# Carbon Cycle — Computer Lab

First steps with R — March 21, 2012 (Abgabe: soon)

Get to know some useful programming tools in R. R is an open source programming language featuring a wealth of functions. For a documentation of functions and language syntax, check `http://de.wikibooks.org/wiki/GNU_R` or type `help(` *function*`)` in your R console. The present problem set is supposed to introduce you to some basic functions that you will use for the subsequent problem sets.

1. Use a 'for' loop to compute the sum of all natural numbers from 1 to 100. Print the result to the screen. (Remember that young Gauss calculated this by head in 3rd grade! How did he do that?)

2. Add up all numbers between 1 and 100 that are at the same time a multiple of 3 and a multiple of 7. Print the result to the screen in the form of:
`Problem 2.:  The answer is (`*result*`)`

3. Define a vector $v_i$ of length 100. $v_i = 6$, for $i = 1 : 25$; $v_i = -20$, for $i = 66 : 100$. Linearly interpolate to remaining elements that are not defined. Plot the values of $v_i$.

4. Read the famous monthly Mauna Loa $CO_2$ record from file `co2_monthly_maunaloa.txt`. Plot the data with x-values from the column `decyr` and y-values from the column `co2_avg`. Plot it as a line graph.

5. Define a function to compute a 12-month running mean of the monthly $CO_2$ time series. Arguments to your function may be a vector of the original (monthly) data and a parameter defining the number of elements over which the mean is to be taken (in this case 12). Plot the running mean overlaid onto the monthly data. Use a different color for the running mean graph.

For the present problems you may use the following functions, operators and syntax structures. For further documentation, check online. This list is by no means complete but should serve as a starting point from where you will discover many more functions by searching the web yourself.

- Functions

  > `print()`

  > `paste()`

  > `seq()`

  - `plot()`
  - `length()`
  - `read.table()`
  - `head()`
  - `array()`
  - `c()`

- Operators
  - `<-` (allocates value to variable)
  - `%%` modulo (returns the remainder of division)
  - `==` logical EQUAL
  - `&&` logical AND

- Syntax structures (examples)
  - **for**
    ```
    for (i in 1:100) {
    sum <- sum + i
    }
    ```
  - **if**
    ```
    if (condition1 && condition2 ) {
    print('The condition is evaluated to be TRUE')
    }else{
    print('The condition is evaluated to be FALSE')
    }
    ```
  - **while**
    ```
    i <- 0
    while (i < 50) {
    i <- i + 1
    print(i)
    }
    ```
  - **function**
    ```
    myexponential <- function(argument){
    result <- exp(argument)
    return(result)
    }
    ```