# Eastern University, Sri Lanka
## Second Examination in Science –First Semester-2002/2003
### OC251 Java Programming – Practical

Time: 3 Hours

The prime numbers 2, 3, 5, 7, 11, 13, 17... are exceedingly important in Computer science because of the part they play in error – correcting codes and data encryption. This problem is about algorithms to find all the prime numbers up to N=10,000. We give you algorithm 1 and 3 and you your self must design algorithm 2. You must implement all three algorithms, run them, and determine which are the best and worst.

Algorithm 1
In this algorithm you consider every number from 2 up to N. If it is prime you include in the list. To test whether a number m is prime number you simply divide it by each of the number 2, 3, ..., m-1; if any of these divisions is without remainder then m is not prime, otherwise it is.

Algorithm 2
Modify Algorithm 1 to make the test for primality more efficient.

Algorithm 3
This algorithm is a little more tricky. It was discovered by the ancient Greeks and nowadays we attribute it to Eratosthenes of Cyrene (276BC – 194 BC). You begin with an array of booleans with one position for each index from 2 up to N (Java arrays begin at index 0 but that just means that we won't actually use positions 0 and 1 ). You fill this array with value "true" at each position, except for P[0] and P[1] which are set to "false". The next step is repeated until we are finished.

We look for the next position in the array which has a "true"value (This will be position 2 initially of course). Suppose this position t. Then we make P[2t],P[4t],.... all "false" and we add t to our list of primes .

That is it! Carry out a few hand calculations to ensure you understand what is going on, in particular what "to be finished" means.

You have to do the following:

1) Create an application class to implement these three algorithms.
2) Add static methods for these three algorithms.
3) Your program should output the primes it finds  form each algorithm using System.out.println.
4) Your program should also output the time each algorithm takes to execute to the standard error stream (System.err).

For example

```
  int start=System.currentTimeMillis();
   // call a method to do the desired algorithm.
int end =System.currentTimeMillis();
System.err.print(end-start);
```

Submission of this exam has two parts.

(i) Submit a floppy containing your source code.

(ii) Submit answer scripts which contain

(a) Description (design) of algorithm 2.

(b) Identification which algorithm is best and which algorithm is worst with brief explanation.

(If you are careful to write clean, well-commented code which makes the specification you can get good marks).