

КОНТРОЛНА РАБОТА № 2 ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ

КН, 2-ри курс, 1-ви поток (16.01.2022 г.)

Задача 1

Да се дефинира функция `squareDigits :: Int -> Int`, която приема цяло число `n` и връща числото, което би се получило, ако се конкатенират квадратите на поредните цифри от `n`.

Примери:

```
squareDigits 9119    → 811181
squareDigits (-9119) → -811181
```

Задача 2

Продавач на книги има много книги, класифицирани в 26 категории с етикети `A`, `B`, ... `Z`. Всяка книга има код `c` от 3, или повече знака. Първият знак на кода е главна буква, която определя категорията на книгата.

В списъка на продавача на книги всеки код е последван от интервал и цяло положително число, което показва количеството книги с този код на склад.

Например, ако дефинираме алгебричен тип, представящ книгите, като:

```
data Stock = Stock String Int, то част от списъка може да изглежда така:
L = [Stock "ABART" 20, Stock "CDXEF" 50, Stock "BKWRK" 25, Stock
     "BTSQZ" 89, Stock "DRTYM" 60].
```

Да се дефинира функция `stocklist :: [Stock] -> [Char] -> [(Char, Int)]`, която за подаден списък `L` и списък с категории от главни букви, напр.

```
M = ["A", "B", "C", "W"],
```

намира всички книги в `L`, принадлежащи към всяка отделна категория от `M`, и тяхното сумарно количество за всяка категория. Резултатът за горния пример трябва да бъде `[('A',20), ('B',114), ('C',50), ('W',0)]`, където `A`, `B`, `C` и `W` са категориите, `20` е броят книги от категория `A` (в този случай само "ABART"), `114` съответства на "BKWRK" и "BTSQZ", `50` - на "CDXEF" и `0` - на "W", т.к. няма книги с код, започващ с 'W'.

Ако `L` или `M` са празни, да се връща празен списък.

Примери:

```
stocks = [Stock "ABAR" 200, Stock "CDXE" 500, Stock "BKWR"
          250, Stock "BTSQ" 890, Stock "DRTY" 600]
stocklist stocks ['A','B'] → [('A',200), ('B',1140)]
stocklist stocks ['C','X'] → [('C',500), ('X',0)]
stocklist stocks ['Y','X'] → [('Y',0), ('X',0)]
stocklist stocks ['C']    → [('C', 500)]
```

Задача 3

Да се дефинира функция `matching :: String -> [(Int, Int)]`, която връща списък с двueleментни вектори, представляващи индексите на отварящите и затварящите квадратни скоби в даден низ. Редът на векторите не е от значение, т.е. за верен отговор във втория пример ще бъде зачетен както списъкът `[(3, 5), (1, 7)]`, така и списъкът `[(1, 7), (3, 5)]`.

Примери:

```
matching "1234"      → []  
matching ",[.-],]"   → [(3,5), (1,7)]  
matching ",+[-.,+]"  → [(2,7)]  
matching "[][]"      → [(0,1), (2,3)]
```

Задача 4

Нека е дефиниран следният полиморфен алгебричен тип, представящ двоично дърво: `data BTree a = Nil | Node a (BTree a) (BTree a)`.

Да се дефинира функция `isPerfectlyBalanced :: BTree a -> Bool`, която проверява дали дадено двоично дърво е идеално балансирано, т.е. броят на неговите възли е равен на $2^n - 1$, където n е височината на дървото.

Пример:

```
isPerfectlyBalanced t1 == True
```

