

**КОНТРОЛНА РАБОТА № 1 ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ**  
**Специалност „Информационни системи“, 1-ви курс**  
**07.04.2019 г.**

**Задача 1.** Да се дефинира функция **checkNumber :: Integer -> (Integer, Integer)**, която намира сумата от цифрите с четни индекси и сумата от цифрите с нечетни индекси в дадено естествено число и връща двете суми като двойка, първият елемент на която съответства на сумата от цифрите с четни индекси. Приемаме, че цифрите на числото са индексирани от 0 и отляво надясно.

*Примери:*

checkNumber 2728 → (4,15)

checkNumber 31415 → (12,2)

checkNumber 121 → (2,2)

**Задача 2.** Според основната теорема на аритметиката, всяко естествено число, което е по-голямо от 2, може да се представи като произведение на прости числа. Да се дефинира функция **factorize :: Int -> [Int]**, която приема естествено число, по-голямо от 1, и връща сортиран списък от елементите на това представяне.

*Примери:*

factorize 13 → [13] -- числото е просто и съответният списък съдържа един елемент

factorize 152 → [2, 2, 2, 19] --  $2 * 2 * 2 * 19 = 152$

factorize 123 → [3, 41] --  $3 * 41 = 123$

**Задача 3.** Да се дефинира функция **myFilter :: Double -> [[Double]] -> [[Double]]**, която за дадени реално число **x** и списък **xss**, елементите на който са непразни списъци от числа, връща като резултат списък от тези елементи на **xss**, средното аритметично на чиито елементи е по-малко от **x**.

*Пример:*

myFilter 10 [[1,2,3],[100,200],[10]] → [[1.0,2.0,3.0]]

**Задача 4.** Run-length encoding е прост начин за компресия на текст, при който последователните срещания на един елемент (символ от текста) се заменят с *<брой на срещания><елемент>*, а в случай, че в резултат на тази замяна биха се получили поне толкова символи, колкото се съдържат в оригиналния текст, се запазва оригиналният текст. Например, ако имаме само едно срещане на буквата "a" и го заменим с "1a", то промененият текст ще има повече символи от оригиналния.

Да се дефинира функция **encode :: String -> String**, която компресира низ по описания метод. Да се дефинира и обратната функция **decode :: String -> String**.

*Примери:*

encode "Haskell" → "Haskell"

encode "aaabccdefff" → "3abccde3f"

encode "aaaaaaaaaaaabbbb" → "12a3b"

decode "12a3b" → "aaaaaaaaaaaabbbb"