Data-intensive Programming

Weekly exercise #1

Task #1: Familiarise yourself with the Scala worksheet

Start Scala Ide by issuing Windows key and starting typing "scala". Create a new Scala project "DIP" (File->New->Scala Project). Create a new Scala Worksheet: DIP->New->"Scala Worksheet". You can name it "ex1", for instance. Scala code can be written in the worksheet. The code is evaluated, when the sheet is saved.

Write a simple function *sum*, that takes two integer parameters and returns their sum. Test the sum function:

```
val s = sum(20, 21)
```

Task #2: Recursive function

Write a function to compute *square root* (*sqrt(x)*) with the following algorithm:

- start with initial estimate (let's pick y = 1)
- repeatedly improve the estimate by taking mean of y and x/y

For example sqrt(2) will be computed as follows:

Round #	Estimation	Quotient	Mean
1	1	2 / 1 = 2	1.5
2	1.5	2 / 1.5 = 1.333	1.4167
3	1.4167	2 / 1.4617 = 1.4118	1.4142
…	…	…	…

Test you function with
sqrt(2)
sqrt(1e-6)
sqrt(1e60)

Task #3: Pascal's triangle

In mathematics, Pascal's triangle (https://en.wikipedia.org/wiki/Pascal%27s_triangle) is a triangular array of the binomial coefficients:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
   ...
```

Write a recursive function 'pascal' for calculating the value for certain column and row pair. The top-most 1 is at row 0 and column 0.

def pascal(column: Int, row: Int): Int = ???

Task #4: Parsing

Write a recursive function for testing that parenthesis are balanced in a string (i.e. there are as many '(' and ')' characters in a string).

def balance(chars: List[Char]): Boolean = ???

Task #5: Higher order functions 1

Assume you have an integer array:

```
val a = Array(1,2,3,4,5)
```

Use functions map and reduceLeft to compute the sum of squares of the values in the array: $\sum a_i^2$

Task #6: Higher order functions 2

Explain the following code snippet. You can try the snippet piece by piece in the worksheet or search help from Scaladoc (http://www.scala-lang.org/api/2.11.8/).

```
"sheena is a punk rocker she is a punk punk".split(" ").map(s => (s, 1)).
        groupBy(p => p._1).mapValues(v => v.length)
```

what about

```
  "sheena is a punk rocker she is a punk punk".split(" ").map((_, 1)).
        groupBy(_._1).mapValues(v => v.map(_._2).reduce(_+_))
```