

CS 232 Introduction to C and Unix

HW2 (Due on Jan. 26, 1:30pm)

The goal of this homework assignment is to test your ability to write a simple program in C. After this homework assignment, you should feel comfortable writing programs in C that use variables, operators, and control flow statements. You should also be familiar with the development process on a *nix computer. Please read the entire homework description carefully before beginning the coding.

Collaboration policy:

Since this is the first real homework assignment, I want to be up-front about the collaboration policy for this course. For this homework assignment, you are to work alone. This means that you are not to discuss this homework with anyone other than me (the instructor). You can go over general concepts (e.g., for loops, variable declarations, using cloud9, etc.) with other class members, but talking about how to figure out prime numbers, looking at others' code, helping them debug it, etc. is expressly prohibited. The purpose behind this policy is to make sure that every member of the class is evaluated fairly, based upon his or her own work. If you have any questions about what is or is not acceptable collaboration, please contact me; it is better to be safe than sorry. Also, if you get stuck on part of the homework, you are encouraged to come by office hours. If you cannot make office hours, just send me an email and we can find a time to meet. I'm here to help you, but I do not know how to help if you don't ask me.

Homework description:

In Cloud9 CS50 workspace, under homework directory, create a new subdirectory "hw2". In the Blackboard, you can find a file "primes.c" with part of a program already written. You should only modify that file in the places that are marked "**TODO: fill in the code ...**". Upload primes.c to your Cloud9 workspace under "hw2" directory.

For this assignment, you will complete the primes.c file so that it prints out all of the prime numbers that are greater than or equal to lowerBound and less than or equal to upperBound. (The values of these variables come from the command line. We will discuss how this is done later in the course.) The prime numbers should be printed in ascending order, with one number on each line. The program should not generate any other output. The number 1 should not be considered a prime number for this homework.

As an example, for lowerBound=1 and upperBound=7, the program should output:

```
2
3
5
7
```

Note that your program should not depend upon other files; all of your work should be done by filling in the appropriate section of `primes.c`. Furthermore, do not include any precomputed list of primes in your code. Here is a sample run in Cloud9:

```
~/workspace/homework/hw2/ (master) $ gcc -o primes primes.c
```

```
~/workspace/homework/hw2/ (master) $ ./primes 2 10
```

```
2
```

```
3
```

```
5
```

```
7
```

Compiling and Testing:

After you have written your program, you will need to compile it using `gcc`. (Look at the lecture slides for how to use `gcc`. If you use functions from the math library, such as `log` or `sqrt`, you will need to link with the math library by adding a `-lm` at the end of the `gcc` command.) If successful, `gcc` will produce a program (e.g., the file `"primes"`), which you will want to test for correctness. You can do this by running the program (from the shell) with two command-line arguments for the lower and upper bounds. For example, if your program is `"primes"` then you would run something like `"./primes 10 20"`, `"./primes 3 7"`, or `"./primes -10, 0"` and check if the output is actually the proper list of primes.

When you think the program is running correctly, you can test it using the provided utility. You can find two files in the Blackboard: `"primesCheckOutput"` and `"runSimpleTest.sh"`. Upload these two files to your Cloud9 workspace under `hw2` directory. Add the execution permission to them by using `chmod` command:

```
chmod +x primesCheckOutput
chmod +x runSimpleTests.sh
```

Now, if you run `"./runSimpleTests.sh"` from the shell, that utility will compile `primes.c` to `primes`, then run `primes` on several bounds, checking the output. You won't get to see what your program prints out, but you will get messages telling you if your program passed the tests or not. You can then run your program manually with the bounds that caused a problem in order to see what might be going wrong. If your program passes all the test cases, that is a good indication that your program is correct.

Submission

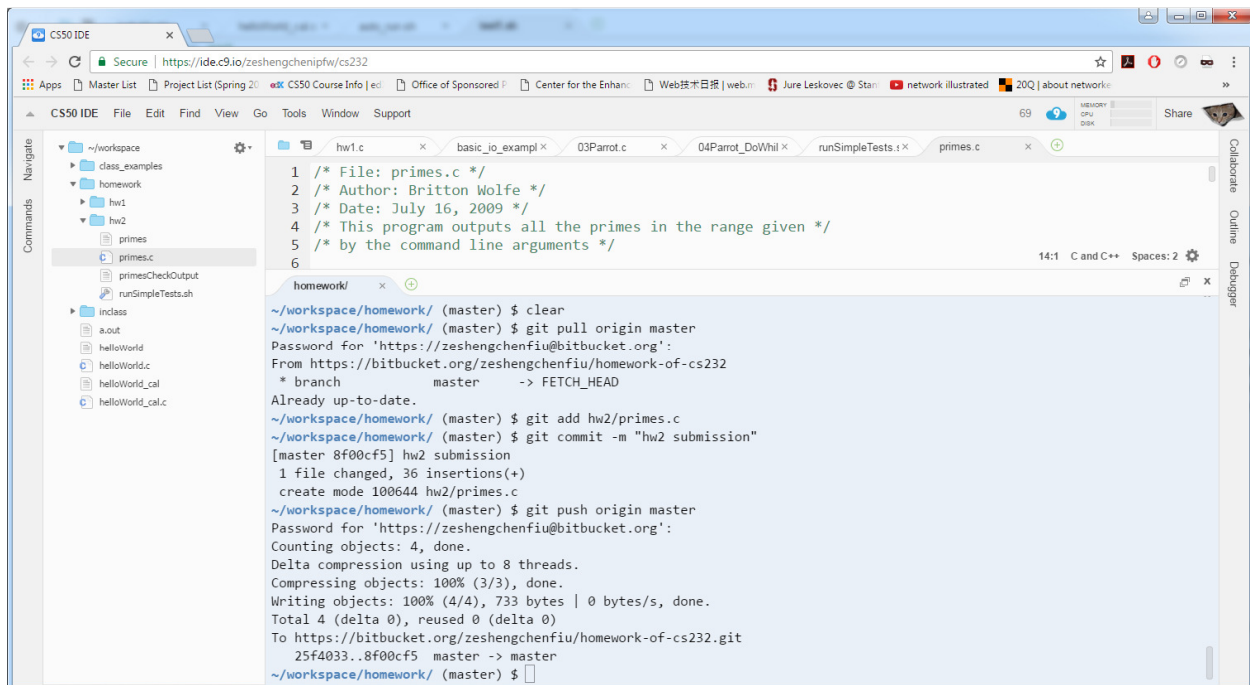
After testing the program, push the code primes.c to your Bitbucket “Homework of CS232” repository under the directory “hw2”.

Before pushing the code, please use Git to run

```
$ git pull origin master
```

TA may have uploaded the score file to your Bitbucket repository. Running the above command can get the score file and keep your cloud9 homework directory in sync with your Bitbucket repository. If you have trouble in pushing the code, please let me know.

Moreover, when you attempt to push the code from cloud9 to Bitbucket, please make sure you are under the “homework” directory, instead of “hw2”. Here is an example of pushing the code for this homework:



```
1 /* File: primes.c */
2 /* Author: Britton Wolfe */
3 /* Date: July 16, 2009 */
4 /* This program outputs all the primes in the range given */
5 /* by the command line arguments */
6

~/workspace/homework/ (master) $ clear
~/workspace/homework/ (master) $ git pull origin master
Password for 'https://zeshengchenfiu@bitbucket.org':
From https://bitbucket.org/zeshengchenfiu/homework-of-cs232
 * branch      master       -> FETCH_HEAD
Already up-to-date.
~/workspace/homework/ (master) $ git add hw2/primes.c
~/workspace/homework/ (master) $ git commit -m "hw2 submission"
[master 8f00cf5] hw2 submission
1 file changed, 36 insertions(+)
create mode 100644 hw2/primes.c
~/workspace/homework/ (master) $ git push origin master
Password for 'https://zeshengchenfiu@bitbucket.org':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 733 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/zeshengchenfiu/homework-of-cs232.git
25f4033..8f00cf5  master -> master
~/workspace/homework/ (master) $
```

If you have any questions, please let the instructor know.

Grading rubric:

- Push primes.c to Bitbucket under hw2 directory using Git – 10pt
- Program primes.c can be compiled – 10pt
- Program primes.c works correctly (e.g., pass runSimpleTests.sh tests) – 70pt

- Program `primes.c` uses correct indentation, and the code is readable with necessary comments – 10pt

Total: 100 points.