

```

import java.text.DecimalFormat;
import java.util.*;
// This program will check all the multiples of numbers until the square root of the inputted
number. The remaining numbers will be the prime numbers from 2 to the inputted number.
public class Sieve {
    public static void main(String args[]) {
        System.out.println("\nSieve of Eratosthenes\n");
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the primes upper bound ==>> ");
        final int MAX = input.nextInt();
        boolean primes[] = new boolean[MAX];
        computePrimes(primes);
        displayPrimes(primes);
    }

    public static void computePrimes(boolean primeArray[]) {
        // This method will compute the prime numbers
        for (int j = 1; j < primeArray.length; j++) {
            primeArray[j] = true;
        }
        for (int i = 2; i <= Math.sqrt(primeArray.length); i++) {
            for (int j = (i * 2) - 1; j < primeArray.length; j++) {
                if ((j + 1) % i == 0) {
                    primeArray[j] = false;
                }
            }
        }
    }

    public static void displayPrimes(boolean primeArray[]) {
        // This method will display the prime numbers

        computePrimes(primeArray);

        int result = 0;
        DecimalFormat format = new DecimalFormat("0000");
        int[] values = new int[primeArray.length];
        for (int j = 0; j < primeArray.length; j++) {
            if (primeArray[j] == true) {
                result = j + 1;
                values[j] = result;
            }
        }
    }
}

```

```
        }  
    }  
    int number = 0;  
    for (int i = 0; i < primeArray.length; i++) {  
        if (values[i] > 0) {  
            System.out.print(format.format(values[i]) + " ");  
            number++;  
        }  
        if (number % 17 == 0) {  
            number++;  
            System.out.print("\n");  
        }  
        else {  
            System.out.print("");  
        }  
    }  
}
```