

Домашна работа №2 по Функционално програмиране
Специалност “Информационни системи”, 2019/2020 учебна година

Крайният срок за предаване на домашните работи е **25.04.2020 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. Напишете функция `generate p n`, която приема реалното число `p` и естественото число `n` и формира списък от следната редица:

$$1, 1 + \frac{1}{2^p}, 1 + \frac{1}{2^p} + \frac{1}{3^p}, 1 + \frac{1}{2^p} + \frac{1}{3^p} + \frac{1}{4^p}, \dots, \sum_{m=1}^n \frac{1}{m^p}$$

Примери:

`(generate 1 3) → [1.0,1.5,1.8333333333333333]`

`(generate 0.1 5) → [1.0,1.93,2.83,3.70,4.55]`

Задача 2. Напишете функция `listSquares a b`, която приема две цели числа `a` и `b`, задаващи краища на интервал от цели числа. Функцията да връща списък от наредени двойки. Първият елемент на всяка двойка да е поредното число в затворения интервал `[a,b]`, което е точен квадрат, а вторият – сумата на квадратите на делителите на съответния първи елемент. Делителите да включват 1, както и самото число.

Примери:

`(listSquares 1 30) → [(1,1), (4,21), (9,91), (16,341), (25,651)]`

`(listSquares 250 300) → [(256,87381), (289,83811)]`

Задача 3. Нека е дефиниран типът `type Point = (Double, Double)`, представящ точка с реални координати. Да се дефинира функция `splitPoints :: Point -> Double -> [Point] -> ([Point], [Point])`, която приема точка `p`, радиус `r` и списък от точки `ps` и връща като резултат двойка от списъци, като първият съдържа тези точки от `ps`, които са в кръга с център `p` и радиус `r`, а вторият - всички останали точки от `ps`.

Примери:

```
(splitPoints (1,1) 5 [(1,2), (2,3), (10,15), (-1,1), (12,14)]) →
([(1.0,2.0), (2.0,3.0), (-1.0,1.0)], [(10.0,15.0), (12.0,14.0)])
```

```
(splitPoints (10,10) 5 [(1,2), (2,3), (10,15), (-1,1), (12,14)]) →
([(10.0,15.0), (12.0,14.0)], [(1.0,2.0), (2.0,3.0), (-1.0,1.0)])
```

Задача 4. Нека имаме тип сметка `Account`, който е вектор с три елемента (`Int, Int, Double`), които са съответно идентификатор на сметка, идентификатор на човек и баланс по сметката. Нека имаме и тип `Person`, който е представен като вектор от три елемента (`Int, String, String`), които са съответно идентификатор на човек, име и местоживеене. Дефинирайте двуаргументна функция `getAverageBalance` от тип `getAverageBalance :: ([Account], [Person]) -> (Person -> Bool) -> Double`. Функцията получава като първи аргумент база от данни, представена като двойка (`[Account], [Person]`) от списък от сметки и списък от хора. Вторият аргумент `p` е предикат. Функцията `getAverageBalance` трябва да връща средния баланс по всички сметки на хората, които удовлетворяват предиката `p`.

Примери:

```
ps :: [Person]
ps = [(1, "Ivan", "Sofia"), (2, "Georgi", "Burgas"),
      (3, "Petar", "Plovdiv"), (4, "Petya", "Burgas")]
```

```
as :: [Account]
as = [(1, 1, 12.5), (2, 1, 123.2), (3, 2, 13.0), (4, 2, 50.2),
      (5, 2, 17.2), (6, 3, 18.3), (7, 4, 19.4)]
```

```
(getAverageBalance (as,ps) (\ (_,_,city) -> city == "Burgas")) →
24.95
```

```
(getAverageBalance (as,ps) (\ (_,(n:_),_) -> n == 'P')) → 18.85
```