Exercises for Christmas 2016

1) Define a parser comment :: Parser () for ordinary Haskell comments that begin with -- and extend to the end of the current line, which is represented by the control character '\n'.

2) Define a suitable tipe Expr for arithmetic expressions and modify the parser for espressions to have expr:: Parser Expr. Hint: Expr should represent a derivation tree of the input expression.

3) Consider expressions build up from natural numbers using a subtraction that is assumed to associate to the left.

a. Translate this description directly into a grammar.

b. implement this grammar as a parser expr:: Parser Int.

c. What is the problem with this parser?

d. Show how it can be fixed. Hint: rewrite the parser using the repetition primitive many and the library function foldl.

4) Define an expression fibs :: [Integer] that generates the infinite sequence of Fibonacci numbers:

0, 1, 1, 2, 3, 5, 8, 13,...

using the following simple procedure:

-the first two numbers are 0 and 1;

-the next number is the sum of the previous two;

-return to second step.

Hint: it may be useful to use the functions zip and tail.