CSE 333 13su Exercise 4

out: Wednesday, July 3, 2013
due: Friday, July 5, 2013 by **9:00 am**.

Your job is to write a multi-file C program. You should write the following three files:

- GetPrime.h: a header file, containing a single function prototype for a function called "GetPrime()", as well as comments above the prototype documenting how to use the function. The function should accept a single uint16_t parameter, and it should return a uint64_t. The function should return the nth prime number, where "n" is the function's parameter. Note that GetPrime(1) should return 2, GetPrime(2) should return 3, GetPrime(3) should return 5, and so on.
- GetPrime.c: a file containing the implementation of GetPrime(). Feel free to use the simplest possible primality testing algorithm. You'll probably want to define a helper function as well.
- ex4.c: a file containing a main() function that tests GetPrime().

Your code must:

- compile without errors or warnings on CSE Linux machines (lab workstations, attu, or CSE home VM)
- have no crashes, memory leaks, or memory errors on CSE linux machines
- be contained in the three files described above. We will compile your code with the following commands:

```
bash$ gcc -Wall -g -std=gnu99 -o GetPrime.o -c GetPrime.c
bash$ gcc -Wall -g -std=gnu99 -o ex4.o -c ex4.c
bash$ gcc -Wall -g -std=gnu99 -o ex4 ex4.o GetPrime.o
```

- be pretty: the formatting, modularization, variable and function names, and so on must make us smile rather than cry. (Suggestion: see if `clint` reports any problems.)
- be robust: you should think about handling bogus input from the user, and you should handle hard-to-handle cases (if there are any) gracefully.
- have a comment at the top of your ex4.c file with your name, student number, and CSE or UW email address.

You should submit your exercise using the assignment dropbox linked on the main course web page.