**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) {
    //// Put your code here
    int n = Integer.parseInt(args[0]);
    DamkaGenerater(n);
  }
  public static void DamkaGenerater(int num) {
    for (int i = 0; i < num;i++){
      String isSpace = "";
      if (i % 2 == 1) {
        isSpace = (" *");
      }
      else{
        isSpace = "* ";
      }
      for (int j = 0; j < num; j++) {
        System.out.print(isSpace);
      }
      System.out.println();
    }

  }
}
```

**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) {
    //// Put your code here
    //get command line argument int
    int divisor = Integer.parseInt(args[0]);
    getDivisors(divisor);
  }
  public static void getDivisors(int num) {
    for (int i = 1; i <= num; i++) {
      if (num % i == 0) {
        System.out.println(i);
      }
    }
  }
}
```

**InOrder:**

```
/**
*  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) {
    //// Write your code here
    int start = 0;
    int end = 10;
    inOrder(start,end);
  }
  public static void inOrder(int begRange, int endRange){
    int num = (int) (Math.random()*(endRange-begRange+1)+begRange);
    int newNum = 0;
    while (newNum <= num) {
      System.out.print(num + " ");
      newNum = num;
      num = (int) (Math.random()*(endRange-begRange+1)+begRange);
    }
  }


//  public void
}
```

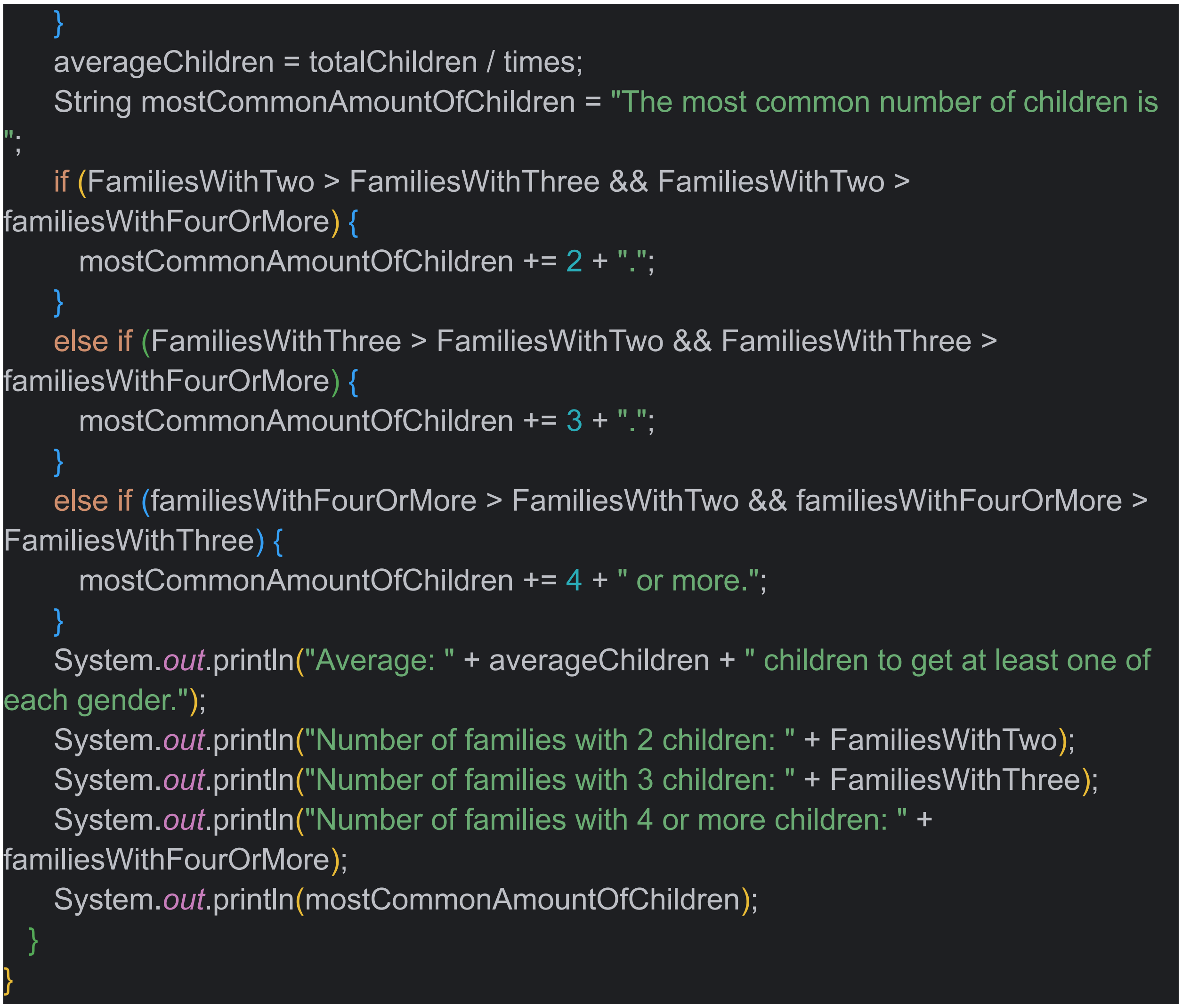
**OneOfEach:**

```
/**
*  Simulates the formation of a family in which the parents decide
*  to have children until they have at least one child of each gender.
*/
public class OneOfEach {
  public static void main (String[] args) {
    //// Put your code here
    oneOfEach();
  }
  public static void oneOfEach() {
    String children = "";
    int numOfBoys = 0;
    int numOFGirls = 0;
    while (numOfBoys == 0 || numOFGirls == 0) {
      double rnd = Math.random();
      if (rnd < 0.5) {
        children += "b";
        numOfBoys++;
      }
      else{
        children += "g";
        numOFGirls++;
      }
      children += " ";
    }
    System.out.println(children);
    System.out.println("You made it. . . and you now have " + numOFGirls+numOfBoys
+ " children.");
  }
}
```

**OneOfEachStats:**

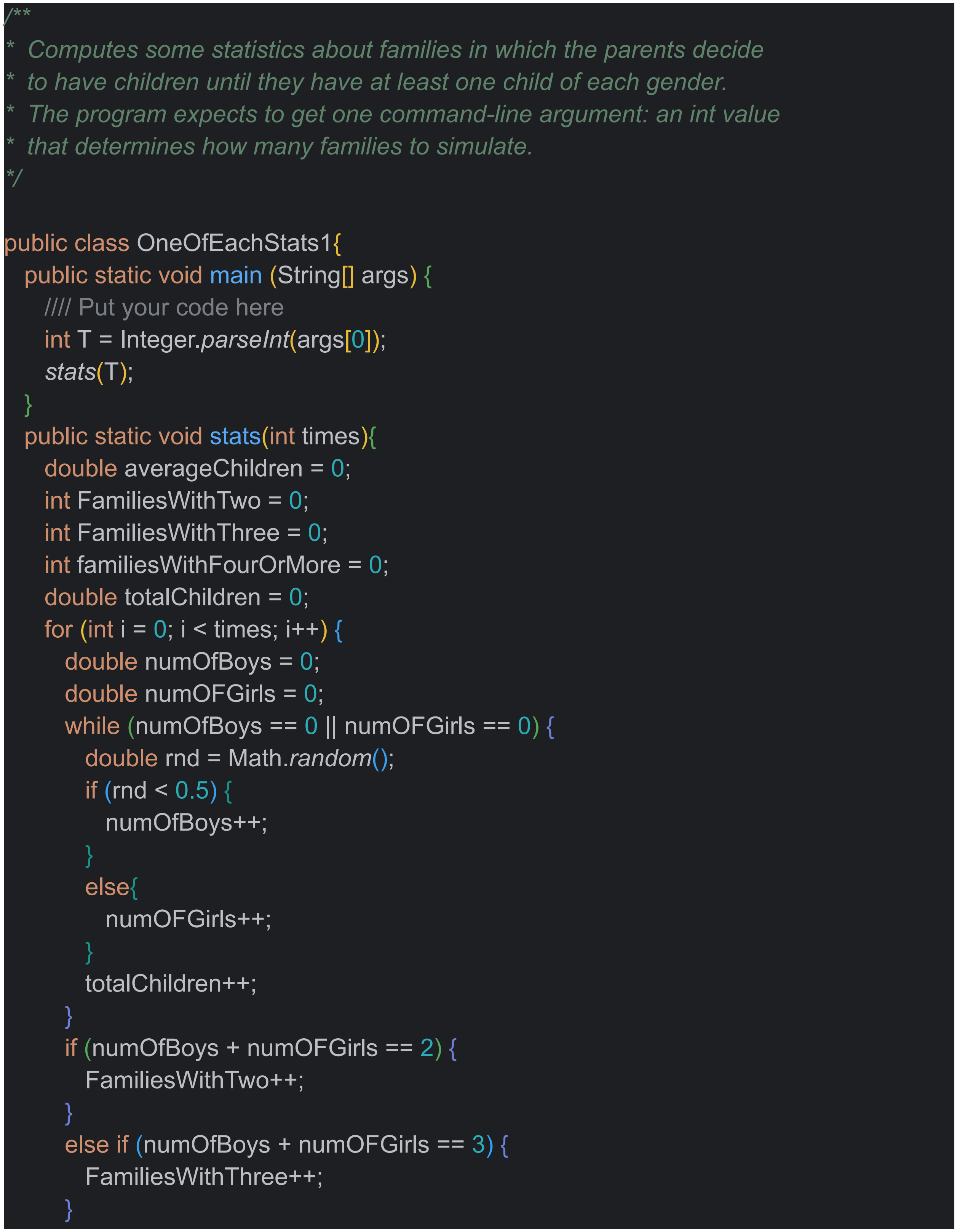
```
import java.util.Random;
public class OneOfEachStats {
  public static void main (String[] args) {
    // Gets the two command-line arguments
    int T = Integer.parseInt(args[0]);
    int seed = Integer.parseInt(args[1]);
    // Initailizes a random numbers generator with the given seed value
     Random generator = new Random(seed);
    stats(T, generator);

  }
  public static void stats(int times, Random seed){
    double averageChildren = 0;
    int FamiliesWithTwo = 0;
    int FamiliesWithThree = 0;
    int familiesWithFourOrMore = 0;
    double totalChildren = 0;
    for (int i = 0; i < times; i++) {
      double numOfBoys = 0;
      double numOFGirls = 0;
      while (numOfBoys == 0 || numOFGirls == 0) {
        double rnd = seed.nextDouble();
        if (rnd < 0.5) {
          numOfBoys++;
        }
        else{
          numOFGirls++;
        }
        totalChildren++;
      }
      if (numOfBoys + numOFGirls == 2) {
        FamiliesWithTwo++;
      }
      else if (numOfBoys + numOFGirls == 3) {
        FamiliesWithThree++;
      }
      else if (numOfBoys + numOFGirls >= 4) {
        familiesWithFourOrMore++;
      }
```



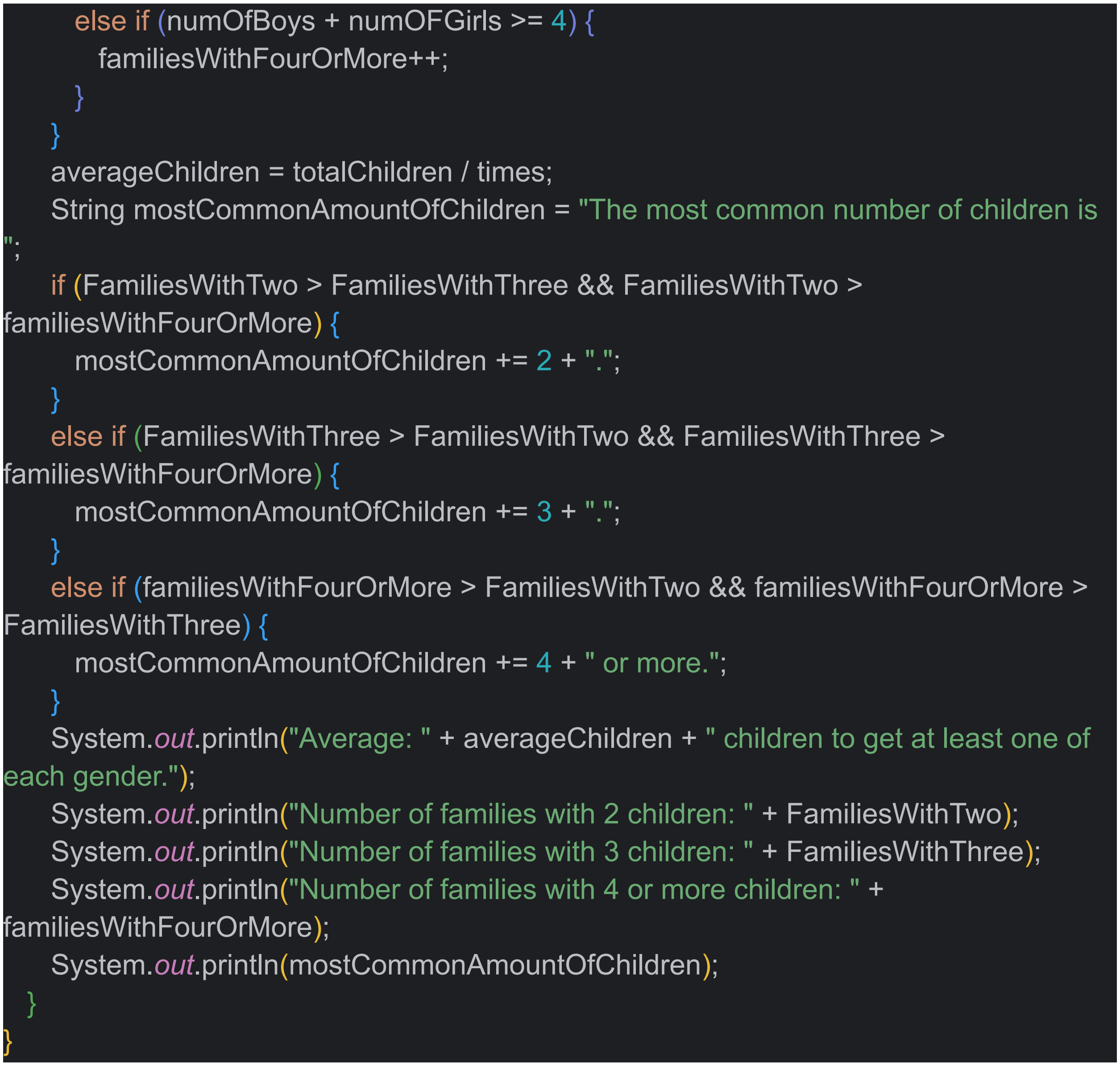
```
    }
    averageChildren = totalChildren / times;
    String mostCommonAmountOfChildren = "The most common number of children is ";
    if (FamiliesWithTwo > FamiliesWithThree && FamiliesWithTwo > familiesWithFourOrMore) {
      mostCommonAmountOfChildren += 2 + ".";
    }
    else if (FamiliesWithThree > FamiliesWithTwo && FamiliesWithThree > familiesWithFourOrMore) {
      mostCommonAmountOfChildren += 3 + ".";
    }
    else if (familiesWithFourOrMore > FamiliesWithTwo && familiesWithFourOrMore > FamiliesWithThree) {
      mostCommonAmountOfChildren += 4 + " or more.";
    }
    System.out.println("Average: " + averageChildren + " children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: " + FamiliesWithTwo);
    System.out.println("Number of families with 3 children: " + FamiliesWithThree);
    System.out.println("Number of families with 4 or more children: " + familiesWithFourOrMore);
    System.out.println(mostCommonAmountOfChildren);
  }
}
```

**OneOfEachStats1:**



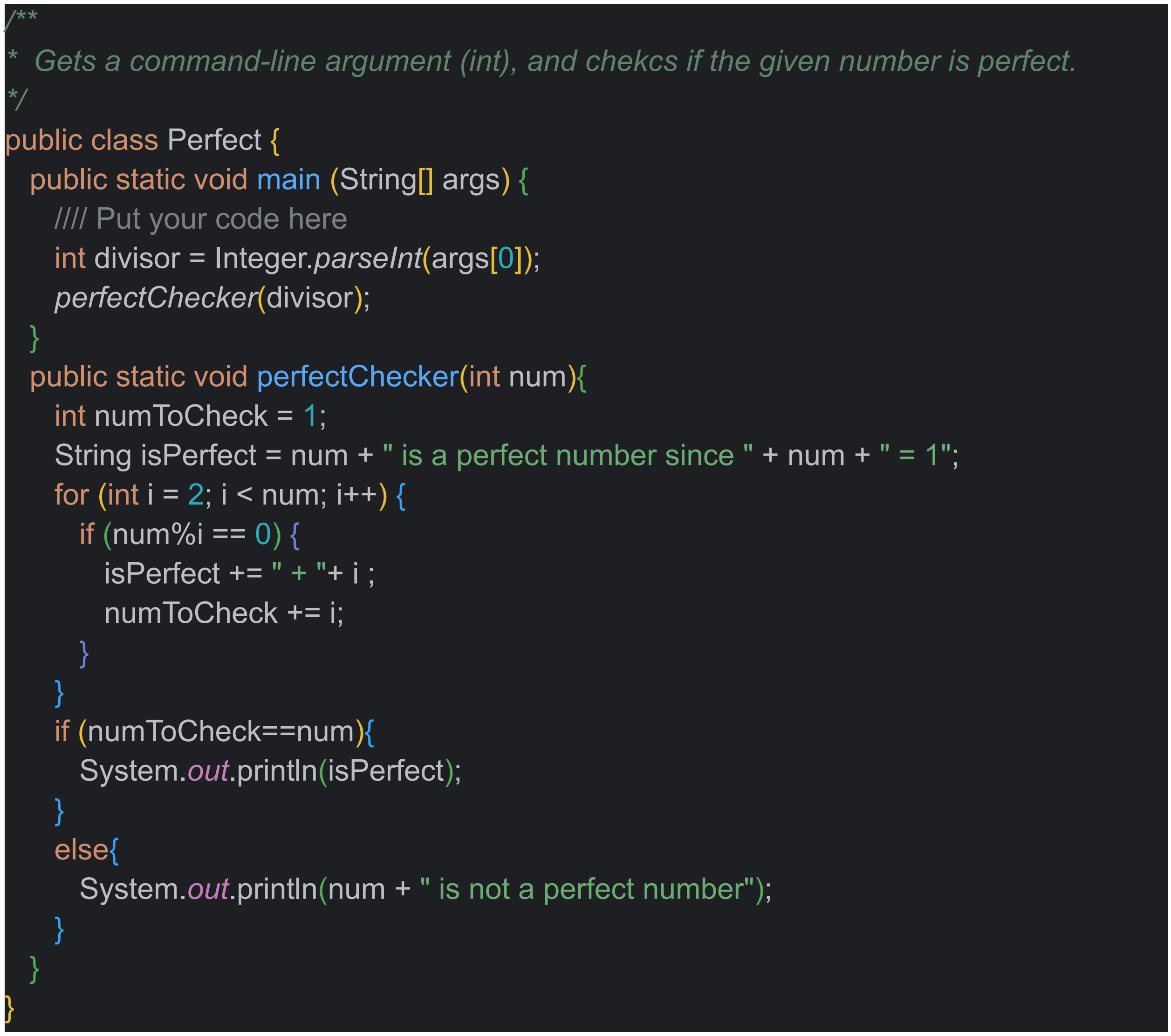
```
/**
*  Computes some statistics about families in which the parents decide
*  to have children until they have at least one child of each gender.
*  The program expects to get one command-line argument: an int value
*  that determines how many families to simulate.
*/

public class OneOfEachStats1{
  public static void main (String[] args) {
    //// Put your code here
    int T = Integer.parseInt(args[0]);
    stats(T);
  }
  public static void stats(int times){
    double averageChildren = 0;
    int FamiliesWithTwo = 0;
    int FamiliesWithThree = 0;
    int familiesWithFourOrMore = 0;
    double totalChildren = 0;
    for (int i = 0; i < times; i++) {
      double numOfBoys = 0;
      double numOFGirls = 0;
      while (numOfBoys == 0 || numOFGirls == 0) {
        double rnd = Math.random();
        if (rnd < 0.5) {
          numOfBoys++;
        }
        else{
          numOFGirls++;
        }
        totalChildren++;
      }
      if (numOfBoys + numOFGirls == 2) {
        FamiliesWithTwo++;
      }
      else if (numOfBoys + numOFGirls == 3) {
        FamiliesWithThree++;
      }
```



```
      else if (numOfBoys + numOFGirls >= 4) {
        familiesWithFourOrMore++;
      }
    }
    averageChildren = totalChildren / times;
    String mostCommonAmountOfChildren = "The most common number of children is ";
    if (FamiliesWithTwo > FamiliesWithThree && FamiliesWithTwo > familiesWithFourOrMore) {
      mostCommonAmountOfChildren += 2 + ".";
    }
    else if (FamiliesWithThree > FamiliesWithTwo && FamiliesWithThree > familiesWithFourOrMore) {
      mostCommonAmountOfChildren += 3 + ".";
    }
    else if (familiesWithFourOrMore > FamiliesWithTwo && familiesWithFourOrMore > FamiliesWithThree) {
      mostCommonAmountOfChildren += 4 + " or more.";
    }
    System.out.println("Average: " + averageChildren + " children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: " + FamiliesWithTwo);
    System.out.println("Number of families with 3 children: " + FamiliesWithThree);
    System.out.println("Number of families with 4 or more children: " + familiesWithFourOrMore);
    System.out.println(mostCommonAmountOfChildren);
  }
}
```

**Reverse:**



```
/**
*  Gets a command-line argument (int), and chekcs if the given number is perfect.
*/
public class Perfect {
  public static void main (String[] args) {
    //// Put your code here
    int divisor = Integer.parseInt(args[0]);
    perfectChecker(divisor);
  }
  public static void perfectChecker(int num){
    int numToCheck = 1;
    String isPerfect = num + " is a perfect number since " + num + " = 1";
    for (int i = 2; i < num; i++) {
      if (num%i == 0) {
        isPerfect += " + "+ i ;
        numToCheck += i;
      }
    }
    if (numToCheck==num){
      System.out.println(isPerfect);
    }
    else{
      System.out.println(num + " is not a perfect number");
    }
  }
}
```

**Reverse:**

```
import java.util.Arrays;

/**
* 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){
    //// Put your code here
    String input = args[0];
    reverse(input);
  }
  public static void reverse(String input){
    String reversed = "";
    for (int i = input.length() - 1; i >= 0; i--) {
      reversed += input.charAt(i);
    }
    System.out.println(reversed);
    System.out.println("The middle character is "+input.charAt((input.length()-1)/2));
  }

}
```