

# Контролна работа 3

## Вариант за групата от 08:00

*Важно: Решете задачите на Haskell. За всяка от функциите напишете подходящи unit test-ове. За тестовете трябва да се използва HUnit.*

### Задача 1

Двоично дърво в Haskell ще представяме чрез следния тип, който ви е познат от домашните:

```
data Tree a = EmptyTree | Node {  
    value :: a,  
    left  :: Tree a,  
    right :: Tree a  
} deriving (Show, Read)
```

Например дървото

```
      a  
     /\   
    b  c  
   /\  \   
  d  e  f  
   \   
    g
```

можем да запишем в променлива по следния начин:

```
sampleTree = Node 'a' (Node 'b' (Node 'd' EmptyTree  
                                (Node 'g' EmptyTree EmptyTree))  
                        (Node 'e' EmptyTree EmptyTree))  
                (Node 'c' EmptyTree  
                (Node 'f' EmptyTree EmptyTree))
```

Нека е дадено двоично дърво от символи от тип **Char**. Напишете функция, която връща списък от всички думи, които могат да се образуват по път от корена, до някое от листата на дървото. Ако дървото е празно, да се връща празният списък.

Например за дървото от примера списъкът от възможни думи е:  
["abdg", "abe", "acf"]

Функцията трябва да има следния тип:  
treeWords :: Tree Char -> [String]

## Задача 2

В решението на тази задача не може да се използва рекурсия. Рекурсивни решения ще бъдат оценявани с нула точки.

**Упътване:** в задачата използвайте *list comprehension* и някоя/и от дадените по-долу библиотечни функции на Haskell. За някои от тях (например `nub` е такава) може да се наложи да импортирате модула `Data.List`.

- `minimum` (намира минимален елемент на списък)
- `maximum` (намира максимален елемент на списък)
- `nub` (премахва повторения от списък)
- `intersect` (намира сечение на два списъка)
- `\\` (оператор, който намира разлика на два списъка)
- и т.н.

Реализирайте функцията

```
mapsto :: Integral t => (t -> t) -> t -> t -> (t,t)
mapsto f a b
```

Тя получава границите на целочислен интервал  $[a,b]$  и функция от вида

```
f :: Integral t => t -> t
```

`mapsto` трябва да върне най-малкия целочислен интервал, който обхваща образа на  $[a,b]$  проектиран от `f`. Казано по-точно, трябва да се върне най-малкият интервал  $[c,d]$ , такъв, че:

$$\forall (x \in [a,b])(f(x) \in [c,d])$$

Интервалът да се върне като наредена двойка от неговите граници. Например интервалът  $[1,5]$  трябва да се върне като `(1,5)`.