

## Programowanie funkcyjne — kolokwium nr 1, 19.12.2013

**Instrukcja:** Rozwiązania zadań należy przesłać do godziny 11:40 na adres `kolokwium.pf@gmail.com` (decyduje data stempla googlowego). Każde zadanie należy przesłać w oddzielnym pliku `Zadanie $n$ .hs` ( $n = 1, 2, 3$ ). W rozwiązaniach nie można korzystać z modułów innych niż `Prelude`, tzn. niedozwolone jest użycie polecenia `import`. Punktacja: po 10 punktów za zadanie.

**Zadanie 1.** Napisać funkcję `oddbins  $n$` , generującą listę wszystkich ciągów binarnych o długości  $n$ , w których liczba jedynek jest nieparzysta. Ciągi reprezentujemy w postaci list, zatem np.

`oddbins 3 = [[1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 1, 1]].`

Kolejność ciągów w liście nie ma znaczenia. Zakładamy, że  $n \geq 1$ .

**Zadanie 2.** Napisać funkcję `diffsums :: [[Int]] → [[Int]]`, która z wejściowej listy usuwa listy o powtarzającej się sumie. Na przykład

`diffsums [[1, 2], [3, 4, 5], [3], [], [7, 5]] = [[1, 2], [3, 4, 5], []]` lub `[[7, 5], [3], []]` itp.

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

**Zadanie 3.** Napisać bezpunktowo funkcję `compref`, podającą długość najdłuższego zgodnego odcinka początkowego dwóch list, oraz podać najogólniejszą możliwą sygnaturę. Na przykład

`compref [1, 2, 3, 4, 5] [1, 2, 3, 0, 0] = 3`  
`compref [1, 2, 3] [9, 9, 9, 9, 9] = 0.`

Wskazówka: Można korzystać ze standardowych funkcji listowych, w tym z funkcji `zip`, która z dwóch list tworzy listę par, np. `zip [1, 2] ['a', 'b', 'c'] = [(1, 'a'), (2, 'b')]`.