

```
/**
 * 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 number = Integer.parseInt(args[0]);
        for(int i= 1 ; i<= number ; i++){
            if(number%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 value = args[0];
        int middle = ((value.length() % 2) == 0) ? ((value.length() / 2)) : (int)
(double) (value.length() / 2);
        String reversed = "";
        for(int i = value.length()-1; i >= 0 ; i--){
            System.out.print(value.charAt(i));
            reversed += value.charAt(i);
        }
        System.out.println();
        System.out.print("The middle character is " + reversed.charAt(middle));
    }
}

```

```

/**
 * Generates and prints random integers in the range [0,10),
 * as long as they form a non-decreasing sequence.
 */
import java.util.Random;
public class InOrder {
    public static void main (String[] args) {
        Random rand = new Random();
        int randomNumber = rand.nextInt(10);
        System.out.print(randomNumber + " ");
        int numberBefore = randomNumber;
        do {
            int newRandom = rand.nextInt(10);
            if(numberBefore <= newRandom){
                System.out.print(newRandom + " ");
                numberBefore=newRandom;
            }
            else{
                break;
            }
        }
        while (true);
    }
}

```

```

/**
 * 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 numberOrLinesAndColumns = Integer.parseInt(args[0]);
        for (int line= 1; line<=numberOrLinesAndColumns ; line++){
            String currentLine ="";
            for(int column = 1;column<=numberOrLinesAndColumns;column++ ){
                if(line%2 == 0 && column == 1){
                    currentLine += " *";
                }
                else if (line%2 != 0 && column == 1) {
                    currentLine += " ";
                }
                else{
                    currentLine += " *";
                }
                if(line%2 == 1 && column == numberOrLinesAndColumns){
                    currentLine += " ";
                }
            }
            System.out.print(currentLine);
            if(line != numberOrLinesAndColumns){
                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 inputNumber = Integer.parseInt(args[0]);
        String text = inputNumber
            + " is a perfect number since " + inputNumber + " = 1";
        int sum = 1;
        for(int i = 2; i<inputNumber-1 ; i++){
            if(inputNumber%i == 0 ){
                text += " + " + i;
                sum += i;
            }
        }
        if(sum == inputNumber){
            System.out.print(text);
        }
        else {
            System.out.print(inputNumber+" is not a perfect number");
        }
    }
}

```

```

import java.util.Random;

/**
 * Simulates the formation of a family in which the parents decide
 * to have children until they have at least one child of each gender.
 */
public class OneOfEach {
    public static void main (String[] args) {
        Random rand = new Random();
        //less than 0.5 is boy more or equal than 0.5 is a girl
        int childrenCount=0;
        boolean boy = false;
        boolean girl = false;
        while (!boy || !girl){
            childrenCount++;
            double randomNumber = rand.nextDouble(1);
            if(randomNumber < 0.5){
                System.out.print("b ");
                boy = true;
            }
            else{
                System.out.print("g ");
                girl = true;
            }
        }
        System.out.println();
        System.out.printf("You made it... and you now have %d children" ,
childrenCount);
    }
}

```

```
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 one command-line argument: an int value  
 * that determines how many families to simulate.
```

```
*/
```

```
public class OneOfEachStats1 {  
    public static void main (String[] args) {  
        int numberOfTimesToRun = Integer.parseInt(args[0]);  
        int twoChildrenFamilies = 0;  
        int threeChildrenFamilies = 0;  
        int fourOrMoreChildrenFamilies = 0;  
        int sumOfAllChildren = 0;  
        String mostCommon = "";  
        Random rand = new Random();  
        for (int i = 0; i < numberOfTimesToRun ; i++){  
            int childrenCount=0;  
            boolean boy = false;  
            boolean girl = false;  
            while (!boy || !girl){  
                childrenCount++;  
                double randomNumber = rand.nextDouble(1);  
                if(randomNumber < 0.5){  
                    boy = true;  
                }  
                else{  
                    girl = true;  
                }  
            }  
            if(childrenCount == 2){  
                twoChildrenFamilies++;  
            }  
        }  
    }  
}
```

```

        else if(childrenCount == 3){
            threeChildrenFamilies++;
        }
        else{
            fourOrMoreChildrenFamilies++;
        }
        sumOfAllChildren += childrenCount;
        if(twoChildrenFamilies > threeChildrenFamilies && twoChildrenFamilies
> fourOrMoreChildrenFamilies){
            mostCommon = "2";
        }
        else if(threeChildrenFamilies > twoChildrenFamilies &&
threeChildrenFamilies > fourOrMoreChildrenFamilies){
            mostCommon = "3";
        }
        else if(fourOrMoreChildrenFamilies > twoChildrenFamilies &&
fourOrMoreChildrenFamilies > threeChildrenFamilies){
            mostCommon = "4 or more";
        }
    }
    var average = (double)sumOfAllChildren/numberOfTimesToRun;
    System.out.println("Average: " + average + " children to get at least one
of each gender.");
    System.out.println("Number of families with 2 children: " +
twoChildrenFamilies);
    System.out.println("Number of families with 3 children: " +
threeChildrenFamilies);
    System.out.println("Number of families with 4 or more children: " +
fourOrMoreChildrenFamilies);
    System.out.println("The most common number of children is " +
mostCommon + ".");
}
}

```



```

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]);
        // Initializes a random numbers generator with the given seed value
        Random rand = new Random(seed);
        //int numberOfTimesToRun = Integer.parseInt(args[0]);
        int twoChildrenFamilies = 0;
        int threeChildrenFamilies = 0;
        int fourOrMoreChildrenFamilies = 0;
        int sumOfAllChildren = 0;
        String mostCommon = "";
        for (int i = 0; i < T ; i++){
            int childrenCount=0;
            boolean boy = false;
            boolean girl = false;
            while (!boy || !girl){
                childrenCount++;
                double randomNumber = rand.nextDouble();
                if(randomNumber < 0.5){
                    boy = true;
                }
                else{
                    girl = true;
                }
            }
        }
    }
}

```

```

    }
}
if(childrenCount == 2){
    twoChildrenFamilies++;
}
else if(childrenCount == 3){
    threeChildrenFamilies++;
}
else{
    fourOrMoreChildrenFamilies++;
}
sumOfAllChildren += childrenCount;
if((twoChildrenFamilies > threeChildrenFamilies && twoChildrenFamilies
> fourOrMoreChildrenFamilies){
    mostCommon = "2";
}
else if((threeChildrenFamilies > twoChildrenFamilies &&
threeChildrenFamilies > fourOrMoreChildrenFamilies){
    mostCommon = "3";
}
else if((fourOrMoreChildrenFamilies > twoChildrenFamilies &&
fourOrMoreChildrenFamilies > threeChildrenFamilies){
    mostCommon = "4 or more";
}
}

var average = (double)sumOfAllChildren/T;
System.out.println("Average: " + average + " children to get at least one
of each gender.");
System.out.println("Number of families with 2 children: " +
twoChildrenFamilies);
System.out.println("Number of families with 3 children: " +
threeChildrenFamilies);
System.out.println("Number of families with 4 or more children: " +

```

```
fourOrMoreChildrenFamilies);
```

```
    System.out.println("The most common number of children is " +  
mostCommon + ".");
```

```
    /// 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.
```

```
    }
```

```
}
```