COMP 53: Arrays and Vectors Lab, part 1

Instructions: In this lab, we are going to review arrays.

- Get into groups of **at most two people** to accomplish this lab.
- At the top of your source code files list the group members as a comment.
- Each member of the group must individually submit the lab in Canvas.
- This lab includes **27 points** in aggregate. The details are given in the following.

1 `main.cpp`

In `main.cpp` do the following step by step:

1. Globally define constant integer `SIZE` and set it to `10` ***(1 points)***.
2. Globally define array `a[]` of integers consisting of values: 5, 7,-2, 8, 11, -9, 4, 6, 12, and -1 in order ***(2 points)***.
3. Globally define array `b[]` of integers consisting of values: 4, 16, 9, -2, 1, 14, -4, 8, 10, and 0 in order ***(2 points)***.
4. Globally define array `c[]` of integers, without initial values ***(2 points)***.
5. Define function `printArray()` that receives an array of integers along with its size. It traverses the array and prints the elements in the standard output in a comma-separated form. The signature of the function may look like the following: `void printArray(int arr[], int size)` ***(2 points)***.
6. Define function `minArray()` that receives an array of integers along with its size. It traverses the array and returns the minimum element (so the return value is an integer) ***(2 points)***.
7. Define function `maxArray()` that receives an array of integers along with its size. It traverses the array and returns the maximum element (so the return value is an integer) ***(2 points)***.
8. Define function `sumArray()` that receives an array of integers along with its size. It traverses the array and returns the summation of all elements (so the return value is an integer) ***(2 points)***.
9. Define function `productArray()` that receives an array of integers along with its size. It traverses the array and returns the product of all elements (so the return value is an integer) ***(2 points)***.
10. Define function `copyArray()` that receives two arrays of integers and the size. It traverses the first array and copies elements from the first array to the second array. The function does not return a value ***(2 points)***.
11. Define function `greaterArray()` that receives three arrays of integers and the size. It traverses the first and second array, compares the elements in the same index, and copies the larger one to the element in the same index within the third array. The function does not return a value ***(3 points)***.

In `main()` function do the following step by step, using the functions defined above:

(a) Print out the minimum element of array `a[]` ***(1 points)***.

(b) Print out the maximum element of array `a[]` ***(1 points)***.

(c) Print out the summation of elements of array `a[]` ***(1 points)***.

(d) Print out the product of elements of array `a[]` ***(1 points)***.

(e) Copy array `a[]` to `c[]` and print out `c[]`'s elements ***(2 points)***.

(f) Collect larger elements from `a[]` and `b[]` to `c[]`. Next, print out `c[]`'s elements ***(3 points)***.

The output of the program may look like the following:

```
minimum of a[]: -9
maximum of a[]: 12
sum of a[]: 41
product of a[]: -15966720
copy a[] to c[]: 5, 7, -2, 8, 11, -9, 4, 6, 12, -1
collect larger elements from a[] and b[] to c[]: 5, 16, 9, 8, 11, 14, 4, 8, 12, 0
```