PROGRAMMING IN HASKELL

Chapter 8.3
zipWith function

zipWith

zipWith takes a function and two lists as parameters and then joins the two lists by applying the function between corresponding elements. Here's how we'll implement it*:

```
zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]
zipWith _ [] _ = []
zipWith _ _ [] = []
zipWith f (x:xs) (y:ys) = f x y : zipWith f xs ys
```

zipWith

```
ghci> zipWith (+) [4,2,5,6] [2,6,2,3]
    = [6,8,7,9]

ghci> zipWith max [6,3,2,1] [7,3,1,5]
    = [7,3,2,5]

ghci> zipWith (++) ["foo ", "bar ", "buzz "]
                        ["fighters", "hoppers", "aldrin"]
    = ["foo fighters", "bar hoppers"," buzz  aldrin"]
```

zipWith

```
ghci> zipWith (*)    (replicate 5 2) [1..]
        [2,4,6,8,10]

ghci> zipWith (zipWith (*))
                [[1,2,3],[3,5,6],[2,3,4]]
                [[3,2,2],[3,4,5],[5,4,3]]

        [[3,4,6],[9,20,30],[10,12,12]]
```

zipWith

Note that you can use function application as the function

```
zipWith ($) funcList valueList


zipWith ($) [(+ 5),(* 3)] [1,5]
    = [6,15]
```

Aside on zipWith

Note that you can use function application as the function

```
zipWith ($) funcList valueList


zipWith ($) [(+ 5),(* 3)] [1,5]
    = [6,15]
```

ANY
QUESTIONS?