Домашна работа № 2 по Функционално програмиране

специалност "Информационни системи", І курс, 2020/2021 учебна година

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран и да съдържа коментари на ключовите места. Предайте решенията на всички задачи в $e\partial uh$ файл с наименование $hw2_<FN>.hs$, където <FN> е Вашият факултетен номер.

Домашните работи се предават като изпълнение на съответното задание в курса по ФП в Moodle (https://learn.fmi.uni-sofia.bg/course/view.php?id=7081) най-късно до 23:55 ч. на 11.04.2021 г. (неделя).

Приятна работа и успех!

Задача 1

Хамелнският ловец на плъхове (на немски: Rattenfänger von Hameln) е немска народна приказка, в която се разказва как през XVI век град Хамелн се справил с нашествие от плъхове. За целта жителите наели флейтист, който бил известен като ловец на плъхове - той ги привличал с вълшебната си флейта и ги отвеждал далеч от града. В град Хамелн обаче някои от плъховете били глухи и се движели в грешната посока.

Да се дефинира функция countRats:: String -> Int, която за даден символен низ, представящ в компактен вид движението на група плъхове, намира броя на плъховете, които се движат в грешна посока. В символния низ винаги ще присъства знакът 'P' и всички знакове в него ще бъдат коректни спрямо легендата (напр. няма да се подава като вход "2\$"). В низа може да присъстват произволен брой интервали (' ').

Легенда:

```
Р = ловецът на плъхове1 ( = плъх, който се движи наляво) 1 = плъх, който се движи надясно
```

Примери:

```
countRats ")1)1)1)1 Р" \to 0 -- всички плъхове се движат към ловеца countRats "Р 1( 1( )1 1(" \to 1 countRats " Р 1( 1( )1 1(" \to 1 countRats ")1)1)1)1)1)1(" \to 2 countRats "1()1)1)11(1()1)1P)11()1)11(1(1(1(" \to 7
```

Задача 2

Деца имат блокче шоколад и искат да си го разделят по равно. За целта всяко от тях си изтегля картонче, което го идентифицира еднозначно, след което всички се нареждат в кръг и започват да си предават блокчето шоколад. Право на отчупване от

блокчето има само всяко k-то дете, и то само по веднъж, т.е. след като си вземе от шоколада. Детето излиза от кръга.

Да се дефинира функция josephus:=[a] -> (Int -> [a]), която получава като аргумент списък от идентификатори на деца и връща едноместна функция на естествено число k, която връща списък от картончетата (идентификаторите) на децата в последователността, в която те ще излязат от кръга (която е еквивалентна на техния ред на отчупване).

Примери:

```
(josephus [1,2,3,4,5,6,7]) 3 \rightarrow [3,6,2,7,5,1,4]
(josephus [1,2,3,4,5,6,7]) (-1) \rightarrow \text{error "k was not natural"}
(josephus [1,2,3,4,5,6,7,8,9,10]) 1 \rightarrow [1,2,3,4,5,6,7,8,9,10]
(josephus [1,2,3,4,5,6,7,8,9,10]) 2 \rightarrow [2,4,6,8,10,3,7,1,9,5]
(josephus "fpFMIsu") 4 \rightarrow \text{"MfsIuFp"}
```

Обяснение за пример 1:

```
[1,2,3,4,5,6,7] => начало
[1,2,4,5,6,7] => 3 си отчупва и излиза от кръга [3]
[1,2,4,5,7] => 6 си отчупва и излиза от кръга [3,6]
[1,4,5,7] => 2 си отчупва и излиза от кръга [3,6,2]
[1,4,5] => 7 си отчупва и излиза от кръга [3,6,2,7]
[1,4] => 5 си отчупва и излиза от кръга [3,6,2,7,5]
[4] => 1 си отчупва и излиза от кръга [3,6,2,7,5,1]
[5] => 4 си отчупва и излиза от кръга [3,6,2,7,5,1,4]
```