```java
/**
 *  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/2 + 1; i++) {
            if (x % i == 0) {
                System.out.println(i);
            }
        }
        System.out.println(x);
    }
}
```

```
/**
 * 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 s = args[0];
          char mid = ' ';
          if (s.length() % 2 == 0){
               mid = s.charAt((s.length() / 2) - 1);
          }
          else {
               mid = s.charAt(s.length() / 2);
          }
          String rvrs = "";
          for (int i = s.length() - 1; i >= 0; i--) {
               rvrs = rvrs + s.charAt(i);
          }
          System.out.println(rvrs);
          System.out.println("The middle character is " + mid);
     }
}
```

```java
/**
 *  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 r1 = (int) ((Math.random() * 10));
          System.out.print(r1);
          int r2 = (int) ((Math.random() * 10));
          while (r1 <= r2){
               System.out.print(" " + r2);
               r1 = r2;
               r2 = (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) {
        int x = Integer.parseInt(args[0]);
        boolean t = true;
        for (int j = 0; j < x; j++){
            System.out.println("");
            for (int i = 0; i < x; i++){
                if (t){
                    System.out.print("* ");
                }
                else {
                    System.out.print(" *");
                }
            }
            t = !t;
        }
    }
}
```

```
/**
 *  Gets a command-line argument (int), and chekcs if the given number
is perfect.
 */
public class Perfect {
    public static void main (String[] args) {
        int p = Integer.parseInt(args[0]);
        int sum = 1;
        String s = p + " is a perfect number since " + p + " = 1";
        for (int i = 2; i <= p/2 + 1; i++) {
            if (p % i == 0) {
                sum = sum + i;
                s = s + " + " + Integer.toString(i);
            }
        }
        if (p == sum){
            System.out.println(s);
        }
        else {
            System.out.println(p + " 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]);
        double c = 0.0;
        int famsum = 0;
```

```
           // Initailizes a random numbers generator with the given
seed value
        Random generator = new Random(seed);
           int count = t;
           int f2 = 0;
           int f3 = 0;
           int f4 = 0;
           while (count != 0) {
                boolean b = false;
                boolean g = false;
                int sum = 0;
                while (!b || !g){
                      sum += 1;
                      c = generator.nextDouble();
                      if (c >= 0.5){
                           g = true;
                      }
                      else if (c < 0.5){
                           b = true;
                      }
                }
                if (sum == 2){
                      f2 += 1;
                }
                else if (sum == 3){
                      f3 += 1;
                }
                else {
```

```
                f4 +=1;
            }
            famsum += sum;
            count -= 1;

        }
        double avg = famsum / (double) t;
        String mode = "4 or more";
        if (f2 >= f3 && f2 >= f4){
            mode = "2";
        }
        else if (f3 >= f4){
            mode = "3";
        }
        System.out.println("Average: " + avg + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + f2);
        System.out.println("Number of families with 3 children: " + f3);
        System.out.println("Number of families with 4 or more children: " + f4);
        System.out.println("The most common number of children is " + mode + ".");


        //// 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.


     }
}
```