

```
/**
 * Gets a command-line argument (int), and prints all the divisors of the given number.
 */
public class Divisors {
    public static void main (String[] s) {
        int n = Integer.parseInt(s[0]);
        for (int i=1; i<=n; i++) {
            if (n%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[] asdf){
        String sIn = asdf[0];
        String sOut = "";
        char mid = ' ';
        for (int i=(sIn.length()-1); i>=0; i--) {
            sOut = sOut + sIn.charAt(i);
            if (i==(sIn.length()-1)/2) {
                mid = sIn.charAt(i);
            }
        }
        System.out.println(sOut);
        System.out.println("The middle character is " + mid);
    }
}
```

```
/**
 * 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[] asdf) {
        int next = 0, prev;
        do {
            prev = next;
            next = (int)(Math.random()*10);
            if (next>=prev) {System.out.print(next + " ");}
        } while (next>prev); // >= / > doesn't matter, simply skips a cycle
    }
}
```

```
/**
 * Gets a command-line argument n (int), and prints an n-by-n damka board.
 */
public class DamkaBoard {
    public static void main(String[] asdf) {
        int n = Integer.parseInt(asdf[0]);
        for (int y=0; y<n; y++) {
            for (int x=0; x<n; x++) {
                if (y%2==0) {System.out.print("* ");} else {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[] s) {
        int n = Integer.parseInt(s[0]);
        int divSum = 0;
        String div = "";
        for (int i=2; i<=n; i++) {
            if (n%i==0 && n!=i) {
                divSum += i;
                div += " + " + i;
            }
        }
        if (n == divSum+1) {
            System.out.println(n + " is a perfect number since " + n + " = 1" + div);
        } else {
            System.out.println(n+ " 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[] asdf) {
        int n = Integer.parseInt(asdf[0]);
        int seed = Integer.parseInt(asdf[1]);
        Random generator = new Random(seed);
        // All Experiments
        float allKids = 0;
        int kids2 = 0;
        int kids3 = 0;
        int kids4 = 0;
        for (int i=0; i<n; i++) {
            // One Experiment
            int kids = 0;
            boolean isBoy = false;
            boolean isGirl = false;
            do {
                allKids++;
                kids++;
                if ((int)(generator.nextDouble()*(2)) == 1) {
                    isBoy=true;
                } else {
                    isGirl=true;

```

```

    }
    } while (!isBoy || !isGirl);
    switch (kids) {
        case 2: kids2++;break;
        case 3: kids3++;break;
        case 4: kids4++;break;
        default: kids4++;break;
    }
}

// Results

System.out.println("Average: " + allKids/n + " children to get at least one of each
gender.");

System.out.println("Number of families with 2 children: " + kids2);
System.out.println("Number of families with 3 children: " + kids3);
System.out.println("Number of families with 4 or more children: " + kids4);
int maxKids = Math.max(kids2, Math.max(kids3, kids4));
String mode;
if (maxKids == kids2) {
    mode = "2";
} else if (maxKids == kids3) {
    mode = "3";
} else {
    mode = "4 or more";
}
System.out.println("The most common number of children is " + mode + ".");

}
}

```