Homework 4(100pt)

Problem 1: Create the following functions:

```
void displayMenu(int& times);
int median(int a, int b, int c);
void mySort(int& a, int& b, int& c);
void myPermutation(int& a, int& b, int& c);
void mySimplify(int& a, int& b, int& c);
int main();
```

where

- (10pt) `displayMenu` prints a list of options to the console and shows the number of times `displayMenu` is called
- (10pt) `median` returns the median of three input integers
- (15pt) `mySort` sorts the three integers in descending order
- (15pt) `myPermutation` applies a permutation to the numbers – moving the last to the front (that is, a list of numbers `1 2 3` will become `3 1 2` after the function call)
- (15pt) `mySimplify` simplifies the three integers by dividing them by their maximum common factor
- (15pt) `main` keeps asking users to enter three positive integers until success (you can assume that users always enter valid integers) and then continuously displays a menu of options and call the corresponding function based on the user input until the user enters `Q` (you can assume the user always enters valid option numbers or `Q`)

Instructions:

- (5pt) Put all functions into one file, named `menu.cpp`, and only submit it to CCLE. Declare the ownership in the beginning of your file.

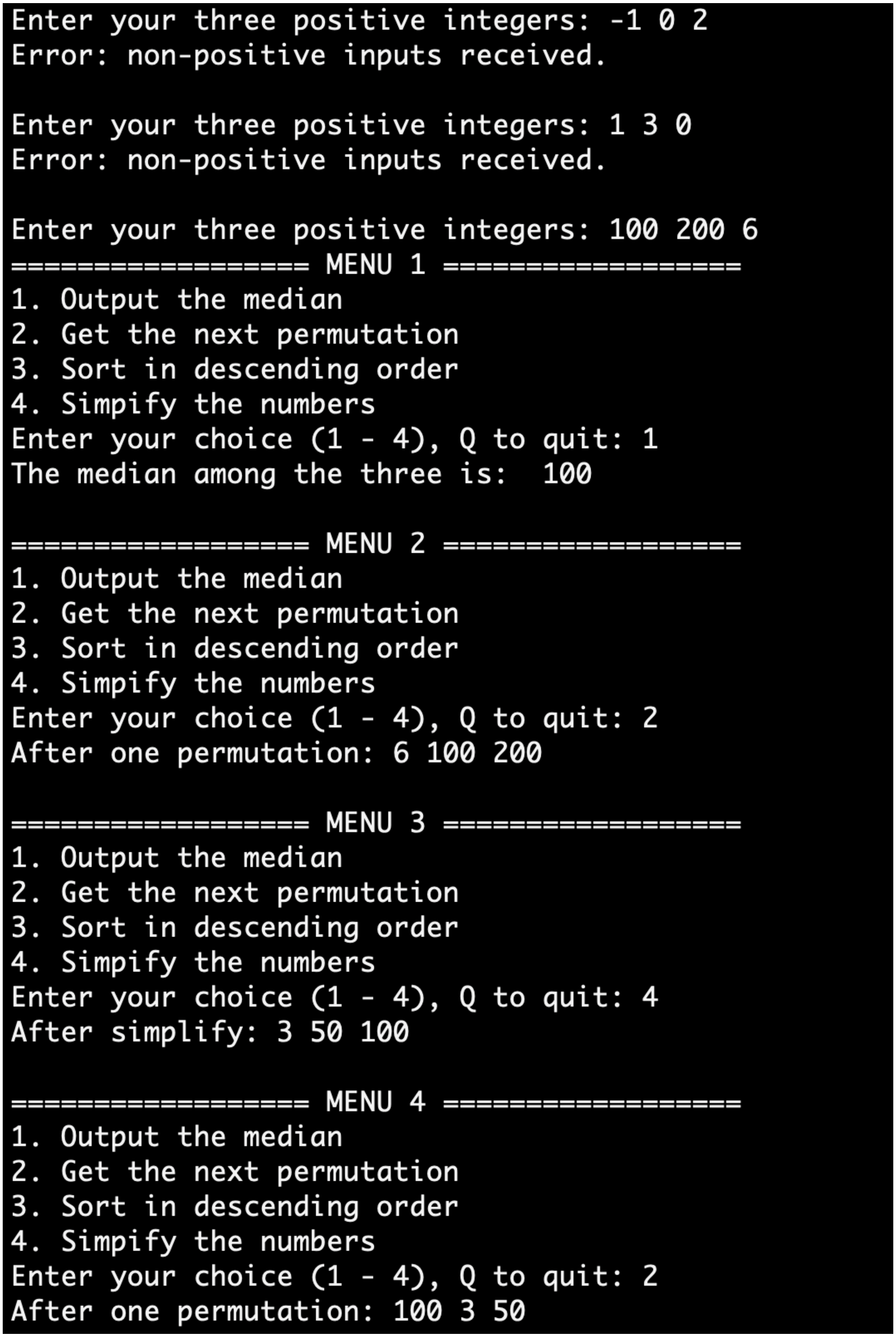
- (80pt) Implement all functions correctly. You are not allowed to use global variables. Your code should work for different inputs, e.g., two integers are equal.

- (15pt) Write your code with good coding practices, including commenting your code, using descriptive variable names, etc.

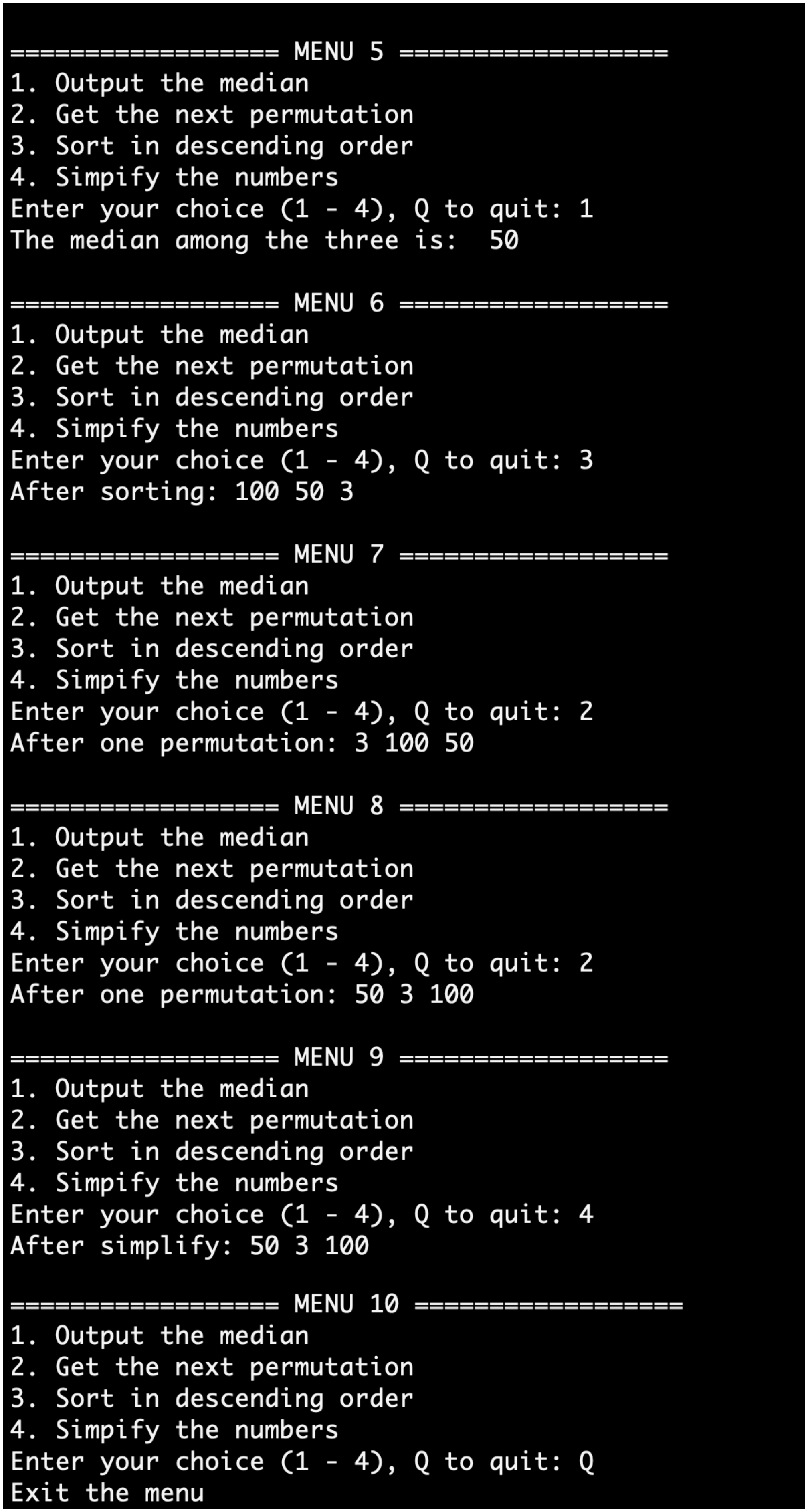
- Code compiles with Visual Studio 2019 and solves the questions. Students may lose the majority of points if their code doesn't compile with VS 2019. Students should test their work with Apporto virtual machines if they don't have Visual Studio 2019 available on their own computer. Please manually log out your account after using the virtual machine.

  To receive full credits, the output must look EXACTLY the same as instructed above, including words, spaces, symbols, etc. Your code should not only work for the above examples, but also work for other different inputs.

A sample run looks like:



```
Enter your three positive integers: -1 0 2
Error: non-positive inputs received.

Enter your three positive integers: 1 3 0
Error: non-positive inputs received.

Enter your three positive integers: 100 200 6
================== MENU 1 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 1
The median among the three is:  100

================== MENU 2 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 2
After one permutation: 6 100 200

================== MENU 3 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 4
After simplify: 3 50 100

================== MENU 4 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 2
After one permutation: 100 3 50
```



```
================== MENU 5 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 1
The median among the three is:  50

================== MENU 6 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 3
After sorting: 100 50 3

================== MENU 7 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 2
After one permutation: 3 100 50

================== MENU 8 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 2
After one permutation: 50 3 100

================== MENU 9 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: 4
After simplify: 50 3 100

================== MENU 10 ==================
1. Output the median
2. Get the next permutation
3. Sort in descending order
4. Simpify the numbers
Enter your choice (1 - 4), Q to quit: Q
Exit the menu
```