2020/09/23 - Prime Number Primer (Part 1)

2020年9月21日　23:55

SYNOPSIS

-Go over LAB 3A.

LAB 3A

-On Canvas, go to the "Lab 3.1" assignment and read the "lab3.html" file attached. All lab details will be there.

-Again... there is 4 parts. There are 2 due dates:
  - Sept 26 (SAT): Prime1, Prime2 (PRIME DETECTION)
  - Oct 3 (SAT): Prime3, Prime4 (PRIMES + STL)

-ALL PARTS have given template code you need to copy to start the lab.
  -Not really for later parts. Just C+P tbh...

SUBMISSION COMMANDS

LAB 3.1

```
tar -cvf lab3_1.tar Prime1.cpp Prime2.cpp
```

LAB 3.2

```
tar -cvf lab3_2.tar Prime3.cpp Prime4.cpp
```

PRIME 1

- Simple. Write a function "isprime" which will return true if a given number is prime, and return false otherwise.

- Exhaust stdin. aka, read numbers until CTRL+D.

- If a number is prime, say so.

- 1 IS NOT A PRIME NUMBER. NOR IS 0. I got into an argument with a few students over it last year and humiliated them by Googling "is 1 a prime number" on the projector. Don't @ me. #StoryTimeWithClara

PRIME 2

- Introduction to functors.
  - Essentially a class that can act like a function. This is done by overloading the operator() function.
  - Why? Class instance has memory which surpasses function scope. And, STL functions which take functions as parameter (aka std::transform *HINT, HINT*) can also take functors.

- Make a class called isprime like so:

```
class isprime {
    public:
        isprime();
        bool operator()(int);
    private:
        void magic(int);
        bool is_prime(int);
        vector<int> cache;
};
```

- We are going to make a way to store prime numbers for lookup rather than checking each time.
- Have the constructor throw 2 and 3 in the cache at the start.
  - Directions say only to put 2 in. I don't care.
- Calling operator() will put in ALL PRIME NUMBERS PRIOR TO N into cache, if not already there.

(Ex. START

[2, 3]

OPERATOR()(11)

[2, 3 5, 7, 11]

OPERATOR()(18)

[2, 3 5, 7, 11, 13 17]

- After inserting, call std::find on cache and return true if the given number n is in cache. Return false otherwise.
  - std::find is in #include <algorithm> btw...

Function Breakdown

operator()(int N)

- Call magic(N) to adjust cache.
- Call std::find and return true or false accordingly.

magic(int N)

- Loop from last number in cache to N (including N, so i <= N). If a number between them is prime (check via is_prime()), it is added into cache.

is_prime(int N)

- Does exactly what you think it does.
- Not obvious? Return true if N is prime. Return false otherwise.
- C+P from Prime1.cpp tbh.

- No. You don't have to follow my class structure. But you must follow the directions in the writeup.

Prime3 + Prime4

- Stay tuned!