```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 dividend = Integer.parseInt((args[0]));
        int i = 1;
        while (i <= dividend) {
            if(dividend % i == 0) {
                System.out.println(i);
            }
            i++;
        }
    }
}
```

```java
/**
 * 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 strOrig = args[0];
        String strRevrs = "";
        for (int i = strOrig.length() - 1; i >= 0; i--) {
            strRevrs += strOrig.charAt(i);
        }
        System.out.println(strRevrs);
        char midChar;
        if (strOrig.length() % 2 == 0) {
            midChar = strOrig.charAt(strOrig.length() / 2 -
1);
        }
        else {
            midChar = strOrig.charAt(strOrig.length() / 2);
        }

        System.out.println("The middle character is " +
midChar);
    }
}
```

```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 firstRand = (int)(Math.random() * 10);
        System.out.print(firstRand);
        int prev = firstRand;
        int current;
        do {
            current = (int)(Math.random() * 10);
            if (current >= prev) {
            System.out.print(" " + current);
            prev = current;
            }
            else {
             break;
        }
    } while (current >= prev);
   }
  }
```

```java
/**
 *  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 boardSize = Integer.parseInt(args[0]);
        int i = 0;
        while (boardSize > i) {
            if ((i + 1) % 2 == 0) {
                System.out.print(" *");
            }
            else {
                System.out.print("* ");
            }
            int j = 1;
            while (boardSize > j) {
                if ((i + 1) % 2 == 0) {
                    System.out.print(" *");
                }
                else {
                    System.out.print("* ");
                }
                j++;
            }
            System.out.println();
            i++;
        }
    }
}
```

```java
/**
 *  Gets a command-line argument (int), and chekcs if the
given number is perfect.
 */
public class Perfect {
    public static void main (String[] args) {
        int numToCheck = Integer.parseInt(args[0]);
        int i = 2;
        int sumDivisors = 1;
        String strDivisorSum = "1";
        while (numToCheck > i) {
            if (numToCheck % i == 0) {
                sumDivisors += i;
                strDivisorSum += " + " + i;
            }
            i++;
        }
        if (numToCheck == sumDivisors) {
            System.out.println(numToCheck + " is a perfect
number since " + numToCheck + " = " + strDivisorSum);
        }
        else {
            System.out.println(numToCheck + " is not a perfect
number");
        }
    }
}
```

```java
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 numOfFam = 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);
        int sumTotal = 0;
        int sumTwo = 0;
        int sumThree = 0;
        int sumFourOrMore = 0;
        for (int i = 0; i < numOfFam; i++) {
            int sumOfOneFamily = 0;
            int boys = 0;
            int girls = 0;
            while (boys == 0 || girls == 0){
             double kid = generator.nextDouble();
             if (kid > 0.5) {
             boys++;
             }
             else {
             girls++;
             }
             sumOfOneFamily ++;
            }
            sumTotal += sumOfOneFamily;
            if (sumOfOneFamily == 2) {
             sumTwo++;
            }
            else if (sumOfOneFamily == 3) {
             sumThree++;
            }
            else {
```

```
                sumFourOrMore++;
            }
        }
        double average = (double) sumTotal / numOfFam;
        System.out.println("Average: " + average + " children
to get at least one of each gender.");
        System.out.println("Number of families with 2
children: " + sumTwo);
        System.out.println("Number of families with 3
children: " + sumThree);
        System.out.println("Number of families with 4 or more
children: " + sumFourOrMore );
        if (sumTwo > sumThree && sumTwo > sumFourOrMore ) {
        System.out.println("The most common number of children
is 2.");
        }
        else if (sumThree > sumTwo && sumThree >
sumFourOrMore) {
        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.

    }
}
```