Implement the following functions using higher-order functions (and other predefined functions) of Haskell without using recursion.

1. Implement a function `flatten :: [Int] → Int` that flattens a list of lists of integers in a list of integers.

2. Implement a function `myLength :: String → Int` that returns the length of a string.

3. Implement a function `myReverse :: [Int] → Int` that reverses a list of integers.

4. Implement a function `countIn :: [Int] → Int → [Int]` that, given a list of sublists `ℓ` and an element `x`, returns the list that tells who many times `x` appears in each sublist of `ℓ`.

5. Implement a function `firstWord :: String → String` that, given a string with blanks and alphabetic characters, returns its first word.

**Scoring**

Each function scores 20 points.

**Sample input**

```haskell
flatten [[1,2,3],[4,5],[6],[],[3,3]]
myLength "Albert"
myReverse [1..10]
countIn [[3,2,3],[3],[],[2,2]] 3
firstWord " Volem pa amb oli "
```

**Sample output**

```
[1,2,3,4,5,6,3,3]
6
[10,9,8,7,6,5,4,3,2,1]
[2,1,0,0]
"Volem"
```

**Problem information**

Author: Albert Rubio / Jordi Petit
Translator: Jordi Petit
Generation: 2016-01-19 11:29:21

http://www.jutge.org