Problem Sheet 6.1: Folds and Recursion

In this tutorial we will look at **folds** over lists, in combination with `map` and `filter`.

Exercise 1:

Library functions that you might want to use are: `and`, `or`, `any`, `all`, `maximum`, `minimum`, `product`, `sum`, `concat`. If you don't know some of them, look up the type and try them out.

In any of the assignments you may use basic functions such as `(==)`, `(<=)`, `(&&)`, `(||)`, `(++)`, `even`, `odd`, `max`, `min`, `head`, `tail`, and `length`.

a) The function `allTrue`, given a list of booleans, returns whether all of them are `True`. Write a function `allTrue2` which uses `foldr` but performs in exactly the same way as `allTrue`.

b) The function `longestLength`, given a non-empty list of lists, should return the length of the longest list. Write it first in recursive style (like `allTrue`), then write another function `longestLength2` using `foldr`.

c) We have given you the function `sumOddSquares`. Work out what it does (the name should help!) and implement it again using `foldr`.

d) The function `shortFWords`, given a list of non-empty strings, should return `True` if any of them is a four-letter word starting with capital letter `'F'`, `False` otherwise. Write this first in recursive style, then with a list comprehension, and finally using `foldr`.

```
*Main> allTrue [False,True,False]
False

*Main> allTrue []
True

*Main> longestLength ws
9

*Main> sumOddSquares [1..100]
166650

*Main> shortFWords ["Fish","for","breakfast","??"]
True
```