```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 n = Integer.parseInt(args[0]);
                int i = 1;
                while ( i <= n ) {
                        if ( n % i == 0)
                        {
                                System.out.println(i);
                        }
                        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 str = args[0];
      int length = str.length();
                  //String reversedStr = " ";
                  StringBuilder reversedStr = new StringBuilder();
      for (int i = length - 1; i >= 0; i--) {
         reversedStr.append(str.charAt(i));
      }
                  System.out.println(reversedStr);

                  int middleIndex = length / 2;
                  char middleChar;
                  if (length % 2 != 0) {
         middleChar = str.charAt(middleIndex);
      }
                  else {
         middleChar = str.charAt(middleIndex - 1);
      }
      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 firstNum = (int) (Math.random() * 10);
   System.out.print(firstNum);
        while (true) {
            int randomNum = (int) (Math.random() * 10);
            if (randomNum >= firstNum) {
                System.out.print(" " + randomNum);
                firstNum = randomNum;
            }
            else {
                break;
            }
   }
    }
}
```

```
/**
 *  Gets a command-line argument (int), and chekcs if the given number is perfect.
 */
public class Perfect {
    public static void main (String[] args) {
        int number = Integer.parseInt(args[0]);
   int sumOfDivisors = 1;
        String resultString = number + " is a perfect number since "+number+" = 1";


        for (int i = 2; i <= number / 2; i++) {
    if (number % i == 0) {
     sumOfDivisors += i;
     resultString= resultString+" + "+i;
    }
   }
   if (sumOfDivisors != number) {
            resultString=number + " is not a perfect number";
   }

        System.out.println(resultString);


    }
}
```

```
/**
 *  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 = 1; i <= n; i++) {
            if (i % 2 == 0) {
                System.out.println(" *".repeat(n));
            }
            else {
                System.out.println("* ".repeat(n));
            }
        }
    }
}
```

```
/**
 *  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) {
                  int childrenCount = 0;
      boolean boyBorn = false;
      boolean girlBorn = false;
                  while (!(boyBorn && girlBorn)) {
                            if (Math.random() < 0.5) {
            System.out.print("b "); // Print 'b' for boy
            boyBorn = true;
         } else {
            System.out.print("g "); // Print 'g' for girl
            girlBorn = true;
         }
                           childrenCount++;
                  }
                  System.out.println("\nYou made it... and you now have "+ childrenCount + "
children.");
         }
}
```

```
/**
 *  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 T = Integer.parseInt(args[0]);
        int totalChildren = 0;
   int twoChildrenCount = 0;
   int threeChildrenCount = 0;
   int fourOrMoreChildrenCount = 0;
        String firstMostCommon = "";
        for  (int i = 0; i < T; i++) {
            int childrenCount = 0;
            boolean boyBorn = false;
            boolean girlBorn = false;
            while (!(boyBorn && girlBorn)) {
                 if (Math.random() < 0.5) {
                    boyBorn = true;
                } else {
                    girlBorn = true;
                }
                childrenCount++;
            }
            totalChildren = totalChildren + childrenCount;
            if (childrenCount == 2) {
                twoChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("2");
            }
            if (childrenCount == 3) {
                threeChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("3");
            }
            if (childrenCount > 3) {
                fourOrMoreChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("4");
            }
        }

        System.out.println("Average: "+ (double) totalChildren/T + " children to get at
least one of each gender");
        System.out.println("Number of families with two children: " + twoChildrenCount);
        System.out.println("Number of families with three children: " +
threeChildrenCount);
        System.out.println("Number of families with four or more children: " +
fourOrMoreChildrenCount);
        if ((twoChildrenCount > threeChildrenCount && twoChildrenCount >
fourOrMoreChildrenCount) || (firstMostCommon.charAt(0) == '2') ){
            System.out.println("The most common number of children is 2.");
        }
        else if ( (threeChildrenCount > twoChildrenCount && threeChildrenCount >
fourOrMoreChildrenCount) || (firstMostCommon.charAt(0) == '3') ) {
            System.out.println("The most common number of children is 3." );
        }
        else if ((fourOrMoreChildrenCount > threeChildrenCount &&
fourOrMoreChildrenCount > twoChildrenCount) || (firstMostCommon.charAt(0) == '4' ) )
        {
            System.out.println("The most common number of children is 4 or more." );
        }
    }
}
```

```
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 totalChildren = 0;
   int twoChildrenCount = 0;
   int threeChildrenCount = 0;
   int fourOrMoreChildrenCount = 0;
        String firstMostCommon = "";
        for  (int i = 0; i < T; i++) {
            int childrenCount = 0;
            boolean boyBorn = false;
            boolean girlBorn = false;
            while (!(boyBorn && girlBorn)) {
                if (generator.nextDouble() < 0.5) {
                    boyBorn = true;
                } else {
                    girlBorn = true;
                }
                childrenCount++;
            }
            totalChildren = totalChildren + childrenCount;
            if (childrenCount == 2) {
                twoChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("2");
            }
            if (childrenCount == 3) {
                threeChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("3");
            }
            if (childrenCount > 3) {
                fourOrMoreChildrenCount ++;
                firstMostCommon = firstMostCommon.concat("4");
            }
        }

        System.out.println("Average: "+ (double) totalChildren / 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: " +
fourOrMoreChildrenCount);
        if ((twoChildrenCount > threeChildrenCount && twoChildrenCount >
fourOrMoreChildrenCount) || (firstMostCommon.charAt(0) == '2') ){
            System.out.println("The most common number of children is 2.");
        }
        else if ( (threeChildrenCount > twoChildrenCount && threeChildrenCount >
fourOrMoreChildrenCount) || (firstMostCommon.charAt(0) == '3') ) {
            System.out.println("The most common number of children is 3." );
        }
        else if ((fourOrMoreChildrenCount > threeChildrenCount &&
fourOrMoreChildrenCount > twoChildrenCount) || (firstMostCommon.charAt(0) == '4' ) )
        {
            System.out.println("The most common number of children is 4 or more." );
        }
    }
}
```