Notes for testing this

Testing descrepancies

Our algorithm uses floats - so some disgrepancies will be noted with testing (cleared this with Ben).

For the n={7,15,51} and i=300 cases the results are identical.

However, for the n=301 and i=500 (lab benchmark) there was minor discrepancies. These were only between the difference between the strings "-0.00000" and "0.00000". Since there is no such thing as negative zero, we devised the `better_cmp.sh` script to use for testing. It uses `sed` to preform a find and replace so all instances of "-0.00000" become "0.00000".

Usage of `better_cmp.sh` is exactly the same as `cmp`

```
# should pass
better_cmp.sh 301_500_double.txt 301_500_float.txt
./main.o 301 500 > test.txt; better_cmp test.txt 301_500_float.txt

# should fail (we manually edited the bad version)
better_cmp.sh 301_500_bad.txt 301_500_float.txt
```

Compiling

Our optiminal version can be compiled with `make`. However if you would like to test different options:

```
# optimal version
gcc -o main.o -O3 -Ofast -DPRINT_OUTPUT_OLD main.c -lpthread

# eight threads version
gcc -o main.o -O3 -Ofast -DPRINT_OUTPUT_OLD -DEIGHT_THREADS main.c -lpthread

# use doubles instead of floating points
gcc -o main.o -O3 -Ofast -DPRINT_OUTPUT_OLD -DUSE_DOUBLE main.c -lpthread

# print only the output time (format is "<nanosecond> <seconds>") using CLOCK_MONOTONIC
gcc -o main.o -O3 -Ofast -DPRINT_TIME main.c -lpthread
```

Running

```
# ./main.o <n> <i>
./main.o 301 500
```