

Домашна работа № 3 по Функционално програмиране
Специалност Информационни системи, 1-ви курс
2016/2017 учебна година

Домашното включва три задачи. *Крайният срок* за предаване на решенията е **понеделник, 15.05.2017 г.**

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

Предайте решенията на трите задачи в един файл с наименование **hw3.<FN>.hs**, където **<FN>** е Вашият факултетен номер.

Приятна работа и успех!

Задача 1. Казваме, че две функции **f** и **g** са строго монотонно растящи в целочисления интервал **[a,b]**, ако за всяко цяло число **x** в интервала **[a,b]** е изпълнено **g (f (x)) > x** и **f (g (x)) > x**.

Да се дефинира функция (**increasing f g a b**), която получава като аргументи функциите **f** и **g** и числата **a** и **b** и връща **True**, ако функциите са строго монотонно растящи в интервала **[a,b]**, или **False** – в противния случай. Може да се предположи, че **a ≤ b**.

Примери:

```
increasing (\x -> x * 2) (\x -> x) 0 1 → False
increasing (\x -> x * 2) (\x -> x + 1) 1 5 → True
```

Задача 2.

а) Да се дефинира функция (**distance f xys**), която получава като аргументи едноместна функция **f** и списък **xys** от двойки от вида **(x,y)** и връща сбора от разстоянията **|y - f(x)|** между **f(x)** и **y** за всички елементи на списъка **xys** (в случая **|a|** означава абсолютната стойност на числото **a**).

Ще наричаме резултата на тази функция „разстояние на функцията **f** от **xys**“.

Примери:

```
distance (\x -> x ^ 2) [(1,1), (2,4), (3,9), (4,15)] → 1
distance (\x -> x + 1) [(1,1), (2,4), (3,9), (4,15)] → 17
```

б) Да се дефинира функция (**closest fs xys**), която получава като аргументи списък от едноместни функции **fs** и списък от двойки **xys** и връща функцията от **fs**, която е на най-малко разстояние от **xys**.

Пример:

```
(closest [(\x -> x ^ 2), (\x -> x + 1)]
         [(1,1), (2,4), (3,9), (4,15)]) 5 → 25
```

Задача 3. Да се дефинира функция (**replaceAssoc list dict**), която получава като аргументи списък **list**, чийто елементи са цели числа, и речник – асоциативен списък **dict**, чийто елементи са двойки от цели числа. Функцията трябва да върне нов списък, в който всеки елемент се получава чрез замяна на съответния елемент на **list** с асоциираната с него стойност в **dict**, ако в **dict** съществува елемент с такъв ключ, или е равен на съответния елемент на **list** – в противен случай.

Пример:

(replaceAssoc [5,4,2,3] [(1,5), (3,7), (5,9), (7,11), (9,13)]) →
[9,4,2,7]