

HW2 – DUE 10 AM 23.JAN.2013

Instructions:

Place all of the definitions into a single file named HW2_[lastname].hs and drop it into your Assignments folder on Google Drive. Please document your code.

1. Write a Haskell function **isaPalindrome** to determine whether a list is a palindrome. A palindrome can be read forward or backward; e.g. [x a m a x].
isaPalindrome :: (Eq b) => [b] -> Bool
2. Write a Haskell function **myFlatten** to transform a list, possibly holding lists as elements into a 'flat' list by replacing each list with its elements (recursively). Thus [a, [b, [c, d], e]] becomes [a, b, c, d, e]. myFlatten :: ListOfLists b -> [b] Hint: define a data type ListOfLists.
3. Write a Haskell function **myCompress** that eliminates consecutive duplicates of list elements. Thus, "aaabbbccc" becomes "abc". myCompress :: Eq b => [b] -> [b]
4. Write a Haskell function **myPack** that places consecutive duplicates of list elements into sublists. Thus "aaabbbccc" becomes ["aaa","bbb","ccc"]. myPack :: Eq b => [b] -> [[b]]
5. Write a Haskell function **myEncode** that places consecutive duplicates of list elements into pairs (n, e) where n is the number of element e. Thus "aaabbbccc" becomes [(3, 'a'), (3, 'b'), (3, 'c')]. myEncode :: Eq b => [b] -> [(Int, b)]