Serie 4 - Types in Haskell

NOTE Please provide all the answers in one file!

Exercise 1

Infer the types of the following functions and say whether they are monomorphic or polymorphic functions. Justify your answer.

a. `xor a b = a && not b || b && not a`

b. `containsChar c s = elem (toLower c) s`

c. `third (a, b, c) = c`

d.
```
length [] = 0
length (x:xs) = 1 + length xs
```

Exercise 2

Some functions can be used with different argument types, but rarely with any type. Consider the following function types, and identify which type is invalid for each function. Justify your answer.

```
a)  square :: Int -> Int
    square :: Float -> Float
    square :: Char -> Char
    square :: Double -> Double

b)  == :: Int -> Int -> Bool
    == :: String -> String -> Bool
    == :: (Char, Bool) -> (Char, Bool) -> Bool
    == :: (Int -> Int) -> (Int -> Int) -> Bool
```

Exercise 3

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

```
foldl f z [] = z
foldl f z (x:xs) = foldl f (f z x) xs
```