```
/**
 * Gets a command-line argument (int), and prints all the divisors of the given
 * number.
 */
public class Divisors {
    public static void main(String[] args) {
        // Put your code here
        int x = Integer.parseInt(args[0]);
        for (int i = 1; i <= x; i++) {
            if (x % i == 0) {
                System.out.println(i);
            }
        }
    }
}
```

```
/**
 * Prints a given string, backward. Then prints the middle character in the
 * string.
 * The program expects to get one command-line argument: A string.
 */
public class Reverse {
    public static void main(String[] args) {
        // Put your code here
        String s = args[0];
        String sOut = "";
        for (int i = s.length() - 1; i >= 0; i--) {
            sOut += s.charAt(i);
        }
        System.out.println(sOut);
        System.out.println("The middle character is " + (s.charAt(s.length() / 2)));
    }
}
```

```
/**
 * Generates and prints random integers in the range [0,10),
 * as long as they form a non-decreasing sequence.
 */
public class InOrder {
    public static void main(String[] args) {
        // Write your code here
        int rand = (int) (Math.random() * 10);
        int prev = 0;
        while (rand >= prev) {
            System.out.print(rand + " ");
            prev = rand;
            rand = (int) (Math.random() * 10);
        }
    }
}
```

```
/**
 * Gets a command-line argument n (int), and prints an n-by-n damka board.
 */
public class DamkaBoard {
    public static void main(String[] args) {
        // Put your code here
        int n = Integer.parseInt(args[0]);
        for (int i = 0; i < n; i++) {
            if (i % 2 == 1) {
                System.out.print(" ");
            }
            for (int j = 0; j < n; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

```
/**
 * Gets a command-line argument (int), and chekcs if the given number is
 * perfect.
 */
public class Perfect {
    public static void main(String[] args) {
        // Put your code here
        int n = Integer.parseInt(args[0]);
        int sum = 1;
        String out = n + " is a perfect number since " + n + " = 1";
        for (int i = 2; i < n; i++) {
            if (n % i == 0) {
                sum += i;
                out += " + " + i;
            }
        }
        System.out.println(n == sum ? out : n + " is not a perfect number");
    }
}
```

```
import java.util.Random;

public class OneOfEachStats {
   public static void main(String[] args) {
      // Gets the two command-line arguments
      int T = Integer.parseInt(args[0]);
      int seed = Integer.parseInt(args[1]);
      // Initailizes a random numbers generator with the given seed value
      Random generator = new Random(seed);

      double average = 0;
      int children2 = 0;
      int children3 = 0;
      int children4plus = 0;

      for (int i = 0; i < T; i++) {
         int sum = 0;
         boolean hasBoy = false;
         boolean hasGirl = false;
         while (!hasBoy || !hasGirl) {
            sum++;
            if (generator.nextDouble() < 0.5) {
               hasBoy = true;
            } else {
               hasGirl = true;
            }
         }

         average += sum;
         switch (sum) {
            case 2:
               children2++;
               break;
            case 3:
               children3++;
               break;
            default:
               children4plus++;
         }
      }
      average /= T;
      System.out.println("Average: " + average + " children to get at least one of each
gender.");
      System.out.println("Number of families with 2 children: " + children2);
      System.out.println("Number of families with 3 children: " + children3);
      System.out.println("Number of families with 4 or more children: " + children4plus);
      int max = (int) Math.max(Math.max(children2, children3), children4plus);
      System.out.println("The most common number of children is "
            + (children2 == max ? "2." : children3 == max ? "3." : "4 or more children."));

   }
}
```