COMS W3157 Advanced Programming, Lab Assignment #1

Please read this assignment carefully and follow the instructions EXACTLY.

Part 1: Learning Make (0 points)

The sample Makefile in the lecture note is a good starting point for the Makefile you need to write for this lab. If you understand everything about the sample Makefile, you should have no problem writing this lab's Makefile.

Here are some additional documentations on Make:

- Stanford CS Education Library has a nice short tutorial on Makefiles. See Unix Programming Tools, Section 2, available at http://cslibrary.stanford.edu/107/.

- The manual for make is available at http://www.gnu.org/software/make/manual/make.html. Reading the first couple of chapters should be sufficient for basic understanding.

There are a few rules that we will follow when writing makefiles in this class:

- Always provide a "clean" target to remove the intermediate and executable files.

- Compile with gcc rather than cc.

- Use "-Wall" option for all compilations.

- Use "-g" option for compiling and linking.

The sample Makefile follows all these rules.

Part 2: C programming (50 points)

Write a C program that reads 2 positive integers from the user, and prints:

- the average of the two (this should be printed as a floating point number.)

- whether each number is a prime number or not

- whether the two numbers are relatively prime (see http://en.wikipedia.org/wiki/Coprime if you don't know what this means.)

You can assume that the user will input only positive integers, i.e., don't do any error checking.

In order to see if two numbers are relatively prime, you should calculate the GCD using Euclidean algorithm. You are allowed to look up the algorithm and/or code on the Internet, in which case you should cite the source in your README.txt file.

Your code should be organized as follows:

- gcd.h & gcd.c: GCD calculation function header and definition
- prime.h & prime.c: prime number testing function header and definition
- main.c: everything else
- Makefile
- README.txt

All files must be named EXACTLY as above, case-sensitive. When you run "make", it should build an executable file called "main".

The README.txt should contain:

- your name
- your UNI
- lab assignment number
- description of your solution

The description can be as short as "My code works exactly as specified in the lab." You may also want to include anything else you would like to communicate to the grader such as extra functionalities you implemented or how you tried to fix your non-working code.

The makefile should have correct dependencies. For example, if you build everything, change prime.h, and run make again, only prime.c and main.c should recompile, not gcd.c. (You can simulate changing a file by using 'touch' command.)

You can use

```
    printf("You typed in %d and %d\n", x, y);
```

to print integers, and

```
    printf("The average is: %f\n", avg);
```

to print a floating point number.

And you can use

```
    scanf("%d", &x);
```

to read an integer that the user types in. Don't forget the ampersand in front of the variable.

--

Good luck!