```
1   /**
2    *  Gets a command-line argument (int), and prints all the divisors of the given
     number.
3    */
4   public class Divisors {
5       public static void main (String[] args) {
6           int num = Integer.parseInt(args[0]);        //gets input
7
8       for (int i = 1; i<=num; i++) {              //checking if i is divisor of num
9           if (num%i==0)
10          System.out.println(i);
11          else ;
12                          }
13
14      }
15  }
16
17
18  /**
19   * Prints a given string, backward. Then prints the middle character in the string.
20   * The program expects to get one command-line argument: A string.
21   */
22  public class Reverse {
23      public static void main (String[] args){
24
25          String word = args[0];
26          int length = word.length()-1;                   //checking word length, -1
            for fit the index counter
27          int middle = (length/2);                        //pulls out the middle char
            location
28
29          char m = word.charAt(middle);                   //pulls out the middle char
30          int i = length;                                 //counter for while loop
31          String reverse="";                              //variable for reverse string
32
33              while (i>=0){                               //loop for reverse the string
34              char c = word.charAt(i);
35              reverse= reverse+c;
36              i=i-1;
37              }
38
39              System.out.println(reverse);
40              System.out.println("The middle character is "+ m);
41          }
42  }
43
44
45  /**
46   *  Generates and prints random integers in the range [0,10),
47   *  as long as they form a non-decreasing sequence.
48   */
49  public class InOrder {
50      public static void main (String[] args) {
51
52          int randomNumber1 = (int) (Math.random() * (10 ));              //generate
            random number
53          int last = randomNumber1;                                       //saves the
            last number
54
55          while (randomNumber1 >= last){                                  //print
            non-decreasing number
56          System.out.print(randomNumber1+" ");
57           last = randomNumber1;                                          // update the
             last
58          randomNumber1 = (int) (Math.random() * (10 ));                  //generate
            new random number
59          }
60      }
61  }
62
63
64  /**
65   *  Gets a command-line argument (int), and chekcs if the given number is perfect.
```

```
 */
public class Perfect {
    public static void main (String[] args) {
        int check = Integer.parseInt(args[0]);
        String str = check+ " is a perfect number since " +check+ " = 1 + ";
        //string for saveing divisors
        int counter = 1;
        for (int i = 2; i < check;i++) {
        //checking if i is divisor of num
            if(check%i == 0){
            counter = counter+i;
            //counting the divisors
            str = str+i+" + ";
            }
            else ;
        }
        if (counter == check){
        String newString = str.substring(0, str.length() - 3);
        //delete the last 3 characters from the string
        System.out.println(newString);
        }
        else{
        System.out.println(check+" is not a perfect number");
        //noting if number is not perfect
        }
        }
}

/**
 *  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 num = Integer.parseInt(args[0]);        //gets input

    for(int i = 1; i <= num;i++){                   // first loop duplicate the lines
        if ( i % 2 == 0 ){                          // if the line number is even
        than....
            for(int j = 1; j <= num;j++){
            System.out.print(" *");
                }
            }
            else
            {
            for(int m = 1; m <= num;m++){           //if its not even, print space
            before *
            System.out.print("* ");
            }
        }
        System.out.println();
        }
        }
    }


/**
 *  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 randomNumber1 = (int) (Math.random() * (10 ));      //generate random
        number
        boolean checkboy=false;                                 //0-4 for boy 5-9 for
        girl
        boolean checkgirl=false;
        if(randomNumber1 > 4)
        checkgirl= true;
        else
            checkboy=true;
        if (randomNumber1 <= 4)
        System.out.print("b ");

```

```
130          else
131          System.out.print("g ");
132
133          int i = 1;                                                  //number of children
134              while (checkboy == false || checkgirl == false){
135              randomNumber1 = (int) (Math.random() * (10 ));
136              if (randomNumber1<=4){
137              System.out.print("b ");
138              checkboy= true;
139              i++;
140              }
141                  else;
142                  if (randomNumber1 > 4) {
143                  System.out.print("g ");
144                  checkgirl=true;
145                  i++;
146                      }
147                      else;
148                  }
149                  System.out.println();
150                  System.out.print("You made it... and you now have "+ i +" children.");
151      }
152  }
153  /**
154   *  Computes some statistics about families in which the parents decide
155   *  to have children until they have at least one child of each gender.
156   *  The program expects to get one command-line argument: an int value
157   *  that determines how many families to simulate.
158   */
159  public class OneOfEachStats1 {
160      public static void main (String[] args) {
161      int T = Integer.parseInt(args[0]);                      //number of trails
162      int Two = 0;                                                //families with 2
         children
163      int Three = 0;                                          //families with 3 children
164      int Four = 0;                                               //families with 4
         children or more
165      double sum = 0;                                             //for caculating the
         average
166
167                                                              //The paste part from
                                                                 OneOfEachStats
168              for (int j = 0; j<T; j++){                          //loop for T times
169          int randomNumber1 = (int) (Math.random() * (10 ));      //generate random
             number
170          boolean checkboy = false;                                   //0-4 for boy 5-9
             for girl
171          boolean checkgirl = false;
172          if(randomNumber1 > 4)
173          checkgirl= true;
174          else
175              checkboy=true;
176          int i = 1;
177              while (checkboy == false || checkgirl == false){
178              randomNumber1 = (int) (Math.random() * (10 ));
179              if (randomNumber1 <= 4){
180              checkboy = true;
181              i++;
182              }
183                  else;
184                  if (randomNumber1 > 4) {
185                  checkgirl=true;
186                  i++;
187                      }
188                      else;                                                   //end
                         of paste part from OneOfEachStats
189                  }
190                  if(i==2)
191                      Two++;
192                  if(i==3)
193                      Three++;
194                  if(i>=4)
195                      Four++;
```

```
196                 sum=sum+i;
197             }
198 double Average= sum/T;
    //loop ends
199     System.out.println("Average: "+Average+" children to get at least one of each
        gender.");
200     System.out.println("Number of families with 2 children: "+Two);
201     System.out.println("Number of families with 3 children: "+Three);
202     System.out.println("Number of families with 4 or more children: "+Four);
203     if(Two>=Three&&Two>=Four)
204     System.out.println("The most common number of children is
        2.");                               //the IF's checking which number-
205
        if(Three>Two&&Three>=Four)
                    //- of children is the most common
206     System.out.println("The most common number of children is 3.");
207     if(Four>Two&&Four>Three)
208     System.out.println("The most common number of children is 4 or more.");
209
210     }
211 }
212 import java.util.Random;
213 /**
214  *  Computes some statistics about families in which the parents decide
215  *  to have children until they have at least one child of each gender.
216  *  The program expects to get two command-line arguments: an int value
217  *  that determines how many families to simulate, and an int value
218  *  that serves as the seed of the random numbers generated by the program.
219  *  Example usage: % java OneOfEachStats 1000 1
220  */
221 public class OneOfEachStats {
222     public static void main (String[] args) {
223         // Gets the two command-line arguments
224         int T = Integer.parseInt(args[0]);                                  //number
            of trails
225         int seed = Integer.parseInt(args[1]);
226         // Initailizes a random numbers generator with the given seed value
227         Random generator = new Random(seed);
228
229         //// In the previous version of this program, you used a statement like:
230         //// double rnd = Math.random();
231         //// Where "rnd" is the variable that stores the generated random value.
232         //// In this version of the program, replace this statement with:
233         //// double rnd = generator.nextDouble();
234         //// This statement will generate a random value in the range [0,1),
235         //// just like you had in the previous version, except that the
236         //// randomization will be based on the given seed.
237         //// This is the only change that you have to do in the program.
238
239     int Two = 0;                                                    //families with 2
        children
240     int Three = 0;                                                //families with 3 children
241     int Four = 0;                                                   //families with 4
        children or more
242     double sum = 0;                                                 //for caculating the
        average
243
244                                                                   //The paste part from
                                                                      OneOfEachStats
245             for (int j = 0; j<T; j++){                              //loop for T times
246         double randomNumber1 = generator.nextDouble();            //generate random number
247         boolean checkboy = false;                                       //0-4 for boy 5-9
            for girl
248         boolean checkgirl = false;
249         if(randomNumber1 >= 0.5)
250         checkgirl= true;
251         else
252             checkboy=true;
253         int i = 1;
254             while (checkboy == false || checkgirl == false){
255             randomNumber1 = generator.nextDouble();;
256             if (randomNumber1 < 0.5){
257             checkboy = true;
```

```
258                 i++;
259                 }
260                     else;
261                     if (randomNumber1 >= 0.5) {
262                     checkgirl=true;
263                     i++;
264                         }
265                         else;                                                    //end
                            of paste part from OneOfEachStats
266                     }
267                     if(i==2)
268                         Two++;
269                     if(i==3)
270                         Three++;
271                     if(i>=4)
272                         Four++;
273                     sum=sum+i;
274                 }
275     double Average= sum/T;
        //loop ends
276         System.out.println("Average: "+Average+" children to get at least one of each
            gender.");
277         System.out.println("Number of families with 2 children: "+Two);
278         System.out.println("Number of families with 3 children: "+Three);
279         System.out.println("Number of families with 4 or more children: "+Four);
280         if(Two>=Three&&Two>=Four)
281         System.out.println("The most common number of children is
            2.");                               //the IF's checking which number-
282
            if(Three>Two&&Three>=Four)
                        //- of children is the most common
283         System.out.println("The most common number of children is 3.");
284         if(Four>Two&&Four>Three)
285         System.out.println("The most common number of children is 4 or more.");
286         }
287     }
288
```