Code examples from 9/14 — recursion, lists

Pattern-matching on values

```
vowel :: Char -> Bool
vowel 'a' = True
vowel 'e' = True
vowel 'i' = True
vowel 'o' = True
vowel 'u' = True
vowel  _  = False
```

Count the length of a collatz sequence starting from `n`. See also the assignment 1 solutions. Note that those are *backquotes* on the `` `div` `` – some of you are getting that wrong. Also, pay attention to the difference between `` `div` `` (integer division) and `/` (float, or fractional division).

```
collatzCount :: Int -> Int
collatzCount n
  | n <= 1 = 0
  | even n = collatzCount (n `div` 2) + 1
  | otherwise = collatzCount (3*n+1) + 1
```

A really simple function that works on any numeric type (class `Num`).

```
squared x = x*x
```

A main function, using the `do` notation and output statements (side effects).

```
main = do
  putStrLn "Hello there"
  putStrLn "Haskell is fun"
```

Generating lists recursively

Produce a list with `count` copies of a given value `elem`.

```
genList :: a -> Int -> [a]
genList elem count
  | count == 0 = []
  | otherwise = elem : genList elem (count-1)
```

Examples:

- `genList 3 5` ⇒ `[3,3,3,3,3]`
- `genList 'a' 8` ⇒ `"aaaaaaaa"`
- `genList 'x' 0` ⇒ `""`

Produce a range of integers (much like the built-in notation `[3..8]` which is *syntactic sugar* for a function `enumFromTo`).

```
fromTo :: Int -> Int -> [Int]
fromTo start end
  | end < start = []
  | otherwise = start : fromTo (start+1) end
```

Examples:

- `fromTo 3 8` ⇒ `[3,4,5,6,7,8]`
- `fromTo 10 14` ⇒ `[10,11,12,13,14]`
- `fromTo 10 8` ⇒ `[]`

Duplicate each element of the list.

```
dupe :: [a] -> [a]
dupe [] = []
dupe (h:t) = h : h : dupe t
```

Examples:

- `dupe [3,4,5]` ⇒ `[3,3,4,4,5,5]`
- `dupe []` ⇒ `[]`
- `dupe "Chris"` ⇒ `"CChhrriiss"`

Square each element of a list, using direct recursion. (A more idiomatic approach would be to use `map`.)

```
squares [] = []
squares (h:t) = h*h : squares t
```

Examples:

- `squares [3,4,5]` ⇒ `[9,16,25]`
- `squares [2,10,8]` ⇒ `[4,100,64]`
- `squares []` ⇒ `[]`