HW 18 **Prof. Caldwell**
Due: 9 November 2006 **COSC 3015**

Exercise 0.1. Write Haskell code to implement `ordered` which tells whether a list is ordered or not.

```
ordered :: (Ord a) => [a] -> Bool
```

Exercise 0.2. Write Haskell code to implement `partition` using the `foldr` function.

```
partition :: (a -> Bool) -> [a] -> ([a],[a])
```

The specification[1] for `partition` is as follows:

```
partition p l = (filter p l, filter (not . p) l)
```

Recall that `(f . g)` denotes function composition, *i.e.* `(f . g)(x) = f(g(x))`.

[1]This means that this equation characterizes the `partition` function, *i.e.* for any implkementation of partition, this property must hold.