```
/**
 *  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 input = Integer.parseInt(args[0]);
		// checks if i is a divisor of input until reaches to value of input *
0.5
		// no need to check further than 0.5 * input because its the
largest integer possible to divide
		for (int i = 1; i <= 0.5 * input; i++)
		{
			if (input % i == 0)
			{
				System.out.println(i);
			}
		}
		System.out.println(input);


	}
}
```

```
/**
 * 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 inputS = args[0];
                String newS = "";
                char middleChar;
                int lengthOfString = inputS.length();
                if (lengthOfString % 2 == 1){
                        middleChar = inputS.charAt((lengthOfString - 1) / 2 );
                } else {
                        middleChar = inputS.charAt(lengthOfString / 2 - 1);
                }
                for (int n = lengthOfString - 1; n >= 0; n--)
                {
                        newS += inputS.charAt(n);
                }
                System.out.println(newS);
                System.out.println("The middle character is " + middleChar);


        }
}
```

```
/**
 *  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 currentNum = (int)(Math.random() * 10);
                int newNum = currentNum;
                do {
                        currentNum = newNum;
                        System.out.println(currentNum);
                        newNum = (int)(Math.random() * 10);

                }
                while (newNum >= currentNum);

        }
}
```

```
/**
 *  Gets a command-line argument (int), and chekcs if the given number is
perfect.
 */
public class Perfect {
        public static void main (String[] args) {
                int inputNum = Integer.parseInt(args[0]);
                String s = inputNum + " is a perfect number since " + inputNum
+ " = 1";
                int sum = 1;
                // checks for all inputNum divisors and sums them
                for (int i = 2; i <= 0.5 * inputNum; i++)
                {
                        if (inputNum % i == 0)
                        {
                                sum += i;
                                s += " + " + i;
                        }
                }
                // if its a perfect number
                if (sum == inputNum)
                {
                        System.out.println(s);
                }
                else{
                        System.out.println(inputNum + " is not a perfect number");
                }


        }
}
```

```
/**
 *  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 n = Integer.parseInt(args[0]);
                for (int i = 0; i < n; i++) {
                        //if the row is odd
         if (i % 2 == 1) {
            System.out.print(" ");
         }
                        // if its the last * in the row
                        for (int j = 0; j < n; j++) {
                                if (j == n-1 && i % 2 == 1) {
                                        System.out.print("*");
                                }
                                else {
                                        System.out.print("* ");
                                }
         }
         System.out.println("");
      }
   }
}
```

```
import java.util.Random;
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);
                int overallCount = 0;
                int twoChildrenCount = 0;
                int threeChildrenCount = 0;
                int fourOrMore = 0;
                int tempCount = 0;
                for (int i = 0; i < T; i++)
                {
                        boolean isBoy = false;
                        boolean isGirl = false;
                        tempCount = 0;
                        while (!isBoy || !isGirl)
                        { // while there is no either boy and girl
                                double boyOrGirl = generator.nextDouble();
                                tempCount += 1;// current number of children
                                if (boyOrGirl < 0.5){
                                        isBoy = true;
                                }
                                else{
                                        isGirl = true;
                                }
                        }
                        //adds 1 according to the size of family
                        overallCount += tempCount;

                        if (tempCount == 2){

                                twoChildrenCount += 1;

                        }

                        else if (tempCount == 3){

                                threeChildrenCount += 1;

                        }

                        else if (tempCount >= 4){

                                fourOrMore += 1;

                        }

                }

                String mostCommon = "";

                // check which group is the largest

                if ((twoChildrenCount >= threeChildrenCount) &&
(twoChildrenCount >= fourOrMore)){

                        mostCommon = "2.";

                }

                else if (threeChildrenCount >= fourOrMore){

                        mostCommon =  "3.";

                }

                else{

                        mostCommon = "4 or more.";

                }

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

                System.out.println("Number of families with 2 children: " +
twoChildrenCount);

                System.out.println("Number of families with 3 children: " +
threeChildrenCount);

                System.out.println("Number of families with 4 or more children: "
+ fourOrMore);
```

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