```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 x = Integer.parseInt(args [0]);

        for (int i = 1; i <= x; i++) {
            if(x % i == 0){

                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 originalString =  (args [0]);
      String reverseString = "";
      char char1;
      int position;
      int length;

      for (int i = 0; i < originalString.length(); i ++){

         char1 = originalString.charAt(i);

         reverseString = char1 + reverseString;


      }
      System.out.println(reverseString);

      if (originalString.length() % 2 == 0){
         position = originalString.length()/2-1;
         length = 1;


      }
      else {
         position = originalString.length()/2;
         length = 1;
      }
```

```
      System.out.println("The middle character is " + originalString.substring(position, position + length));


   }
}
```

```
/**
 * 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 randomNum = (int) (Math.random() * 10);
    int nextNum;
    int temp;
    System.out.println(randomNum);


      do{
        nextNum =  (int) (Math.random() * 10);
       if ( nextNum > randomNum){
         System.out.println(" " + nextNum);
         temp = randomNum;
         randomNum = nextNum;
         nextNum = temp;
       }

      } while (randomNum > nextNum);
    }
```

```
}
```

```
/**
 * 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]);
        boolean change = true;



        for (int i= 0; i< n; i++){
            String str="";
            for (int j= 0; j< n; j++){
                if(change==true){
                    str +=  "* ";
                }
                if(change==false){
                    str +=  " *";
                }
            }
            if(change==true){
```

```
            change=false;
        }else{
            change=true;
        }

        System.out.println(str);

    }

  }
}
```

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

      for (int i=2; i<=n; i++){
         if(n%i==0 && i!=n){
            sum += i;
            divisors += " + " + i;
```

```
            }
        }
        if (sum  +  1 == n) {
            System.out.println(divisors);

        }
        else{
            System.out.println(n + " 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
```

```java
 * 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 avrgChildren = 0;
      String mostCommon = "";
      int twoChild =0;
      int threeChild =0;
      int fourOrMoreChild =0;

      for (int i=0; i<T; i++){
         boolean boy= false;
         boolean girl= false;
         int counter = 0;


         while(boy == false || girl == false){
            double n = generator.nextDouble();
            if(n ==  0){
               boy = true;

         }
            else {
               girl = true;

         }
            counter++;
      }
         avrgChildren += counter;
         if (counter==2){
```

```
            twoChild++;

        }
        else if (counter==3){
            threeChild++;
        }
        else if(counter>=4){
            fourOrMoreChild++;
        }


    }
    System.out.println("Average: "+ ((double)avrgChildren) /T +" children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: "+twoChild);
    System.out.println("Number of families with 3 children: "+threeChild);
    System.out.println("Number of families with 4 or more children: "+fourOrMoreChild);
    if(twoChild> threeChild && twoChild> fourOrMoreChild){
        mostCommon += 2 +".";
    }
    else if(threeChild> twoChild && threeChild> fourOrMoreChild){
        mostCommon += 3+".";
    }
    else if(fourOrMoreChild> twoChild && fourOrMoreChild> threeChild){
        mostCommon += 4+" or more.";
    }
    System.out.println("The most common number of children is " + mostCommon);



}

    //// 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.


}
```