HW 9 **Prof. Caldwell**
Due: 23 September 2010 **COSC 3015**

1

Exercise 1.1. Write the function `remove :: (Eq a) => a -> [a] -> [a]` where the expression `(remove x xs)` results in a list that is like the list `xs` except that the leftmost occurrence of `x` has been removed.

```
*Main> :t remove
remove :: (Eq a) => a -> [a] -> [a]
*Main> remove 1 []
[]
*Main> remove 1 [1]
[]
*Main> remove 1 [1,1]
[1]
*Main> remove 1 [2,1,3]
[2,3]
*Main> remove 1 [2,1,3,1]
[2,3,1]
```

Exercise 1.2. Write the function `remove_all :: (Eq a) => a -> [a] -> [a]` so that the result of evaluating the expression `(remove_all x xs)` is the list like `xs` where *all* the elements `(==x)` have been removed.

```
*Main> :t remove_all
remove_all :: (Eq a) => a -> [a] -> [a]
*Main> remove_all 1 []
[]
*Main> remove_all 1 [1]
[]
*Main> remove_all 1 [1,1]
[]
*Main> remove_all 1 [1,2,1,3]
[2,3]
```

Exercise 1.3. Write the function `unique:: (Eq a) => [a] -> [a]` which keeps the first occurrence of each unique value encountered in a list and removes the rest. The resulting list should not contain any duplicate elements. You may find `remove_all` useful in writing this function.

```
*Main> :t unique
unique :: (Eq a) => [a] -> [a]
*Main> unique []
[]
*Main> unique [1]
[1]
*Main> unique [1,1]
[1]
```

```
*Main> unique [1,1,1]
[1]
*Main> unique [1,1,1,2]
[1,2]
*Main> unique [2,1,1,1]
[2,1]
*Main> concat [[x,y] | x <-[1..5],y<- [1..5]]
[1,1,1,2,1,3,1,4,1,5,2,1,2,2,2,3,2,4,2,5,3,1,3,2,3,3,3,4,3,5,4,1,4,2,4,3,4,4,4,5,5,1,5,2,5
*Main> unique (concat [[x,y] | x <-[1..5],y<- [1..5]])
[1,2,3,4,5]
```

Exercise 1.4. Write the function `count :: (Num t, Eq a) => a -> [a] -> t` where `(count x xs)` evaluates to the number of occurrences of the value `x in the list xs.`

```
*Main> :t count
count :: (Num t, Eq a) => a -> [a] -> t
*Main> count 'a' []
0
*Main> count 'a' ['a'..'z']
1
*Main> count 'a' "xyzzy"
0
*Main> count 'a' "a big brown dog jumped over the lazy cat"
3
*Main> count ' ' "a big brown dog jumped over the lazy cat"
8
*Main> count 1 [1,2,2,3,3,3]
1
*Main> count 2 [1,2,2,3,3,3]
2
*Main> count 3 [1,2,2,3,3,3]
3
```