Exercises
Higher-order functions

1. Show how the list comprehension

   ```
   [f x | x <- xs, p x]
   ```

   can be re-expressed using the higher-order functions ***map*** and ***filter***.

2. Without looking at the definitions from the standard prelude, define the following higher-order library functions on lists.

   (a) Decide if all elements of a list satisfy a predicate:

   ```
   all :: (a -> Bool) -> [Bool] -> Bool
   ```

   (b) Decide if all elements of a list satisfy a predicate:

   ```
   any :: (a -> Bool) -> [Bool] -> Bool
   ```

   (c) Select elements from a list while they satisfy a predicate:

   ```
   takeWhile :: (a -> Bool) -> [a] -> Bool
   ```

   (d) Remove elements from a list while they satisfy a predicate

   ```
   dropWhile :: (a -> Bool) -> [a] -> Bool
   ```

3. Redefine the functions

   ```
   map f
   ```

   and

   ```
   filter p
   ```

   using

   ```
   foldr
   ```