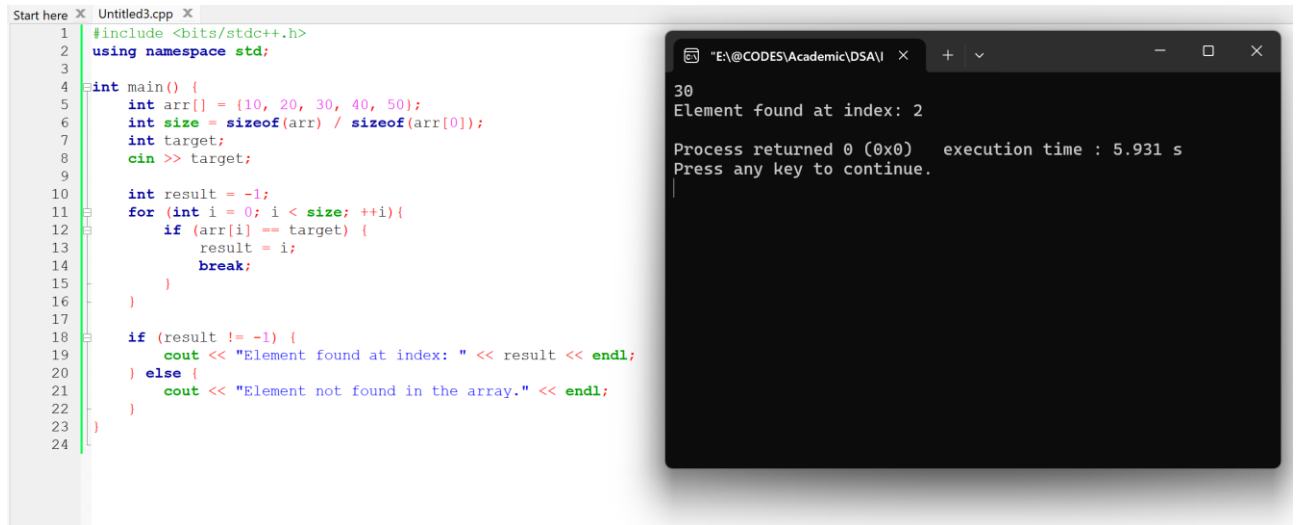
The screenshot shows a C++ program in a code editor and its execution output in a terminal window. The code defines a main function that takes a maximum size 'maxSize', an initial size 'n', and a new element size 'm'. It creates an array of size 'maxSize', fills it with values 1 through 'n', and then adds 'm' new elements at the end. The output shows the array elements and the new elements being added.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int maxSize, n, m, i;
6     cout << "Enter Maximum Size: ";
7     cin >> maxSize;
8     int array[maxSize];
9
10    cout << endl;
11    cout << "Enter Initial Size: ";
12    cin >> n;
13
14    if (n > maxSize) {
15        cout << "Error: Initial size exceeds maximum allowed size." << endl;
16        return 1;
17    }
18
19    cout << endl;
20    for (i = 0; i < n; i++) {
21        cout << "Element - " << i + 1 << ":- ";
22        cin >> array[i];
23    }
24    cout << endl;
25
26    cout << "New element Size: ";
27    cin >> m;
28
29    if (n + m > maxSize) {
30        cout << "Error: Total number of elements exceeds the maximum array size." << endl;
31        return 1;
32    }
33
34    for (i = n; i < (n + m); i++) {
35        cout << "Element - " << (i + 1) << ":- ";
36        cin >> array[i];
37    }
38
39    cout << endl;
40    for (i = 0; i < (n + m); i++) {
41        cout << "Element - " << i + 1 << ":- " << array[i] << endl;
42    }
43
44    return 0;
45 }
```

Enter Maximum Size: 10  
Enter Initial Size: 5  
Element - 1: 1  
Element - 2: 2  
Element - 3: 3  
Element - 4: 4  
Element - 5: 5  
New element Size: 2  
Element - 6: 6  
Element - 7: 7  
Element - 1: 1  
Element - 2: 2  
Element - 3: 3  
Element - 4: 4  
Element - 5: 5  
Element - 6: 6  
Element - 7: 7  
Process returned 0 (0x0) execution time : 11.989 s  
Press any key to continue.

**3. Problem Name:** A program that performs a linear search to find an element in an array.

### Problem Code & Output:



The image shows a C++ program in a code editor and its execution output in a terminal window. The program implements a linear search algorithm on an array of integers. The array contains the values 10, 20, 30, 40, and 50. The user has entered 30 as the target value. The program successfully finds the element at index 2 and displays the result.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int arr[] = {10, 20, 30, 40, 50};
6     int size = sizeof(arr) / sizeof(arr[0]);
7     int target;
8     cin >> target;
9
10    int result = -1;
11    for (int i = 0; i < size; ++i){
12        if (arr[i] == target) {
13            result = i;
14            break;
15        }
16    }
17
18    if (result != -1) {
19        cout << "Element found at index: " << result << endl;
20    } else {
21        cout << "Element not found in the array." << endl;
22    }
23 }
24
```

30  
Element found at index: 2  
Process returned 0 (0x0) execution time : 5.931 s  
Press any key to continue.