Types and Polymorphism

Instructions:

Solutions of the exercises are to be delivered before Thursday, the 22th of March at 10:15AM.
Solutions should be placed in a separate folder with the name "**Assignment04**".
Please submit answers to all the exercises in **one** text file.

Exercise 1 (3 points)

Infer types of the functions `factors`, `isPerfect` and `insert` and say whether they are monomorphic or polymorphic functions. Justify your answer.

- ```
  mod ::  Int -> Int -> Int
  factors n = [x | x <- [1..n-1], mod n x == 0 ]
  isPerfect n = sum (factors n) == n
  ```
- ```
  insert _ n [] = [n]
  insert 0 n l = n:l
  insert i n (x:xs) = x :  insert (i-1) n xs
  ```

Exercise 2 (3 points)

Infer the type of the following function and explain each of the steps.

```
f1 f x
    | f x < 0 = []
    | otherwise = x :  (f1 f (f x))
```

Optional Haskell exercise (2 points)

Write a function `deleteRepetitions l` which deletes all consecutive repetitions of elements in the list `l`. For example, `deleteRepetitions [4, 5, 5, 2, 11, 11, 11, 2, 2]` would return as the result `[4, 5, 2, 11, 2]`. **No built-in function for working with lists may be used. Only pattern matching is allowed.**