1. **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) {
      int num = Integer.parseInt(args[0]);
      int index = 1;
      while (index <= num) {
         if (num % index == 0) {
            System.out.println(index);
         }
         index = index + 1;
      }

      //// Put your code here
   }
}
```

2. **Reversing a string**

```
/**
 * 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 original = args[0];
      int index = original.length() - 1;
      String reversed = "";
      while (index >= 0) {
         reversed += original.charAt(index);
         index -= 1;
      }
      //// Put your code here
      System.out.println(reversed);
      if (original.length() % 2 == 0) {
         System.out.println("The middle character is " + reversed.charAt((original.length() /
2)));
      } else {
         System.out.println("The middle character is " + reversed.charAt(((original.length() - 1)
/ 2)));
      }
   }
}
```

3. **Lucky Streak**

```
/**
 * 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 rand_num = (int) (Math.random() * 10); // going from 0-9
      int minimum = rand_num; // initialising my minimum as the first random
      do {
         System.out.print(rand_num + " "); // print out first random
         rand_num = (int) (Math.random() * 10); // generate a new random
         if (rand_num > minimum) { // if the random number is less than the max, make it the
new max
            // i.e. if my first random is 9, and my new is 5, make 5 the new max.
            minimum = rand_num;
         }
      } while (rand_num >= minimum); // loop while my random number is less than my max
num. i.e. if I get 4 and max
                              // is 5, end loop.

   }
}
```

4. Perfect Numbers

```
/**
 * Gets a command-line argument (int), and chekcs if the given number is
 * perfect.
 * we say a number is perfect if it equals the sum of all its divisors (6 = 3 +
 * 2 + 1)
 */
public class Perfect {
   public static void main(String[] args) {
      int num = Integer.parseInt(args[0]);
      int index = 1;
      int test = 0;
      String s = "";
      while (index < num) {
         if (num % index == 0) {
            test += index;
         }
         index += 1;
      }
      if (test == num) {
         System.out.print(num + " is a perfect number since " + num + " = ");
         int second_index = 1;
         while (second_index < num) {
            if (num % second_index == 0) {
               s += second_index + " + ";
            }
            second_index += 1;
         }
         s = s.substring(0, s.length() - 2);
         System.out.print(s);
      } else {
         System.out.println(num + " is not a perfect number");
      }
      //// Put your code here
   }
}
```

5. **Damka**

```
/**
 * 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]);
      int row_index = 1;
      String row = "";
      do {
         // even rows
         if (row_index % 2 == 0) {
            System.out.print(" ");
            for (int i = 0; i < n; i++) {
               row += "* ";
            }
            row = row.substring(0, n * 2 - 1);
            // odd rows
         } else {
            for (int i = 0; i < n; i++) {
               row += "* ";
            }
         }
         System.out.print(row + "\n");
         row = "";
         row_index += 1;
      } while (row_index <= n); //// Put your code here
      System.out.print("\n");
   }
}
```

6. One of Each

```
/**
 * 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 both = false;
      boolean boy = false;
      boolean girl = false;
      int num_kids = 0;
      while (!both) {
         double rand = (Math.random());
         if (rand > 0.5) {
            System.out.print("b ");
            boy = true;
         } else {
            System.out.print("g ");
            girl = true;
         }
         num_kids += 1;
         both = girl && boy;
      }
      System.out.print("\n");
      System.out.println("You made it... and you now have " + num_kids + " children");
      //// Put your code here
   }
}
```

7. One of each Stats

```
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);

      //// Put your code here
      int num_trials = Integer.parseInt(args[0]);
      // I want my program to run the previous code each time, and then store the
      // number of children in a total
      // and store the number of children in a batch either 2,3, or 4+
      int two_children = 0;
      int three_children = 0;
      int four_children = 0;
      int trial_index = 0;
      int sum_children = 0;
      while (trial_index < num_trials) {
         boolean both = false;
         boolean boy = false;
         boolean girl = false;
         int num_kids = 0;
         while (!both) {
            double rand = (generator.nextDouble());
            if (rand > 0.5) {
               boy = true;
            } else {
               girl = true;
            }
            num_kids += 1;
            both = girl && boy;
         }
         switch (num_kids) {
            case 2:
```

```
            two_children += 1;
            break;
          case 3:
            three_children += 1;
            break;
          default:
            four_children += 1;
            break;
        }
        sum_children += num_kids;
        trial_index += 1;
      }
      double average_kids = (double) sum_children / num_trials;
      System.out.println("Average: " + average_kids + " children to get at least one of each
gender.");
      System.out.println("Number of families with 2 children: " + two_children);
      System.out.println("Number of families with 3 children: " + three_children);
      System.out.println("Number of families with 4 or more children: " + four_children);
      int largest_group = Math.max(Math.max(two_children, three_children),
            four_children);
      if (largest_group == two_children) {
        System.out.println("The most common number of children is 2.");
      } else if (largest_group == three_children) {
        System.out.println("The most common number of children is 3.");
      } else {
        System.out.println("The most common number of children is 4.");
      }

      //// 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.

    }
}
```

8. One of each Stats

```
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);

      //// Put your code here
      int num_trials = Integer.parseInt(args[0]);
      // I want my program to run the previous code each time, and then store the
      // number of children in a total
      // and store the number of children in a batch either 2,3, or 4+
      int two_children = 0;
      int three_children = 0;
      int four_children = 0;
      int trial_index = 0;
      int sum_children = 0;
      while (trial_index < num_trials) {
         boolean both = false;
         boolean boy = false;
         boolean girl = false;
         int num_kids = 0;
         while (!both) {
            double rand = (generator.nextDouble());
            if (rand > 0.5) {
               boy = true;
            } else {
               girl = true;
            }
            num_kids += 1;
            both = girl && boy;
         }
         switch (num_kids) {
            case 2:
```

```
            two_children += 1;
            break;
          case 3:
            three_children += 1;
            break;
          default:
            four_children += 1;
            break;
        }
        sum_children += num_kids;
        trial_index += 1;
      }
      double average_kids = (double) sum_children / num_trials;
      System.out.println("Average: " + average_kids + " children to get at least one of each
gender.");
      System.out.println("Number of families with 2 children: " + two_children);
      System.out.println("Number of families with 3 children: " + three_children);
      System.out.println("Number of families with 4 or more children: " + four_children);
      int largest_group = Math.max(Math.max(two_children, three_children),
            four_children);
      if (largest_group == two_children) {
        System.out.println("The most common number of children is 2.");
      } else if (largest_group == three_children) {
        System.out.println("The most common number of children is 3.");
      } else {
        System.out.println("The most common number of children is 4.");
      }

      //// 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.

    }
}
```