/\*\*

 \*  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 num = Integer.parseInt(args[0]);

        // loop which prints the divisors of num

        for (int i = 1; i < num + 1; i++ ){

            if (num % 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 word = args[0];

        String reverse\_Order = "";

        // creating a new string with reverse\_Order

        for (int i = word.length() - 1; i >= 0; i-- ){

            reverse\_Order = reverse\_Order + word.charAt(i);

        }

        System.out.println(reverse\_Order);

        if (word.length() % 2 == 0) {

            char middle = word.charAt(word.length() / 2);

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

        }

        else {

            char middle = word.charAt(((word.length() + 1) / 2) -1);

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

        }

    }

}

/\*\*

 \*  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 num1 = (int) (Math.random() \* 10);

        //convert int to string

        String answer = Integer.toString(num1);

        // check if there will be another numbers in the program

        int random = (int) (Math.random() \* 10);

        boolean non\_decreasing = false;

        int num2 = (int) (Math.random() \* 10);

        while (random < 5 ||  non\_decreasing == false){

            if (num2 == num1 - 1){

                non\_decreasing = true;

            }

            else {

                // add number that meet the conditions

                answer += " " + num2;

                num1 = num2;

            }

            // create another number between 0-10

            num2 = (int) (Math.random() \* 10);

            // random a number and check if the while loop will continue to work

            random = (int) (Math.random() \* 10);

        }

        System.out.println(answer);

    }

}

/\*\*

 \*  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 num = Integer.parseInt(args[0]);

        for(int i1 = 0; i1 < num; i1++){

            for(int i2 = 0; i2 < num; i2++){

                if(i1 % 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[] args) {

        int num = Integer.parseInt(args[0]);

        String print = num + " is a perfect number since " + num + " = 1" ;

        // calculate the sum of the divisors

        int sum = 1;

        // every number divide by one, so we start counting from two and so on...

        for (int i = 2; i < num; i++){

            if (num % i == 0){

                sum = sum + i;

                print = print + " + " + i;

            }

        }

        // if the sum of the divisors is equal to the number - print perfect, else print not perfect

        if(sum == num){

            System.out.println(print);

        }

        else{

            System.out.println(num + " 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[] 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 twoChildren = 0;

        int threeChildren = 0;

        int fourOrMoreChildren = 0;

        // count the number of born children

        int count = 0;

        for(int i = 0; i < T; i++){

            boolean boy = false;

            boolean girl = false;

            int numOfChildren = 0;

            while (girl == false || boy == false) {

                // random a number and decides weather its a boy or a girl

                double random\_human = generator.nextDouble();

                // add human that were born

                count += 1;

                if (random\_human >= 0.5){

                    boy = true;

                    numOfChildren += 1;

                }

                else{

                    girl = true;

                    numOfChildren+= 1;

                }

            }

            // add 1 to the correct group of children

            if(numOfChildren == 2){

                twoChildren += 1;

            }

            else if (numOfChildren == 3){

                threeChildren += 1;

            }

            else{

                fourOrMoreChildren += 1;

            }

        }

        // print the average children in family

        double average = count / T;

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

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

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

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

        // find  what is the most commom number of children in familys and prints them

        int max = (Math.max(twoChildren, threeChildren));

        max = Math.max(max, fourOrMoreChildren);

        if(max == twoChildren){

            System.out.println("The most common number of children is 2.");

        }

        else if(max == threeChildren){

            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");

        }

    }

}