Assignments to Lecture 7 – Algorithms and Functions

Danny Arends

More practicing of the basics

The first 3 assignments here are taken from projecteuler.net which is a great resource to help you learn programming by giving 'easy' assignments to solve. For more practice you can also do these assignments and have them checked online (just create an account at project euler)

1) If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

2) Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed one million, find the sum of the even-valued terms.

3) A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99.

Find the largest palindrome made from the product of two 3-digit numbers.

Exercises belonging to the lecture

4) Write a recursive function to count towards 0.
e.g. when starting from 100 it counts to 0, when it starts at -100 it counts up to 0

5) Write your own version of Lapply, make sure to check if the input is a list and throw an error (use: the stop function) if it is not a list.

The function signature should look like

```
mylapply <- function(x, FUN, …){

}
```

6) Make a function which as an input only takes dot dot dot, this function when called should print out all the parameters passed to it.

Additional

7) Write a recursive function that calculates the greatest common divisor (gcd) of two integers a, b
Hint: Both a and b need to be non zero, the greatest common divisor is the largest positive integer that divides the numbers without a remainder. For example, the GCD of **8** and **12** is 4. Calculating it recursively is much easier then other ways, this algorithm is called Euclid's algorithm. You can just google/wikipedia the algorithm, and then implement it into R.

8) A hard question, which you can spend a long time on
Write a recursive function to draw a tree, the function signature I used

```
drawTree <- function(x1, y1, angle, depth, maxdepth = 8, nbranch = 2){
  # x1 and y1 are the starting points
  # Calculate new x and y, by using sin and cosine
  # Draw a line from x1, y1 to the new x and y
  # For each of the nbranches argument call the drawTree function again
}

plot(x = c(-10, 10), y = c(0,50), t ='n')      # Setup a plot window
drawTree(0, 0, 10, 6)                          # Add lines for the tree
```

Hint: You will need to use sin and cos, to transform from degree to radians use 1 degree = (pi / 180) radians