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