## Concurrent & Parallel Programming

**Question 1**

Given below is a single threaded program that computes the index of the **leftmost zero** in a huge array. Your task is to write a parallel solution that distributes the workload fairly over the threads. You may assume that your machine has 4 cores. It is important to write an optimal solution.

public class FindLeftmostZero {

static final int N = 10000000;

public static void main(String[] args) {

int data[] = new int[N];

//assume occurrence of zero equally likely for all numbers generated

for(int j = 0; j < N; j++) data[j] = (int)(Math.random()\*N);

int index = 0;

while(index < data.length && data[index] != 0) index++;

if(index == data.length) System.out.println("No zero");

else System.out.println(index);

}

}

**Question 2 (Sieve of Erathostenes)**

A [prime number](http://en.wikipedia.org/wiki/Prime_number) is a [natural number](http://en.wikipedia.org/wiki/Natural_number) which has exactly two distinct natural number [divisors](http://en.wikipedia.org/wiki/Divisor): [1](http://en.wikipedia.org/wiki/1_(number)) and itself. Erathostenes deivsed a strategy to find prime numbers up to a given number N. Place all the numbers in a bag and by a process of elimination sieve out all the non-prime values leaving a bag of primes.

To find all the prime numbers less than or equal to a given integer *n* by Eratosthenes' method:

1. Create a list of consecutive integers from 2 to *n*: (2, 3, 4, ..., *n*).
2. Initially, let *p* equal 2, the first prime number.
3. Starting from *p2*, count up in increments of *p* and mark each of these numbers greater than *p* itself in the list. These numbers will be 2*p*, 3*p*, 4*p*, etc.; note that some of them may have already been marked.
4. Find the first number greater than *p* in the list that is not marked; let *p* now equal this number (which is the next prime).
5. If *p* is less than *n*, repeat from step 3. Otherwise, stop.

When the algorithm terminates, all the numbers in the list that are not marked are prime. A single threaded solution is given in the java file on Moodle accompanying this assignment.

Your task is to write a simple multi-threaded solution and test its performance.