```
/**
 *  Gets a command-line argument (int), and prints all the divisors of the given number.
 */
public class Divisors {
    public static void main (String[] args) {
        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){
        String word = args[0];
        char middle = ' ';
        for ( int i = 0; i<word.length(); i++){
            System.out.print(word.charAt(word.length() - i - 1));
        }
        middle = word.charAt(((word.length() + 1) / 2) - 1);
        System.out.println();
        System.out.print("The middle character is " + middle);
    }
}
```

```
/**
 *  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) {
        int current = 0;
        int next = (int)(10*Math.random());
        do {System.out.println(next);
            current = next;
            next = (int)(10*Math.random());
        } while (next>=current);

    }
}
```

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

        if (sum == x) {
            System.out.print(x + " is a perfect number since " + x + " = ");
            for (int i = 1; i <= x/2 ; i++){
                if (x % i == 0){
                    System.out.print(i);
                    if (x/2 != i)
                        System.out.print(" + ");
                }
            }
        }else{
            System.out.println(x + " is not a perfect number");
        }
    }
}
```

```
import java.util.Random;
/**
 *  Computes some statistics about families in which the parents decide
 *  to have children until they have at least one child of each gender.
 *  The program expects to get two command-line arguments: an int value
 *  that determines how many families to simulate, and an int value
 *  that serves as the seed of the random numbers generated by the program.
 *  Example usage: % java OneOfEachStats 1000 1
 */
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 avarage = 0;
      int familiesof2, familiesof3, familiesof4ormore;
      familiesof2 = familiesof3 = familiesof4ormore = 0;
      for (int i = 0; i < T; i++) {
         String gender = "";
         String newborn = "";
         int babycounter = 1;
         double borg = generator.nextDouble(); // boy or girl
         if (borg < 0.5)
            gender = "b ";
         else
            gender = "g ";
         do {babycounter++;
            borg = generator.nextDouble();
```

```
            if (borg < 0.5)
                newborn = "b ";
            else
                newborn = "g ";
        } while (gender == newborn);
        avarage +=babycounter;
        switch (babycounter) {
            case 2:
                familiesof2++;
                break;
            case 3:
                familiesof3++;
                break;
            default:
                familiesof4ormore++;
                break;
        }
    }
    avarage = avarage/T;
    System.out.println("Average: " + avarage + " children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: " + familiesof2);
    System.out.println("Number of families with 3 children: " + familiesof3);
    System.out.println("Number of families with 4 or more children: " + familiesof4ormore);
    if (familiesof2 == Math.max(familiesof4ormore, Math.max(familiesof2, familiesof3))) {
        System.out.println("The most common number of children is 2.");
    }else if (familiesof3 == Math.max(familiesof4ormore, Math.max(familiesof2, familiesof3))) {
        System.out.println("The most common number of children is 3.");
    }else {
```

```
            System.out.println("The most common number of children is 4 or more.");
        }
        //// In the previous version of this program, you used a statement like:
        //// double rnd = Math.random();
        //// Where "rnd" is the variable that stores the generated random value.
        //// In this version of the program, replace this statement with:
        //// double rnd = generator.nextDouble();
        //// This statement will generate a random value in the range [0,1),
        //// just like you had in the previous version, except that the
        //// randomization will be based on the given seed.
        //// This is the only change that you have to do in the program.

    }
}
```