```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 num = Integer.parseInt(args[0]); //Get the number from the user
        for(int i = 1; i <= num; i++){
            if (num % i ==0) { //checks if the current i is a divisor of num
                System.out.println(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 input = args[0]; //Set input to be the word the user chose
        int n = input.length();
        for(int i = n; i>0; i--){
            System.out.print(input.charAt(i-1)); //Prints the input in reversed order
        }
        System.out.println("");
        n = n/2; //Set N to be the middle value of the string.
        if(input.length() % 2 == 0){ // in case that the string is even n/2 will give us the middle char + 1. because we
count fom 0.
            n = n-1; // adjust the value of n to be in the right place.
            System.out.println("The middle character is " + input.charAt(n));
        }
        else{
            System.out.println("The middle character is " + input.charAt(n));
        }


    }
}
```

```java
/**
 *  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 current = (int)(Math.random()*10); //generates a number between 0 , 10 exclusive.
      int new_num = current;

      while(new_num >= current){ // generates new number untill the nre number is lower then the current number
         System.out.print(new_num + "");
         current = new_num;
         new_num = (int)(Math.random()*10);
      }
   }
}
```

```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 num = Integer.parseInt(args[0]);
        String perfect =  num + " is a perfect number since " + num + " = ";
        int count = 0; // count will sum up the divisiors

        for(int i = 1; i <num; i++){
            if (num % i ==0) { //checks if the current i is a divisor of num
                count += i;
                perfect += i + " + ";
            }
        }
        if(count == num){
            String new_perfect = perfect.substring(0, perfect.length()-3); //new string output without the " + " at the end.
            System.out.println(new_perfect);
        }
        else{
            System.out.println(num + " 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 size = Integer.parseInt(args[0]); //gets the size of the board from the user
        for(int i = 0; i < size; i++){
            for(int j = 0; j<size; j++){
                if (i%2 == 0) { //allign the printing with the demand in the exercise
                    System.out.print("* ");
                }
                else{
                    System.out.print(" *");
                }
            }
            System.out.println("");
        }
    }
}
```

```java
/**
 *  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) {
      boolean IsABoy = false; //  This variable will become true when a boy is born.
      boolean IsAGirl = false; //  This variable will become true when a girl is born.
      double Birth = Math.random(); //holds the results of the birth.
      int child_count = 0; //counts how many children were born.

      while (IsABoy == false || IsAGirl == false) {
         child_count +=1;
         if(Birth <= 0.5){ //the chances for a girl
            System.out.print("g ");
            IsAGirl = true;
         }
         else{
            System.out.print("b ");
            IsABoy = true;
         }
         Birth = Math.random();
      }
      System.out.println("");
      System.out.println("You made it... and you now have " + child_count + " children.");
   }
}
```

```java
/**
 *  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 Rep = Integer.parseInt(args[0]);
      int sim_count = 0; //This variable will count how many times we did the test

      boolean IsABoy = false; //This variable will become true when a boy is born.
      boolean IsAGirl = false; //This variable will become true when a girl is born.
      double Birth = Math.random(); //This variable holds the results of the birth.
      int child_count = 0; //This variable counts how many children were born inside the current simulation.
      double Allchildren = 0; //This variable counts how many children were totally born.
      double Avarage = 0.0;


      //variables that count the amount of families with 2, 3 ,4/more childrens
      int fam2_count = 0;
      int fam3_count = 0;
      int fam4_count = 0;

      while(sim_count < Rep)
      {
         IsABoy = false;
         IsAGirl = false; // restes the 3 Variabeles in the start of each simulation
         child_count = 0;
         while (IsABoy == false || IsAGirl == false) {
            child_count +=1;
            Allchildren +=1;
            if(Birth <= 0.5){ //checks if a boy or a girl was born.
               IsAGirl = true;
            }
            else{
               IsABoy = true;
```

```
            }
            Birth = Math.random();
        }
        //checks how many children were in the family and updates the counters.
        if (child_count == 2) {
            fam2_count += 1;
        }
        else if (child_count ==3) {
            fam3_count +=1;
        }
        else{
            fam4_count +=1;
        }

        sim_count +=1;
    }
    Avarage = (double)(Allchildren/Rep); //Average of how many children were born untill each family had a b oy and a girl.
    System.out.println("Avarge: " + Avarage + " children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: " + fam2_count);
    System.out.println("Number of families with 3 children: " + fam3_count);
    System.out.println("Number of families with 4 or more children: " + fam4_count);
    if(fam2_count > fam3_count && fam2_count > fam4_count) {
        System.out.println("The most common number of children is 2.");
    }
    else if (fam3_count > fam2_count && fam3_count > fam4_count) {
        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.");
    }
  }
}
```

```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 Rep = 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);

      //// 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.
      int sim_count = 0;

      boolean IsABoy = false; //This variable will become true when a boy is born.
      boolean IsAGirl = false; //This variable will become true when a girl is born.
      double Birth =  generator.nextDouble(); //This variable holds the results of the birth.
      int child_count = 0; //This variable counts how many children were born inside the current simulation.
      double Allchildren = 0; //This variable counts how many children were totally born.
      double Avarage = 0.0;

      //variables that count the amount of families with 2, 3 ,4/more childrens
      int fam2_count = 0;
```

```
        int fam3_count = 0;
        int fam4_count = 0;

        while(sim_count < Rep)
        {
            IsABoy = false;
            IsAGirl = false; // restes the 3 Variabeles in the start of each simulation
            child_count = 0;
            while (IsABoy == false || IsAGirl == false) {
                child_count +=1;
                Allchildren +=1;
                if(Birth <= 0.5){ //checks if a boy or a girl was born.
                    IsAGirl = true;
                }
                else{
                    IsABoy = true;
                }
                Birth = generator.nextDouble();
            }
            //checks how many children were in the family and updates the counters.
            if (child_count == 2) {
                fam2_count += 1;
            }
            else if (child_count ==3) {
                fam3_count +=1;
            }
            else{
                fam4_count +=1;
            }

            sim_count +=1;
        }
        Avarage = (double)(Allchildren/Rep); //Average of how many children were born untill each family had a b oy and a girl.
        System.out.println("Average: " + Avarage + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + fam2_count);
        System.out.println("Number of families with 3 children: " + fam3_count);
```

```
        System.out.println("Number of families with 4 or more children: " + fam4_count);
        if(fam2_count > fam3_count && fam2_count > fam4_count) {
            System.out.println("The most common number of children is 2.");
        }
        else if (fam3_count > fam2_count && fam3_count > fam4_count) {
            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.");
        }



    }
}
```