### ARRAY PROGRAMMING EXERCISES

1. Write a Java application that reads a positive integer \( n \) and uses an array to generate, store and print the first \( n \) numbers of the *Fibonacci sequence*. The first number of the Fibonacci sequence is 0, the second number is 1 and each succeeding number is the sum of the previous two. For example, the first 12 Fibonacci numbers are:

\[
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
\]

If you give your array the component data type of `int` then you’ll be able to compute up to 47 Fibonacci numbers; the 48th number exceeds 32 bits.

2. Modify your solution to the previous problem by giving your array the component data type of `long`. How many Fibonacci numbers can it compute before the number exceeds 64 bits?

3. Modify your solution to the previous problem so that your program quits with an appropriate error message if the user’s input will result in the calculation of a Fibonacci number that exceeds 64 bits.

4. Look up the specification of the API class `java.math.BigInteger`, a class that allows you to create and manipulate indefinitely large integer values. Solve Exercise 1, giving your array the component data type of `BigInteger`.