```
Sieve of Eratosthenes

Please enter an upper bound:
1000

0002 0003 0005 0007 0011 0013 0017 0019 0023 0029 0031 0037 0041 0043 0047 0053
0059 0061 0067 0071 0073 0079 0083 0089 0097 0101 0103 0107 0109 0113 0127 0131
0137 0139 0149 0151 0157 0163 0167 0173 0179 0181 0191 0193 0197 0199 0211 0223
0227 0229 0233 0239 0241 0251 0257 0263 0269 0271 0277 0281 0283 0293 0307 0311
0313 0317 0331 0337 0347 0349 0353 0359 0367 0373 0379 0383 0389 0397 0401 0409
0419 0421 0431 0433 0439 0443 0449 0457 0461 0463 0467 0479 0487 0491 0499 0503
0509 0521 0523 0541 0547 0557 0563 0569 0571 0577 0587 0593 0599 0601 0607 0613
0617 0619 0631 0641 0643 0647 0653 0659 0661 0673 0677 0683 0691 0701 0709 0719
0727 0733 0739 0743 0751 0757 0761 0769 0773 0787 0797 0809 0811 0821 0823 0827
0829 0839 0853 0857 0859 0863 0877 0881 0883 0887 0907 0911 0919 0929 0937 0941
0947 0953 0967 0971 0977 0983 0991 0997
```

```
import java.util.Scanner;
import java.text.DecimalFormat;
public class Sieve {
        public static void main(String[] args){
                // This main method needs additions for the 100 point
version.
                System.out.println("\nSieve of Eratosthenes\n");
                Scanner scan= new Scanner(System.in);
                System.out.println("Please enter an upper bound: ");
                int number=scan.nextInt();

                boolean[] primes = computePrimes(number); // Upper
bound of 100
                displayPrimes(primes);
        }

        public static boolean[] computePrimes(int upperBound){
                // This method will compute the prime numbers
                boolean[] primeArray = new boolean[upperBound+1];
                for (int i=2; i<upperBound+1;i++) {
                        primeArray[i]=true;
                }
                for (int i=2; i<Math.pow(upperBound, 0.5);i++) {
                        if (primeArray[i]) {
                                for (int j=2; i*j<=upperBound;j++) {
                                        primeArray[i*j]=false;
                                }
                        }
                }
                return primeArray;
        }
```

```
        public static void displayPrimes(boolean[] primeArray){
                // This method will display the prime numbers
                DecimalFormat df= new DecimalFormat("0000");
                String primeValues=new String();
                int count=0;

                for (int i=2; i<primeArray.length;i++) {
                        if (primeArray[i]==true) {
                                if (count%16==0) {

primeValues=primeValues.concat("\n");
                                }

primeValues=primeValues.concat(df.format(i)+" ");
                                count++;
                        }
                }
                System.out.println(primeValues);
```