| **Exercise 4.28** Write out the header of a for-each loop to process an **ArrayList<Track>** called **tracks**. Don’t worry about the loop’s body. |  |
| --- | --- |
| **Exercise 4.29** Suppose we express the first version of the key search in pseudo-code as follows:**boolean missing = true;****while(missing) {** **if(***the keys are in the next place***) {** **missing = false;** **}****}**Try to express the second version by completing the following outline:**boolean found = false;****while(...) {** **if(***the keys are in the next place***) {** **...** **}****}** |  |
| **Exercise 4.30** Write a while loop (for example, in a method called **multiplesOfFive**) that prints out all multiples of 5 between 10 and 95. |  |
| **Exercise 4.31** Write a while loop to sum the values 1 to 10 and print the sum once the loop has finished. |  |
| **Exercise 4.32** Write a method called **sum** with a while loop that adds up all numbers between two numbers **a** and **b**. The values for **a** and **b** can be passed to the **sum** method as parameters. |  |
| **Exercise 4.33** *Challenge task* Write a method **isPrime(int n)** that returns *true* if the parameter **n** is a prime number, and *false* if it is not. To implement the method, you can write a while loop that divides **n** by all numbers between **2** and **(n–1)** and tests whether the division yields a whole number. You can write this test by using the modulo operator (**%**) to check whether the integer division leaves a remainder of 0 (see the discussion of the modulo operator in Section 3.8.3). |  |
| **Exercise 4.34** In the **findFirst** method, the loop’s condition repeatedly asks the **files** collection how many files it is storing. Does the value returned by **size** vary from one check to the next? If you think the answer is no, then rewrite the method so that the number of files is determined only once and stored in a local variable prior to execution of the loop. Then use the local variable in the loop’s condition rather than the call to **size**. Check that this version gives the same results. If you have problems completing this exercise, try using the debugger to see where things are going wrong. |  |