

КОНТРОЛНА РАБОТА № 1 ПО ФУНКЦИОНАЛНО ПРОГРАМИРАНЕ
Специалност „Информационни системи“, 1-ви курс
01.06.2020 г.

Задача 1. Да се дефинира предикат `checkSequence :: [Int] -> Bool`, който приема списък от цели числа `[a1, a2, ..., an]` и проверява дали следните две условия са изпълнени:

1. за всяко $i < j$: $a_i < a_j$,
2. за всяко $i < j$: $\text{div } a_j a_i \neq 0$.

Примери:

```
checkSequence [2, 9, 15]           → True
checkSequence [11, 14, 20, 27, 31] → True
checkSequence [11, 14, 28, 27, 31] → False
checkSequence [11, 14, 14, 29, 31] → False
```

Задача 2. Да се дефинира функция `removeNb :: Int -> [(Int, Int)]`, която приема естествено число `n` и връща списък от двойки естествени числа `(a, b)` – такива, че:

1. `a` и `b` са по-малки от `n`,
2. тяхното произведение е равно на сумата от числата от 1 до `n` без `a` и `b`.

Примери:

```
removeNb 26 → [(15,21), (21,15)]
removeNb 100 → []
removeNb 101 → [(55,91), (91,55)]
```

Задача 3. Ако `A(x1, y1)` и `B(x2, y2)` са две точки в декартовата равнина – такива, че $x_1 \neq x_2$, то уравнението на правата `AB`, която минава през тези две точки, е $y = f(x)$, където $f(x) = y_1 + (x - x_1) * (y_2 - y_1) / (x_2 - x_1)$.

а) Нека имаме тип „точка“, представен така: `type Point = (Double, Double)`. Да се дефинира функция `line :: Point -> Point -> (Double -> Double)`, която по две точки връща функцията, определяща уравнението на минаващата през тях права.
б) Да се дефинира функция от по-висок ред `liesOn :: (Double -> Double) -> (Point -> Bool)`, която за дадена функция `f`, определяща уравнението на права, връща като резултат функция, която по дадена точка `P(x, y)` проверява дали точката `P` лежи на правата `f` (дали $y = f(x)$).

Пример:

```
Ako diagonal = line (0,0) (1,1), a onDiag = liesOn diagonal, то
onDiag (5.5, 5.5) → True
onDiag (0.5, 0)   → False
```