This is an individual project. This project is very challenging and carries a negligible weight (3%) in the total grade, so you may consider skipping it unless you want an A-level grade (A^-, A, A^+). Make your algorithm as efficient as possible, and write a rigorous analysis. You may write the analysis as comments or attach a separate text file (only a plain text file is accepted).

We have been dealing with a small amount of data during the semester. This project gives you a taste of massive data, or to be precise, infinite. We deal with natural numbers. I hope you have learned in primary school that only a tiny fraction of natural numbers are prime. The task is to list as many prime numbers as possible with the classical sieve method. For example, you may create a data structure (array or linked list?) for the numbers from 2 to 2^{30} ($\approx 10^9$) and use the sieve method to remove the non-prime numbers. (One billion is *small* for a modern computer.) Only a tiny proposition is left. Output them into a file and continue to process larger ones. The file name is your student ID: e.g., use `12345678d.txt` if your student ID is 12345678d.

```
// Running time: O(    )
void smallestPrimes (String fileName)
```

With more and more numbers, your program will eventually exhaust your memory and crash. We will use the same computer to grade all your submissions, and your grade depends on the number of primes you output before that happens.

You may use the lab computers or the department servers if your laptop does not have sufficient memory for testing.