Extra foldr/foldl practice:

What do the following calculate? Write your answer below; if any result in an error, write “error”. Try to do these on your own on paper, and use ghci to check your work. If you get it wrong, figure out why that is before moving on – these can get pretty tricky! HINT: you can check out this link for foldr and this for foldl to see a pretty handy visual on how the two functions derive their results.

You should be able to solve problems similar to a) – f) on this handout for exams in this course. Questions g) – j) are beyond what we’d expect you to work out by hand for an exam, but they’re great practice (and show some really crazy behavior!)

a. `foldr (-) 0 [8,7,6,5]`

b. `foldl (-) 0 [8,7,6,5]`

c. `foldr (:) [] [1,2,3,4,5]`

d. `foldl (:) [] [1,2,3,4,5]`

e. `foldr (/) 0 [1, 2, 3, 4, 5]`

f. `foldl (/) 0 [1, 2, 3, 4, 5]`

The following involve function composition; remember f . g x is the same as f(g(x)), where f and g are functions:

g. `foldr ((++) . map (* 2)) [] [[1,2,3],[4,5,6],[7,8,9]]`
(This one does what you might expect)

h. `foldl ((++) . map (* 2)) [] [[1,2,3],[4,5,6],[7,8,9]]`
(This one does something really bizarre)

i. `foldr ((++) . reverse) [] ["Spyro", "the", "Dragon"]`

j. `foldl ((++) . reverse) [] ["Spyro", "the", "Dragon"]`