```
/******************************************************************************
 *  Compilation:  javac PrimeSieve.java
 *  Execution:    java -Xmx1100m PrimeSieve N
 *
 *  Computes the number of primes <= to N using the Sieve of Eratosthenes.
 *
 *  % java PrimeSieve 100
 *  The number of primes <= 100 is 25
 *
 *  % java -Xmx100m PrimeSieve 100000000
 *  The number of primes <= 100000000 is 5761455
 *
 *  % java PrimeSieve -Xmx1100m 1000000000
 *  The number of primes <= 1000000000 is 50847534
 *
 *  The 110MB and 1100MB is the amount of memory you want to allocate
 *  to the program. If your computer has less, make this number smaller,
 *  but it may prevent you from solving the problem for very large
 *  values of N.
 *
 *                  N     Primes <= N
 *  ---------------------------------
 *                 10               4
 *.                25               9
 *               100              25
 *             1,000             168
 *            10,000           1,229
 *           100,000           9,592
 *         1,000,000          78,498
 *        10,000,000         664,579
 *       100,000,000       5,761,455
 *     1,000,000,000      50,847,534
 ******************************************************************************/

public class PrimeSieve {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);

        // initially assume all integers are prime
        boolean[] isPrime = new boolean[N + 1];
        for (int i = 2; i <= N; i++) {
            isPrime[i] = true;
        }

        // mark non-primes <= N using Sieve of Eratosthenes
        for (int i = 2; i*i <= N; i++) {

            // if i is prime, then mark multiples of i as nonprime
            // suffices to consider mutiples i, i+1, ..., N/i
            if (isPrime[i]) {
                for (int j = i; i*j <= N; j++) {
                    isPrime[i*j] = false;
                }
            }
        }

        // count primes
        int primes = 0;
        for (int i = 2; i <= N; i++) {
            if (isPrime[i]) primes++;
        }
        System.out.println("The number of primes <= " + N + " is " + primes);
    }
}
```