

Писмен изпит по Функционално програмиране

спец. Информационни системи, 05.07.2012 г.

Вариант А

Задача 1. Да се напише функция `sumMax :: [[Int]] -> Int`, която по матрица от цели числа намира най-голямата сума на колона в матрицата.

Пример: `sumMax [[2,5,3], [12,4,3], [6,9,11]]` \rightarrow 20 (= 2 + 12 + 6)

Задача 2. Да се напише функция `findMiddle :: [Int] -> Int`, която по списък от числа намира това число, за което броят на числата в списъка, от които то е по-голямо, съвпада с броя на числата в списъка, от които то е по-малко. Ако такова число не съществува, да се върне кое да е число от списъка.

Пример: `findMiddle [5,2,9,4,2,1,6,6,5,7]` \rightarrow 5 ($> \{4,2,2,1\}$, $< \{6,6,7,9\}$)

Упътване: напишете функция, която за всеки елемент на списъка намира броя на числата, по-големи и по-малки от него.

Задача 3. Хранителен продукт се описва с типа `type Food = (String, Int)`, като наредената двойка задава името на продукта и броя оставащи дни от срока на годност. Да се напише функция `discountFood :: [Food] -> String`, която по даден списък от поръчки намира името на продукта, който има най-малко оставащи дни до края на срока на годност, за да може този продукт да бъде пуснат на промоция. Продуктите, които са с минал срок на годност, не се вземат предвид.

Пример: `discountFood [(“Fish”, 3), (‘Cheese”, 5), (‘Canned beans”, 200), (‘Yoghurt”, 4), (‘Bread”, 1), (‘Cream”, -2)]` → “Bread”

Упътване: напишете функция, която намира най-малкия неотрицателен брой оставащи дни от всички продукти в списъка.

Задача 4. Да се напише функция `twins`, която за дадено двоично дърво от цели числа намира списък с всички възли, които имат близък, т.е. брат/сестра със същата стойност. Опишете избраното от вас представяне на дървото и напишете как се представя дървото, написано на дъската.

Писмен изпит по Функционално програмиране

спец. Информационни системи, 05.07.2012 г.

Вариант Б

Задача 1. Да се напише функция `prodMin :: [[Int]] -> Int`, която по матрица от цели числа намира най-малкото произведение на колона от матрицата.

Пример: `prodMin [[2,5,3], [12,4,3], [6,9,11]]` \rightarrow 99 (= 3 * 3 * 11)

Задача 2. Да се напише функция `sumLess :: [Int] -> Int`, която по списък от цели числа намира такова число, което е точно равно на сумата на всички по-малки от него числа в списъка. Ако такова число няма, да се върне кое да е от числата от списъка.

Пример: `sumLess [5,2,1,2,5,8]` \rightarrow 5 (= 2 + 1 + 2)

Упътване: напишете функция, която намира сумата на всички числа от даден списък, по-малки от дадено число.

Задача 3. Задача от изпит се описва с типа `type Problem = (String, Int)`, като наредената двойка задава условието на задачата и броя точки, които носи. Да се напише функция `trickiestProblem :: [Problem] -> String`, която по списък от задачи намира тази от тях, която дава най-много точки пропорционално на дължината на описанието ѝ, т.е. дава най-много точки за буква от описанието ѝ.

Пример: `trickiestProblem [(“2+2”, 4), (“sqrt(10)”, 8), (“P=NP?”, 7)]` → “P=NP?” (дава 1.4 точки на буква).

Упътване: напишете функция, която намира за всяка от задачите колко точки дава на буква от условието.

Задача 4. Напишете функция `cousins`, която за дадено двоично дърво от цели числа и даден възел намира всички възли, които са негови братовчеди, т.е. имат общ дядо/баба с дадения възел. Опишете избраното от вас представяне на дървото и напишете как се представя дървото, написано на дъската.