ASSIGNMENT-3

Read about maps, folds, lambda functions and filters from the slides or from other sources online. Particularly try to understand how the foldl and foldr functions work. Then solve the following questions:

1. Using the function foldr define a function sumsq which takes an integer n as its argument and returns the sum of the squares of the first n integers. For example, sumsq 3 = 14 . Do not use the function map.

2. Understand the following implementation of foldl

```
myFoldl :: (a -> b -> a) -> a -> [b] -> a
myFoldl f z xs = foldr step id xs z
                where step x g a = g (f a x)
```

3. Implement map and filter using folds

4.Try to understand why the command ‘*foldr const 0 [1..]’* works when executed on ghci but the command ‘*foldl const 0 [1..]’* doesn’t work. ‘*const’* is an inbuilt function in haskell.

5. Define a function reverse which reverses a list using foldr and foldl.

6. The function remdups removes adjacent duplicates from a list. For example, remdups [1,2,2,3,3,3,1,1] = [1,2,3,1]. Define remdups using foldr and foldl.

7. The function inits returns the list of all initial segments of a list. Thus, inits "ate" = [“ ”, "a", "at", "ate"]. Define inits using foldr.