**Week6 Assignment**

{-

1. Define a function dropOdds :: Int -> Int with the following behaviour.

For any positive number m, dropOdds m is got by dropping all the odd digits

in m. (If all the digits in the number are odd, the answer should be 0.)

Test cases:

dropOdds 0 = 0

dropOdds 8 = 8

dropOdds 1357 = 0

2. Define a function moreZeros :: Int -> Bool such that moreZeros n returns

True exactly when the binary representation of n has strictly more 0s than 1s.

Test cases:

moreZeros 0 = True

moreZeros 1 = False

moreZeros 2 = False

moreZeros 4 = True

3. Define a function binToTer :: Int -> Int which takes as input the binary

representation of a number n and outputs the ternary representation of n.

(You can assume that the input consists only of the digits 0 and 1, and the

output should only consist the digits 0, 1 and 2.)

Test cases:

binToTer 0 = 0

binToTer 1 = 1

binToTer 11 = 10

binToTer 100 = 11

4. Define a function palindrome :: Int -> Bool which outputs True exactly when

the number is a palindrome (digits read from left to right is the same as

digits read from right to left).

Test cases:

palindrome 0 = True

palindrome 121 = True

-}

dropOdds :: Int -> Int

dropOdds 0 = 0

dropOdds x

| odd (x `mod` 10) = dropOdds (x `div` 10)

| otherwise = dropOdds (x `div` 10) \* 10 + x `mod` 10

moreZeros :: Int -> Bool

moreZeros x = check0n1 x 0 0

where check0n1 0 0 0 = True

check0n1 0 z o

| z > o = True

| otherwise = False

check0n1 x z o

| even (x `mod` 2) = check0n1 (x `div` 2) (z+1) o

| otherwise = check0n1 (x `div` 2) z (o+1)

binToTer :: Int -> Int

binToTer x = foldr k 0 (decToTer ( binToDec x 0))

where k e acc = acc\*10 + e

binToDec 0 \_= 0

binToDec x i = round (2\*\*i) \* (x `mod` 10) + binToDec (x `div` 10) (i+1)

decToTer 1 = [1]

decToTer 0 = [0]

decToTer 2 = [2]

decToTer x = x `mod` 3:decToTer (x `div` 3)

palindrome :: Int -> Bool

palindrome x

|check x 0 == x = True

|otherwise = False

where check 0 rev = rev

check x rev = check (x `div` 10 ) (rev\*10 + x `mod` 10)