CS 5035 (Fall 2016)

### Project 5b. Higher-order functions (part 2) (first attempt by Oct 10).

Now that you understand the solution from last week:

1. Look up uncurry and divMod and explain [this solution](https://www.rosettacode.org/wiki/Luhn_test_of_credit_card_numbers#Haskell).
2. That solution assumes that the credit card number is represented as a String rather than an Integer. That’s a bit of a cheat. Fix it so that it takes an Integer as input.
3. Replace map (uncurry (+) . (`divMod` 10)) with the approach used above, i.e., an expression that uses toDigits and concat. In other words, write:

myLuhn :: Integer -> Bool

myLuhn = <something similar to luhn but without uncurry and divMod>