Изпит по Функционално програмиране
Специалност „Информационни системи“, 27.06.2019 г.

Задача 1. Да се дефинира функция **biggestNumber**, която за даден непразен списък от положителни едноцифрени цели числа намира най-голямото число, което може да се образува от цифрите на всички числа в списъка.
Примери:

biggestNumber [1,2,3,4,5] → 54321

biggestNumber [1,5,5,3,5] → 55531

Задача 2. Да се дефинира функция **intersectionPoints**, която получава две едноаргументни целочислени функции и краищата на целочислен интервал. Функцията да връща списък от точките от дадения интервал, в които двете функции се пресичат (т.е имат една и съща стойност), или празния списък, ако функциите не се пресичат в този интервал.
Примери:

intersectionPoints (\x **->** x) (\x **->** x * x) (-5) 5 → [0,1]

intersectionPoints (\x **->** x) (\x **->** x * x + 1) (-5) 5 → [ ]

Задача 3. Да се дефинира функция **iterator l f**, която проверява дали всеки елемент (без първия) на числовия списък **l** се получава от предишния чрез прилагане на едноаргументната числова функция **f.**
Примери:

iterator [3, 4, 5] (+1) → True

iterator [1, 2, 4] (+1) → False

Задача 4. Казваме, че едно двоично дърво е конус, ако сумата на върховете на всяко ниво на дървото е по-голяма от сумата на върховете на предишното ниво.
Като се използва следното представяне на двоично дърво:
data BTree = Empty | BTree Int BTree BTree
a) Да се дефинира функция **levelsum t k**, която намира сумата на върховете на ниво **k** в двоичното дърво **t**.
b) Да се дефинира функция **cone t**, която проверява дали двоичното дърво **t** е конус.