Exercises
List Comprehensions

* Exercise 1

Using list comprehension define the list of cubes of the values between (and including) 1 and 10.

```
cubes = [(1,1),(2,4),(3,9),(4,16),(5,25),(6,36),
                (7,49),(8,64),(9,81),(10,100)]
```

* Exercise 2

Using list comprehension define the following list (note that the second element in the 2-tuple is always 1.

```
myConstFunc = [(1,1),(2,1),(3,1),(4,1),(5,1)]
```

** Exercise 3

Write down the values as defined in the following lists l1, l2, l3. Check your answers.

```
f1 :: [(Int, Int)]
f1 = [(x, y) | x <-[1..3], y<- [4..5]]

f2 :: [(Int, Int)]
f2 = [(x, y) | y<- [4..5], x <-[1..3]]

f3 :: [(Int, Int)]
f3 = [(y, x) | x <-[1..3], y<- [4..5]]
```

** Exercise 4

Given the following definition of

```
isEven :: Integer -> Bool
isEven n = (n `mod` 2 == 0)
```

Write down the values as defined in the following list: Check your answer.

```
[2*n | n <- [2,4,7], isEven n, n>3]
```

** Exercise 5

Give a definition of a function

```
doubleAll :: [Integer] -> [Integer]
```

which doubles all the elements of a list of integers.

** Exercise 6

Give a definition of a function

```
capitalize :: String -> String
```

which converts all small letters in a String into capitals.
Hint:You can use the following function (having imported Data.Char):

```
import Data.Char
toupper :: Char -> Char
```