**Exercise: Structured programming (and some plotting) in R**

BIOS13, Lund University Pedro Rosero

1. Write a mymax function that takes two input values and returns the largest of the two.
2. Write a function that calculates the weighted sum of an input vector. The second argument to the function should be the weights, one per element of the first vector.  
   Example: If the input is v = c(3, 5, -1) and w = c(1, 2, 1), then the output should be calculated as 1\*3 + 2\*5 + 1\*(-1) = 12

*Hint*: a for loop will work for sure. Or is there a simpler way of doing this perhaps?

1. Leonardo of Pisa (about 1170 - 1250 AD) is often regarded as Europe’s first mathematician after the dark ages. Under the pseudonym Fibonacci he wrote *Liber Abaci* (Book of Calculation), where he strongly advocated the use of the Hindu-Arabic number system, which we use today. He also presented what is possibly the first population dynamic model, the *Fibonacci sequence*, which is an infinite sequence of numbers. The first two numbers are both 1. All the following numbers are calculated as the sum of the two previous. Thus the sequence starts: 1 1 2 3 5 8 11 ...

Fibonacci thought of it as a model of rabbit population growth...  
Note: This sequence was known in India before, but Fibonacci was first (I think) to give it a biological interpretation. Now:  
Write a function which takes an input variable n, calculates the n first Fibonacci numbers, and plots them. (no return values)

*Hint:* A for loop here that depends on positions i-1 and i-2 may be useful for this.

*Extra*: Extend the function above to also plot C^i/sqrt(5), where C is the Golden Ratio = (1+sqrt(5))/2 and *i* is the index in the sequence (the x-coordinate). This function is very close to the Fibonacci sequence at higher values of i.

1. Write a plotmax **function** that
   1. takes an input parameter n
   2. generates a random vector of length n (runif(n))
   3. plots the vector as a blue line
   4. highlights local maxima with red crosses
   5. **returns** two vectors in a list. The first vector contains the indices (positions) and the second vector contains the values of the local maxima.

***Tips***: Solve the points a-e one by one. Make sure your solution works before you move on to the next. Divide and conquer!  
In e), you can create an empty vector using c() the empty vector can contain either the index of the local maximum or the value of the local maximum, I’d suggest you create one vector for each:  
v <- c()

That vector can then grow using the c() command repeatedly:

v <- c( v, newvalue)

1. Write a function is\_palindrome() that tests whether a string (input parameter) is a palindrome (the same thing backwards) and returns TRUE or FALSE.  
   > is\_palindrome("anna")

[1] TRUE

> is\_palindrome("jack")

[1] FALSE

*Hint:* you’re comparing first with last letter, then second with second-to-last, then third to third-to-last, etc.

1. Write a function that merges two sorted vectors (input variables) into a new sorted vector (return value). Example: [1,4,6],[2,3,5] → [1,2,3,4,5,6].  
   *Tips! Solve the simplest of cases first, such as the input of two numbers instead of vectors: 4, 5* → *[4,5]. Next, solve the case with a fixed length of the vectors (length 2 or 3, for example). Finally, allow any length of the two input vectors (they do not have to be the same length). Make sure you have a working solution before you move on to the next step.*
2. Figure out how to create subplots, i.e. how to plot several plots in the same window (see the lecture notes).
3. Play with other plot commands! Try polygon, image, contour, persp, or try out the ggplot2 commands!