

Exercise 10

```
-- write elem
-- type of elem' :: Eq a => a -> [a] -> Bool
-- i think we can use fold to write elem
elem' :: Eq a => a -> [a] -> Bool
elem' x [] = False
elem' x (hd:tl) =
    if x == hd then True else elem' x tl
-- map left fold use more time to compute because it need to run
-- all element in the list
-- but map right fold use more memory

-- write partition with fold (copy from the slide)
partition' p = foldr (select' p) ([], [])
select' p x (l, r)
    | p x      = (x:l, r)
    | otherwise = (l, x:r)
```