Exercise: Array Insertion

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

Using the project **ex_01_array_insertion**, implement the declaration and definition of the following functions:

- **fillArray**
  - **Parameters:** the array and the number of elements
  - Does not return a value.
  - Ask the user to enter **no more** than **20 positive integers** (the user might enter **less** than 20) and ask to type a **negative integer** when done.
  - Use a **while** loop.
- **printArray**
  - **Parameters:** the array and the number of elements
  - Does not return a value.
  - Prints the array.
  - If the array is empty, prints the error message, "Array is empty." (Use **cerr** instead of cout.)
  - Use a **for** loop.
- **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.
  - Does not return a value.
  - Example:

    Array is: [10,54,81,45,95,25,12,67]
    Element to insert: 79
    Index: 4
    After inserting element ➔ [10,54,81,95,79,25,12,67]
  - To insert an element, you need to shift all the elements and increment the variable that holds the number of elements in the array.
  - 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 *past* the end of the array (for example, the array contains 10 elements and the index is 13), 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 any loops.

The **main** function already contains testing cases.

Make sure you:

- Add a **name header** with your name, date, etc.
- 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

What to turn in:

- A **printed** copy of the following sections of your program:
  - Name header
  - Definition of function **fillArray**
  - Definition of function **printArray**
  - Definition of function **insertAtIndex**
  - Output (copy and paste the output so that the background is **NOT** black)

- Your **project folder**
  - On second class meeting next week at the beginning of class ➔ I will show you how to drop a project in the **Q drive**.

When to turn in your assignment:

- MW class → Wednesday, September 2, at the beginning of class.
- TTh class → Thursday, September 3, at the beginning of class.

Possible Output (next page)

```
Enter up to 20  non-negative integers.
Mark the end of the list with a negative integer.
3 5 7 9 -1

Array elements: 3 5 7 9

TEST insertAtIndex ----------------------------

Enter a number to insert: 7
Enter the index where to place the number: 2

Array is now: 3 5 7 7 9

Again (y/n)? y

Enter a number to insert: 25
Enter the index where to place the number: 5

Array is now: 3 5 7 7 9 25

Again (y/n)? y

Enter a number to insert: 100
Enter the index where to place the number: 33
The array cannot have more than 20 elements.

Array is now: 3 5 7 7 9 25

Again (y/n)? y

Enter a number to insert: 100
Enter the index where to place the number: 0

Array is now: 100 3 5 7 7 9 25

Again (y/n)? y

Enter a number to insert: 1
Enter the index where to place the number: 10
You can only insert contiguous elements in the array.

Array is now: 100 3 5 7 7 9 25

Again (y/n)? n

Press any key to continue . . .
```