Math 6833 assignments

28. Write functions that do the following.

    1. `shuffle :: [a] -> [a] -> [a]`, that has the effect
       `shuffle [x1,x2,x3,x4] [y1,y2,y3,y4] = [x1,y1,x2,y2,x3,y3,x4,y4]`
       Write a version `robustShuffle` that just drops extra terms, but write `shuffle` to allow either the lengths to be equal or the first string to have an extra term, as in
       `shuffle [x1,x2,x3] [y1,y2] = [x1,y1,x2,y2,x3]`
       but to give an error message otherwise. Use `robustShuffle` in the definition of `shuffle`, if you wish.
    2. `oddTerms :: [a] -> [a]` returns a list of the first term, third term, fifth term, etc. `oddTerms [2,4,6,8,10] = [2,6,10]`. Similarly for `evenTerms`.

29. Determine what the following function does:
```
mystery list1 list2 = concat ( zipWith (\x y -> [x,y]) list1 list2 )
```

30. The Prelude function `takeWhile` is defined by:
```
takeWhile p [] = []
takeWhile p (x:xs)
  | p x = x :  takeWhile p xs
  | otherwise = [ ]
```

    1. Determine the type of `takeWhile`
    2. Use `takeWhile` to define functions `takeWord` and `removeWord` that take or remove the initial non-blank characters of a string, for example
       `takeWord "first test string" = "first"`
       `takeWord " test string" ""`
       `removeWord " test string" = " test string"`
       `removeWord "first test string" = " test string")`
       Define corresponding functions `takeBlanks` and `removeBlanks`.
    3. Using these auxiliary functions to define the following functions:
       (a) A function to split a string into its words:
       `splitString "A list of characters."`
       `= ["A","list","of","characters."]`
       (b) A function to split a string into pairs consisting of its words and the number of blanks that follow them:
       `splitStringWithBlanks "A list of characters.  End." =`
       `[("A",1),("list",1),("of",1),("characters.",2),("End.",0)]`
       `splitStringWithBlanks "  Another." = [("",2),("Another.",0)]`
    4. Define the inverse of `splitStringWithBlanks`.