

## Lab 1: Array Insertion

Work on this exercise on your own. This will be a good practice to help you review the topic.

Using the project **lab\_01\_array\_insertion**, implement the **declaration** and **definition** of the following function where indicated:

- **insertAtIndex**
  - **Parameters:** the array, the number of elements in the array, the element to insert, and the index where the element needs to be inserted.
  - Example:
    - Array is: [10, 54, 81, 45, 95, 25, 12, 67]
    - Element to insert: 79
    - Index: 4
    - After inserting element → [10, 54, 81, 45, 79, 95, 25, 12, 67]
  - To insert an element, you need to shift all the elements.
  - Consider the following cases:
    - If the array is full, output the following error message "Array is full. Cannot insert another element." (Use **cerr** instead of **cout**.)
    - If the index is larger than the number of elements in the array (for example, the array contains 10 elements and the index is **11 or larger**), output the error message "You can only insert contiguous elements in the array." (Use **cerr** instead of **cout**.)
    - If the index exceeds the capacity, output the error message "The array cannot have more than ### elements." where **###** is the capacity of the array. (Use **cerr** instead of **cout**.)
  - Can use either **FOR** or **WHILE** loop.

Testing cases are already provided for you.

### Make sure you:

- Add a **name header** with your name, date, etc. (use the same format shown on the syllabus).
- Pass by **reference** when needed and you add the **const** modifier to the parameters **ONLY when necessary**.
- Do **NOT** use a **return** statement without returning anything! → **return;**
- Do **NOT** use the **break** and **continue** statements (there are no switch statements to use **break**).
- Do **NOT** use global variables **ever**.
- Do **NOT** modify any code given.

### Keep in mind the following:

- Divide your code in meaningful blocks for readability
- Name your variables using descriptive names
- Use all appropriate conventions for naming

- Do not leave unnecessary spaces or lines in your code

### Expected Output

```
Initial Array: No elements in the array.
Insert 10 at idx 0...
Modified array: 10
-----
Initial Array: 1
Insert 20 at idx 0...
Modified array: 20 1
-----
Initial Array: 3
Insert 30 at idx 1...
Modified array: 3 30
-----
Initial Array: 5 3
Insert 40 at idx 1...
Modified array: 5 40 3
-----
Initial Array: 7 4
Insert 50 at idx 2...
Modified array: 7 4 50
-----
Initial Array: 5 3 1 7
Insert 60 at idx 4...
Modified array: 5 3 1 7 60
-----
Initial Array: 4 2 7 4
Insert 70 at idx 5...
Modified array: You can only insert contiguous elements in the array.
4 2 7 4
-----
Initial Array: 8 4 2 6 7 8 2
Insert 80 at idx 7...
Modified array: 8 4 2 6 7 8 2 80
-----
Initial Array: 9 8 5 6 3 2 1 4
Insert 90 at idx 8...
Modified array: 9 8 5 6 3 2 1 4 90
-----
Initial Array: 1 6 4 8 9 0 7 5 2 3
Insert 100 at idx 10...
Modified array: Array is full. Cannot insert another element.
1 6 4 8 9 0 7 5 2 3
-----
Initial Array: 4 6 2
Insert 110 at idx 20...
Modified array: The array cannot have more than 10 elements.
4 6 2
-----
Initial Array: 0 1 2 3 4 5 6 7 8 9
Insert 120 at idx 5...
Modified array: Array is full. Cannot insert another element.
0 1 2 3 4 5 6 7 8 9
-----
Press any key to continue . . .
```