

Домашна работа № 3 по Функционално програмиране
Специалност „Информационни системи“, 1-ви курс
2018/2019 учебна година

Крайният срок за предаване на домашните работи е **27.05.2019 г. (понеделник), 23:55 ч.**

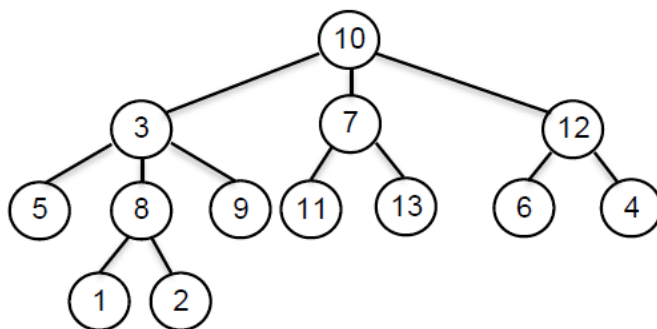
Решенията трябва да са готови за компилиране и автоматично тестване. Важно е писмените работи да бъдат добре форматираны и да съдържат коментари на ключовите места.

Предайте решенията на двете задачи в един файл с наименование **hw3_<FN>.hs**, където **<FN>** е Вашият факултетен номер.

Приятна работа и успех!

Задача 1. Дадено е дърво **tree** от цели числа, представено с асоциативен списък, описващ преките наследници (синовете) на върховете, които не са листа. Да се дефинира функция (**numOfNodes tree**), която намира броя на вътрешните върхове **node** на **tree**, за които сумата на синовете на **node** е равна по стойност на родителя на **node**.

Пример:



(numOfNodes [(10, [3, 7, 12]), (3, [5, 8, 9]), (7, [11, 13]), (12, [6, 4]), (8, [1, 2])]) → 2 (върховете 12 и 8)

Задача 2. Температурно измерване се описва с типа **data Measuring = Temp Int Float**, където стойността от тип **Int** задава ден от месеца, а стойността от тип **Float** – измерена температура за този ден. Да се дефинира функция **closestToAverage :: [Measuring] -> Int**, която по списък от температурни измервания намира деня, в който измерената температура е най-близо до средната температура през месеца.

Пример:

closestToAverage [(Temp 1 23.6), (Temp 6 24.2), (Temp 11 24.2), (Temp 16 21.2), (Temp 21 23.8), (Temp 26 26.5), (Temp 31 24.5)] → 6 или 11 или 21
(средната температура е 24.0)

Задача 3. Нека за представянето на двоично дърво от цели числа се използва алгебричен тип със следната дефиниция:

data BTree = Empty | Node Int BTree BTree .

Да се дефинира функция (**grandchildrenIncreased tree**), която проверява дали всеки връх на двоичното дърво **tree** е поне с единица по-голям от своя дядо (ако има такъв).