```
/* SELF ASSESSMENT
   1.  createSequence:
Did I use the correct method definition?
Mark out of 5:         5
Comment:        Yes, my functions were correct and did their roles respectivrly.
Did I create an array of size n (passed as the parameter) and initialise it?
Mark out of 5:         5
Comment:        I created an Array of size N + 1, as I felt it would make the code easier to understand as well as easier to code. The code
was initialised with 0s and 1s.
Did I return the correct item?
Mark out of 5:         5
Comment:        The item I returned was correct.

   2.  crossOutMultiples
Did I use the correct method definition?
Mark out of 5:         5
Comment:        I used the correct method function and the function carries out its function appropriately.
Did I ensure the parameters are not null and one of them is a valid index into the array
Mark out of 2:         2
Comment:        None of the parameters could be null as I filled them all.
Did I loop through the array using the correct multiple?
Mark out of 5:         5
Comment:        I looped through the array with the correct multiple.
Did I cross out correct items in the array that were not already crossed out?
Mark out of 3:         3
Comment:        I crossed out all items that were not already crossed out.

   3.  sieve
Did I have the correct function definition?
Mark out of 5:         5
Comment:        I used the correct method function and the function carries out its function appropriately.
Did I make calls to other methods?
Mark out of 5:         5
Comment:     I made calls to the other methods in my function.
Did I return an array with all non-prime numbers are crossed out?
Mark out of 2:         2
Comment:        I returned an array with all contantents crossed out in my function.

   4.  sequenceTostring
Did I have the correct function definition?
Mark out of 5:         5
Comment:        I used the correct method function and the function carries out its function appropriately.
Did I ensure the parameter to be used is not null?
Mark out of 3:         3
Comment:         I ensured the parameters were not null.
Did I Loop through the array updating the String variable with the non-crossed out numbers and the crossed numbers in brackets?
Mark out of 10:        10
Comment:    The function succesfully updates the array for any non-prime numbers and effectively crosses them all.

   5.  nonCrossedOutSubseqToString
Did I have the correct function definition
Mark out of 5:         5
Comment:        I used the correct method function and the function carries out its function appropriately.
Did I ensure the parameter to be used is not null?
Mark out of 3:         3
Comment:        I ensured that the parameters were not null.
Did I loop through the array updating the String variable with just the non-crossed out numbers?
Mark out of 5:         5
Comment:         The function sucsesfully updates the array for any non-prime numbers and effectively crosses them all.



   6.  main
Did I ask  the user for input n and handles input errors?
Mark out of 5:         5
Comments:        I handled numbers less than 2 appropriately.
Did I make calls to other methods (at least one)?
Mark out of 5:         5
Comment:  Yes, constatntly throughout the function.
Did I print the output as shown in the question?
Mark out of 5:         5
Comment:  Yes, the output is shown as expected.

   7.  Overall
Is my code indented correctly?
Mark out of 4:         4
Comments:        Yes, I indented appropriately as I went along.
Do my variable names make sense?
Mark out of 4:         4
Comments:        All variable names make sense and are understandable.
Do my variable names, method names and class name follow the Java coding standard
Mark out of 4:         4
Comments:        All of the above follow the coding standard.
      Total Mark out of 100 (Add all the previous marks):        100
*/

import java.util.Scanner;
```

```java
public class SieveOfEratosthenes {

        public static int[] createSequence ( )
        {

                Scanner input = new Scanner ( System.in);
                System.out.print("Enter int>=2:");
                int positiveInterger = input.nextInt();
                if ( positiveInterger < 2)
                {
                        System.out.println("What you inputed was not an appropriate answer.\nPlease try again.");
                }
                positiveInterger++;
                int[] numberArray = new int[positiveInterger];
                for (int count = 0; count < numberArray.length; count++)
                {
                        if ( count < 2)
                        {
                                numberArray[count] = 0;
                        }
                        else
                        {
                                numberArray[count] = 1;
                        }
                }
                input.close();
                return numberArray;
        }

        public static int[] crossOutHigherMultiples ( int[] numberArray, int count)
        {
                int multiplier = 2;
                int currentNumber = 0;
                if(numberArray[count] == 1)
                {
                        while ( currentNumber < numberArray.length)
                        {
                                numberArray[currentNumber] = 0;
                                currentNumber = count * multiplier++;
                        }
                }
                multiplier = 2;
                currentNumber = 0;

                return numberArray;
        }

        public static int[] sieve ( int[] numberArray)
        {

                double sqrtOfNumberArray = Math.sqrt(numberArray.length);
                for ( int count = 2; count < sqrtOfNumberArray; count++)
                {
                        numberArray = crossOutHigherMultiples( numberArray, count);
                        if ( numberArray[count] == 1)
                        {
                                String stringOfNumbers = sequenceToString ( numberArray);
                                System.out.println(stringOfNumbers);
                        }
                }
                return numberArray;
        }

        public static String sequenceToString (int[] numberArray)
        {
                String stringOfNumbers ="";
                for ( int count = 2; count < numberArray.length; count++)
                {
                        if ( count == 2)
                        {
                                stringOfNumbers += count;
                        }
                        else
                        {
                                if ( numberArray[count] == 1)
                                {
                                        stringOfNumbers += ", "+count;
                                }
                                else
                                {
                                        stringOfNumbers += ", ["+count+"]";
                                }
                        }
                }
                return stringOfNumbers;
        }
        public static String nonCrossedOutSubseqToString (int[] numberArray)
        {
                String nonCrossedOutSubseqToString = "";
                for (int count = 2; count < numberArray.length ; count++)
                {
                        if ( numberArray[count] == 1)
                        {
                                if ( count == 2)
                                {
                                        nonCrossedOutSubseqToString += count;
                                }
                                else
                                {
                                        nonCrossedOutSubseqToString += (", "+count);
                                }
                        }
                }
                return nonCrossedOutSubseqToString;
        }

        public static void main(String[] args) {

                Scanner input = new Scanner ( System.in );
                int[] numberArray = createSequence ();
                String stringOfNumbers = sequenceToString ( numberArray);
                System.out.println(stringOfNumbers);
                if ( input.hasNextInt())
                {

                        System.out.println("What you inputed was not an appropriate answer.\nPlease try again.");
                }
                else
                {
                        numberArray = sieve (numberArray);
                        String nonCrossedOutSubseqToString = nonCrossedOutSubseqToString ( numberArray);
                        System.out.println(nonCrossedOutSubseqToString);
                }
                input.close();
        }

}
```