DamkaBoard

```
/**
 *  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]);
		int k =1;
		for(int i =0; i<n; i++)
		{
			if(k%2!=0)
			{
				for(int j=0; j<n; j++)
				{
				System.out.print("* ");
				}
			}
			else
			{
				for(int j=0; j<n; j++)
				{
				System.out.print(" *");
				}
			}
			System.out.println("");
			k++;
		}
	}
}
```

Divisors

```
/**
 *  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 x = Integer.parseInt(args[0]);
		for(int i =1; i < x; i++)
		{
			if(x%i == 0)
			{
				System.out.println(i);
			}
		}
		System.out.println(x);
	}
}
```

InOrder

```
/**
 *  Generates and prints random integers in the range [0,10),
 *  as long as they form a non-decreasing sequence.
 */
import java.util.concurrent.ThreadLocalRandom;
public class InOrder {
	public static void main (String[] args) {
		boolean check = true;
		int lastNum = ThreadLocalRandom.current().nextInt(0, 10);//getting a
random number within the range
		int newNum=0;
		System.out.print(lastNum);//print of the first random number
		while(check)
		{
			newNum = ThreadLocalRandom.current().nextInt(0, 10);
			if(newNum >= lastNum)
			{
				lastNum = newNum;
				System.out.print(" "+lastNum);
			}
			else{
				check = false;
			}
		}
		System.out.println();
	}
}
```

OneOfEach

```
/**
 *  Simulates the formation of a family in which the parents decide
 *  to have children until they have at least one child of each gender.
 */
import java.util.concurrent.ThreadLocalRandom;
public class OneOfEach {
	public static void main (String[] args) {
		int boyGirl = ThreadLocalRandom.current().nextInt(0, 2);
		boolean check = true;
		boolean boy = false;
		boolean girl = false;
		int kidCounter = 0;
		while(check)
		{
			if(boyGirl == 0)
			{
				System.out.print("g");
				girl = true;
				kidCounter++;
			}
			if(boyGirl == 1)
			{
				System.out.print("b");
				boy = true;
				kidCounter++;
			}
			if(boy && girl)//exit if you have both
			{
				check = false;
			}
			boyGirl = ThreadLocalRandom.current().nextInt(0, 2);
		}
		System.out.println("");
		System.out.println("You made it... and you now have "+kidCounter+"
children.");
	}
}
```

OneOFEachStat1

```
/**
 *  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.
 */
import java.util.concurrent.ThreadLocalRandom;
public class OneOfEachStats1 {
        public static void main (String[] args) {
                int boyGirl = ThreadLocalRandom.current().nextInt(0, 2);
                boolean check = true;
                boolean boy = false;
                boolean girl = false;
                int kidCounter = 0;
                //variable above from last code
                double avgSum =0;
                int twoKid=0;
                int threeKid=0;
                int fourOrMore=0;
                int T = Integer.parseInt(args[0]);
                for(int i=0; i<T; i++)//simulates familys T times
                {
                        while(check)//family simulation
                        {
                                kidCounter++;
                                if(boyGirl == 0)
                                {
                                        girl = true;
                                }
                                if(boyGirl == 1)
                                {
                                        boy = true;
                                }
                                if(boy && girl)//exit if you have both
                                {
                                        check = false;
                                }
                                boyGirl = ThreadLocalRandom.current().nextInt(0, 2);
                        }
                        avgSum = avgSum + kidCounter;
                        if(kidCounter == 2)//check how many kids it took
                        {
                                twoKid++;
```

```
                        }
                        else if(kidCounter == 3)
                        {
                                threeKid++;
                        }
                        else
                        {
                                fourOrMore++;
                        }
                        kidCounter = 0;
                        check = true;
                        girl = false;
                        boy = false;
                }
                avgSum = avgSum/(twoKid+threeKid+fourOrMore);
                System.out.println("Average: "+avgSum+" children to get at least one of
each gender.");
                System.out.println("Number of families with 2 children: "+twoKid);
                System.out.println("Number of families with 3 children: "+threeKid);
                System.out.println("Number of families with 4 or more children:
"+fourOrMore);
                if(twoKid>=threeKid && twoKid>=fourOrMore)
                {
                        System.out.println("The most common number of children is 2.");
                }
                else if(threeKid>fourOrMore)
                {
                        System.out.println("The most common number of children is 3.");
                }
                else{
                        System.out.println("The most common number of children is 4.");
                }
        }
}
```

OneOfEachStat

```
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 boyGirl = generator.nextDouble();
                boolean check = true;
                boolean boy = false;
                boolean girl = false;
                int kidCounter = 0;
                //variable above from last code
                double avgSum =0;
                int twoKid=0;
                int threeKid=0;
                int fourOrMore=0;
                //int T = Integer.parseInt(args[0]);
                for(int i=0; i<T; i++)//simulates familys T times
                {
                        while(check)//family simulation
                        {
                                kidCounter++;
                                if(boyGirl <= 0.5)
                                {
                                        girl = true;
                                }
                                if(boyGirl > 0.5)
                                {
                                        boy = true;
                                }
                                if(boy && girl)//exit if you have both
                                {
                                        check = false;
```

```
                        }
                        boyGirl = generator.nextDouble();
                }
                avgSum = avgSum + kidCounter;
                if(kidCounter == 2)//check how many kids it took
                {
                        twoKid++;
                }
                else if(kidCounter == 3)
                {
                        threeKid++;
                }
                else
                {
                        fourOrMore++;
                }
                kidCounter = 0;
                check = true;
                girl = false;
                boy = false;
        }
        avgSum = avgSum/(twoKid+threeKid+fourOrMore);
        System.out.println("Average: "+avgSum+" children to get at least one of
each gender.");
        System.out.println("Number of families with 2 children: "+twoKid);
        System.out.println("Number of families with 3 children: "+threeKid);
        System.out.println("Number of families with 4 or more children:
"+fourOrMore);
        if(twoKid>=threeKid && twoKid>=fourOrMore)
        {
                System.out.println("The most common number of children is 2.");
        }
        else if(threeKid>fourOrMore)
        {
                System.out.println("The most common number of children is 3.");
        }
        else{
                System.out.println("The most common number of children is 4.");
        }
        //// 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.

        }
}
```

Perfect

```
/**
 *  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]);
		int sum=1;
		String add = " 1";
		for(int i =2; i<num; i++)//check the sum of the dividing numbers
		{
			if(num%i == 0)
			{
				sum += i;
				add = add + " + " + i;
			}
		}
		if(sum == num)
		{
			System.out.println(num + " is a perfect number since "+ num + " ="
+ add);
		}
		else
		{
			System.out.println(num+ " is not a perfect number");
		}
	}
}
```

Reverse

```
/**
 * 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];
		int first=1;
		int last = word.length()-1;//first and last will find the middle when they meet
		char middle ='.';
		int charPlace = 0;
		for(int i=word.length() - 1; i>=0; i--)//loop that prints the string backwords
		{
			System.out.print(word.charAt(i));
		}
		System.out.println();
		while(first < last)//loop to find the middle character
		{
			first++;
			last--;
			charPlace++;
		}
		middle = word.charAt(charPlace);
		System.out.println("The middle character is "+middle);
	}
}
```