```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 divisor = Integer.parseInt(args[0]);
        int i=1;
        while (i<=divisor){
            if (divisor % 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 str = args[0];
                String backward = "";
                int i = 0;
                while (i<str.length()){
                        backward = backward + str.charAt(str.length()-i -1);
                        i++;
                }
                System.out.println(backward);
                System.out.println("The middle character is "+backward.charAt(str.length()/2));
        }
}
```

```java
/**
 * 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 randomnum = rand.nextInt(10);
                System.out.print(randomnum+"");
                int newrandom = rand.nextInt(10);
                while(newrandom>=randomnum){
                        System.out.print(" "+newrandom+"");
                        randomnum = newrandom;
                        newrandom = rand.nextInt(10);
                }
                System.out.println("");
        }
}
```

```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 perfnum = Integer.parseInt(args[0]);
                int sumdividors = 0;
                for(int i=1; i<perfnum; i++){
                        if(perfnum % i == 0){
                                sumdividors = sumdividors + i;
                        }
                }
                if(sumdividors == perfnum){
                        System.out.print(perfnum+" is a perfect number since "+perfnum+" =");
                        for (int i=1; i<perfnum; i++){
                                if(perfnum % i == 0){
                                        if(i == 1){
                                                System.out.print(" "+i);
                                        }
                                        else{
                                                System.out.print(" + "+i);
                                        }
                                }
                        }
                        System.out.println("");
                }
                else{
                        System.out.println(perfnum +" is not a perfect number");
                }
        }
}
```

```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 damkarow = Integer.parseInt(args[0]);
                for(int i=1; i<=damkarow; i++){
                        for(int j=1; j<=damkarow; j++){
                                if(i%2==1){
                                        if(i==1){
                                                System.out.print("* ");
                                        }
                                        else{
                                                System.out.print("* ");
                                        }
                                }
                                else{
                                        System.out.print(" *");
                                }
                        }
                        System.out.println("");
                }
        }
}
```

```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 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 rand =0;
                boolean boy = false;
                boolean girl = false;
                int count = 0;
                int fam2 = 0, fam3 = 0, fam4 =0;
                double averg = 0;
                int countT = 0;
                while(T != countT){
                        boy = false;
                        girl = false;
                        count = 0;
                        while(boy != true || girl != true){
                                rand = generator.nextDouble();
                                if(rand<0.5){
                                        boy =true;
                                }
                                else{
                                        girl = true;
                                }
                                count++;
                        }
                        if(count == 2){
                                fam2++;
                        }
                        else if(count == 3){
                                fam3++;
                        }
                        else{
```

```java
                    fam4++;
            }
            averg = averg + count;
            countT++;
        }
        averg = averg / T;
        System.out.println("Average: "+ averg+" children to get at least one of each
gender.");

        System.out.println("Number of families with 2 children: "+fam2);
        System.out.println("Number of families with 3 children: "+fam3);
        System.out.println("Number of families with 4 or more children: "+fam4);

        if(fam2 > fam3 && fam2>fam4){
            System.out.println("The most common number of children is 2.");
        }
        else if(fam3 > fam2 && fam3 > fam4){
            System.out.println("The most common number of children is 3.");
        }
        else{
            System.out.println("The most common number of children is 4.");

        }

    }
}
```