FP Lab 2

1. What are the types of the following

   (a) values:

   ```
   ['a','b','c']
   ('a','b','c')
   [(False,'0'),(True,'1')]
   ([False,True],['0','1'])
   [tail,init,reverse]
   ```

   (b) functions:

   ```
   second xs     = head (tail xs)
   swap (x,y)    = (y,x)
   pair x y      = (x,y)
   double x      = x*2
   palindrome xs = reverse xs == xs
   twice f x     = f (f x)
   ```

   (c) Use GHCi to check your answers to the above two questions.

2. Is `[[False,True],['0','1']]` well-typed? Why?

3. Give definitions of the following functions:

   (a) `doubleAll :: [Integer] -> [Integer]` which doubles all the elements of a list of integers.

   (b) `isEven :: Integer -> Bool` that tests whether a number is even. (Hint: you may either use the function `mod` or mutual recursion).

   (c) `halve :: [a] -> ([a], [a])` that splits an even-length list into two halves and, in the case that the length of the input list is odd, returns the value `([],[])`. For example,

   ```
   > halve [1,2,3,4]
   ([1,2],[3,4])
   > halve [1]
   ([],[])
   ```

(Hint: you may use library functions in the definition.)

4. Consider a function `safetail` that behaves in the same way as `tail`, except that `safetail` maps the empty list to the empty list, whereas `tail` gives an error in this case. Define `safetail` using:

   (a) a conditional expression;

   (b) guarded equations;

   (c) pattern matching.

   (Hint: the library function `null :: [a] -> Bool` can be used to test if a list is empty.)