102-A01 Let's play with positive integers

Note about all the exercises in 103-Axx series

All the exercises in this series are going to be written as one big application spread over several C files;

- **main.c**
  - Will contain the *main* function of your program
  - It will offer a text based menu to the user which will allow him or her to call interactively each of the functions we will define below.
  - The menu will list all functions with a number associated to them.
  - The user will enter a choice as an integer and the *main* function will then prompt the user for value for the parameters. It will then call the function with the parameters.
- **tools.c**
  - will contain the definitions of all the functions we will write in this exercises series
- **tools.h**
  - will contain the prototype of all these functions and will be included from **tools.c** and **main.c**
- **tests.c**
  - Will contain another *main* function.
  - Unlike the one in the **main.c** file, this one is not meant to allow the user to interactively call the functions we wrote but will rather call them with pre-defined parameters value which we picked in order to test them.
  - We will compare the return values to make sure each call led to a correct answer and, if this is not the case, we will display a message on the screen.
  - Please note that you won't display anything at all if all calls are correct. This is a way to allow us to compile our code, run it, and see immediately if every test we designed so far was successful or if one or more of these tests displayed error messages to help us track down the bugs.

Obviously, we can't compile together **main.c** and **tests.c** since that would mean our application has two entry points. However, we can choose to compile our application either with **main.c** or **tests.c** and therefore have an executable which allows us to interactively call our functions or simply run a rudimentary test harness on them.

To compile your program in "interactive mode" use the following command line;

```
GCC tools.c main.c
```

To compile your program in "automatic testing mode" use the following command line;

GCC tools.c tests.c

In both cases, you don't need to mention **tools.h** on the above command lines and you will run your executable as you did with all previous codes;

./a.out

Work to do

- The main function in **tests.c** will be empty except for an include of **tools.h**.
- The file **tools.c** will contain only an include of **tools.h**.
- The file **tools.h** will be totally empty for now.
- The file **main.c** will include **tools.h** and contain the following code;

To get us started with this work, you will have to write the main for **main.c** so that it displays a text-based menu as follows;

Application Menu
Let us consider x and y, two positive integers…
Current Values: X = 0 Y = 0
Do you want to:
[1] Set new values for X and Y
[2] Compute the LCM of these two values
[3] Compute the GCD of these two values
[4] Compute x^y
[0] Exit this application
Your choice:

X and *Y* are two local variables of your main, initialized to zero. Their current value is displayed each time you display the whole menu.

You will write a function in **main.c** to display the menu. This function will take the variables *X* and *Y* as parameters to be able to display their current values on the screen. This function will be called from inside the main function each time we need to display the menu on the screen. It will have the following prototype;

void display_menu (int xvalue , int yvalue);

In the *main*, you will keep displaying the menu and then prompt the user for a choice (an integer). When the user selects 1, simply ask them new values for *X* and *Y*. When the user selects 0, exit your loop and program. As long as the user selects an invalid value, just display;

Invalid choice, please try again

and redisplay the whole menu. When the user selects a valid item, you will simply display;

Function not yet implemented

And then re-display the whole menu. Of course, later on, we will call the appropriate functions but we didn't write them yet.

Make sure you test your code so that it exits correctly, keep repeating the menu with an error message when an invalid choice is made or after displaying the function not yet implemented message when a correct entry is picked. Also make sure that you can use option 1 to change the values of *X* and *Y* and that the new values are correctly displayed when the menu is redisplayed. Write down your test harness.

Example(s)

n/a

Hints

- n/a

Testing

Input	Output	
Menu choice	Expected	Observed