```haskell
-- Model solution for Tut06
--
-- Copyright [2000..2004] Manuel M T Chakravarty

module Tut06
where

-- Given a nested list (ie, list of lists) of integers, yield a list that
-- contains the largest element of each of the sublists (and zero for empty
-- sublists)
--
-- Example: largest [[1,2,3], [2,3,1], [4,0], []] = [3,3,4,0]
--
largest :: [[Int]] -> [Int]
largest []       = []
largest (xs:xss) = largerThan 0 xs : largest xss
                   where
                     largerThan k []     = k
                     largerThan k (x:xs)
                       | x > k     = largerThan x xs
                       | otherwise = largerThan k xs

-- Delete all occurences of the given value from a list
--
-- Example: delete 'x' "x-files sux" = "-files su"
--
delete      :: Eq a => a -> [a] -> [a]
delete x ys  = filter isNotX ys
               where
                 isNotX y = x /= y

-- This version uses a higher-order function and a section; this is the
-- version that a Haskell expert would write
--
delete'      :: Eq a => a -> [a] -> [a]
delete' x ys  = filter (x /=) ys

-- Substitute elements in a list
--
-- Example: substitute 'e' 'i' "eigenvalue" = "iiginvalui"
--
substitute        :: Eq a => a -> a -> [a] -> [a]
substitute x y zs  = map subst zs
                     where
                       subst z | x == z    = y
                               | otherwise = z
```