

Zadanie 1 (bardzo łatwe, na rozgrzewkę)

Napisać, używając funkcji `foldl/r`, funkcję

`bin2Int :: [Int] -> [Int]`,

która zamienia liczbę zapisaną w postaci binarnej na jej wartość dziesiętną. Na przykład `bin2Int [1,1,0,0,1] = 25`.

Zadanie .

Napisać funkcję

- ▶ `insertElement :: Ord a => a -> [a] -> [a]`, który wstawia element na "właściwe" miejsce do listy. "Właściwe" = takie, że utworzona lista reprezentuje ciąg posortowany (np. rosnąco).
- ▶ `insertionSort :: Ord a => [a] -> [a]`, która sortuje podaną listę przez wstawianie.

W rozwiązaniu należy (przynajmniej raz) w istotny sposób użyć funkcji `foldl/r`.

Zadanie (Kolokwium 2013/14)

Napisać funkcję

`diffsums :: [[Int]] -> [[[Int]]]`,

która z wejściowej listy list usuwa listy o powtarzającej się sumie.

Na przykład

`diffsums [[1,2],[3,4,5],[3],[],[7,5]] =`
`[[1,2],[3,4,5],[]]` (w odpowiedzi mogą też być inne listy).

Każda suma z wejściowej listy ma być reprezentowana w liście wynikowej przez dokładnie jedną listę. W rozwiązaniu należy użyć funkcji `foldl/r`. Kolejność w liście wynikowej nie ma znaczenia, ale kolejność w blokach ma zostać zachowana.

Zadanie (Kolokwium 2016/17).

Napisać funkcję `ps`, która dla podanej listy zwraca listę zawierającą wszystkie jej prefiksy – w kolejności od najkrótszego (jednoelementowego) do najdłuższego (cała lista) – a następnie kolejno coraz krótsze sufiksy. Przykładowo

```
ps "Test" = ["T", "Te", "Tes", "Test", "est", "st", "t"].
```

Zadanie (bardzo łatwe, na koniec).

Napisać funkcję `sep`, która dla wywołania `sep list element` rozdzieli elementy `list` przy użyciu separatora `element`. Na przykład

```
sep "ala i pies" ' ' = ["ala","i","pies"].
```