

CS252: Homework 3

Prime Number Finder

Purpose:

The purpose of this assignment is to give you practice programming in C and creating a prime number finder which you will use for future homework and labs.

Submission Requirements:

Submit your code (.c files) along with screenshots of your code compiling and running using the Unix-level time command to determine how long it takes. Also include a screenshot of your code passing Valgrind with no memory leaks.

Grading Criteria:

Your code must compile to receive a score. Programs without comments will not receive full credit.

Problem 1: Prime Number Finder

The focus of this problem includes:

- Preparation for future lab and homework
- Practice coding in C

Write a prime number finder. This can be done in two stages.

The first stage should be to write a function that will tell you if a given number is prime or not. Simply test the numbers from 2 up to the number to see if any of them evenly divide the number. If any do, then the number is not prime. You should be sure you stop early if you discover a number is not prime. And please be sure to use the correct type of loop for this (while). Do not use breaks or returns from the middle of the loop as these are poor coding practices.

The second stage is to write code that finds how many primes are in a given range. For example, the number of primes in the range [10..100] is 21. And for [1,000..1,000,000] it is 78,330. Make sure your numbers come out correct for this serial test. This is simply a loop that keeps calling your prime checking the first stage code.

You should also time how long it takes to determine the number. In the C version, you may use the Unix-level time command to do so: `time ./findprime`.

Bonus Points: Earn up to 2 bonus points by creating a prime finder that is faster than mine.

Sample output:

CS252: Homework 3

Prime Number Finder

```
~/cs252/homework> gcc -o findprime hw3.c
Mon Nov 04 16:54:23 gurunghl@Thing1
~/cs252/homework> time ./findprime
start: 1000, end: 1000000
Total number of primes: 78330

real    2m30.111s
user    2m30.105s
sys      0m0.002s
```

```
~/cs252/homework> valgrind ./findprime
==19450== Memcheck, a memory error detector
==19450== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==19450== Using Valgrind-3.14.0 and LibVEX; rerun with -h for copyright info
==19450== Command: ./findprime
==19450==
start: 10, end: 100
Total number of primes: 21
==19450==
==19450== HEAP SUMMARY:
==19450==      in use at exit: 0 bytes in 0 blocks
==19450==    total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==19450==
==19450== All heap blocks were freed -- no leaks are possible
==19450==
==19450== For counts of detected and suppressed errors, rerun with: -v
==19450== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```