The Writing R Extensions manual (http://cran.r-project.org/doc/manuals/R-exts.pdf) contains more information on the `.C` and `.Call` external interfaces. The Rcpp package offers a higher degree of flexibility for interfacing C++ code and enables using numeric C++ libraries such as GSL, Eigen, and Armadillo.

7.9 Exercises

1. Type the R code in this chapter into an R session and observe the results.

2. Implement a function that computes the log of the factorial value of an integer using a for loop. Note that implementing it using $\log(A)+\log(B)+\cdots$ avoids overflow while implementing it as $\log(A \cdot B \cdot \cdots)$ creates an overflow early on.

3. Implement a function that computes the log of the factorial value of an integer using recursion.

4. Using your two implementations of log-factorial in (2) and (3) above, compute the sum of the log-factorials of the integers $1, 2, \ldots, N$ for various N values.

5. Compare the execution times of your two implementations for (4) with an implementation based on the official R function `lfactorial(n)`. You may use the function `system.time()` to measure execution time. What are the growth rates of the three implementations as N increases? Use the command `options(expressions=500000)` to increase the number of nested recursions allowed. Compare the timing of the recursion implementation as much as possible, and continue beyond that for the other two implementations.