

Домашна работа № 2 по Функционално програмиране
Специалност „Информационни системи“, 1-ви курс
2018/2019 учебна година

Крайният срок за предаване на домашните работи е **07.05.2019 г., 23:55 ч.**

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 всички задачи в един файл с наименование **hw2_<FN>.hs**, където **<FN>** е Вашият факултетен номер.

Приятна работа и успех!

Задача 1. Да се дефинира функция (**pairCompose fs**), която получава като аргумент списък [**f1,f2,f3, ... ,fn**] с функции от тип **Int -> Int** и връща нова едноаргументна числова функция **g** – такава, че оценката на (**g x**) е равна на сумата (**f1.f2**) (**x**) + (**f3.f4**) (**x**) + ... + (**fn-1.fn**) (**x**), където “.” е операторът за композиция на функции. Ако оригиналният списък с функции има нечетен брой елементи, то последната функция от списъка се композира с функцията *идентитет* (**id**).

Пример:

```
(pairCompose [(\x -> x+1), (\x -> x+2), (\x -> x+3)]) 1 ->
((1+2)+1)+(1+3) = 8
```

Задача 2. Ако **f** и **g** са числови функции и **n** е естествено число, да се дефинира функция от по-висок ред **switchsum f g n**, която връща като резултат функция, чиято стойност в дадена точка **x** е равна на **f(x)+g(f(x))+f(g(f(x)))+...** (сумата включва **n** събираеми).

Примери:

```
switchsum (\x -> x + 1) (\x -> x * 2) 1 $ 2 -> 3
switchsum (\x -> x + 1) (\x -> x * 2) 2 $ 2 -> 9
switchsum (\x -> x + 1) (\x -> x * 2) 3 $ 2 -> 16
switchsum (\x -> x + 1) (\x -> x * 2) 4 $ 2 -> 30
```

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]
```