

TCES 202
Advanced Computer Programming
Assignment 2 – Stacks & Queues
Submit sieve.h and sieve.c on Canvas.
25 Points

This assignment will give you practice with queues. It must be completed in groups or pairs and submitted before due date and time. ***Late assignments will not be graded.*** Go through the starter code provided before solving the homework. You are going to implement functions that compute all the primes up to some integer n . The technique you are to use was developed by a Greek named Eratosthenes who lived in the third century BC. The technique is known as the Sieve of Eratosthenes.

The algorithm is described by the following pseudocode:

```
create a queue and fill it with the consecutive integers 2 through n inclusive.
create an empty queue to store primes.
do {
    obtain the next prime p by removing the first value in the queue of numbers.
    put p into the queue of primes.
    go through the queue of numbers, eliminating numbers divisible by p.
} while (p < sqrt(n))
all remaining values in numbers queue are prime, so transfer them to primes queue
```

You are to use the queue code discussed in lecture and use the STARTER CODE provided.

You should define a struct for the sieve with the appropriate members and must define the following functions:

Functions Description

`compute_to` - This is the function that should implement the sieve algorithm. All prime computations must be implemented using this algorithm. The function should compute all primes up to and including n .

`report_results` - This function should report the primes to console. See `sieve.txt` for sample output.

`get_max` - This is a convenience function that will let the client (main) find out the value of n that was used the last time `computeTo` was called.

`get_count` - This function should return the number of primes that were found on the last call on `computeTo`.

Your `report_results` function should print the maximum `n` used and should then show a list of the primes, 12 per line with a space (width needs to be changed for larger numbers) after each prime. Notice that there is no guarantee that the number of primes will be a multiple of 12. The calls on `report_results` must exactly reproduce the format of the sample log. The final line of output that appears in the log reporting the percentage of primes is generated by the main program, not by the call on `report_results`.