

**Контролна работа № 2 по Функционално програмиране**  
**Специалност „Информационни системи“, I курс, 02.06.2024 г.**

### **Задача 1**

Да се дефинира на функционално ниво функцията `longestList :: [[a]] -> [a]`, която приема списък от списъци с различна дължина и намира най-дългия подсписък. За решаване на задачата да се използва foldl.

*Примери:*

```
longestList [[1,2], [3,4,5], [1], [100,200,300,900]] →  
[100,200,300,900]  
longestList [[5], [12,128,11,441,21,8], [], [12,64,261,7]] →  
[12,128,11,441,21,8]  
longestList [[1,2], [3], [4,5,6], [7,8,9]] → [7,8,9]
```

### **Задача 2**

Поредица от скоби се счита за валидна, ако за всяка отваряща скоба в поредицата е налична съответна затваряща скоба.

Да се дефинира функцията `validBraces :: String -> Bool`, която приема поредица от скоби и определя дали тя е валидна. Всички подавани низове ще се състоят само от валидни символи, представляващи скоби: `(, ), [, ], {, }` и `}`.

*Примери:*

```
validBraces "()" → True  
validBraces "[()]" → False  
validBraces "())({}){()}[]]" → False  
validBraces "({})[({)}]" → True  
validBraces "([{}])" → False
```

### **Задача 3**

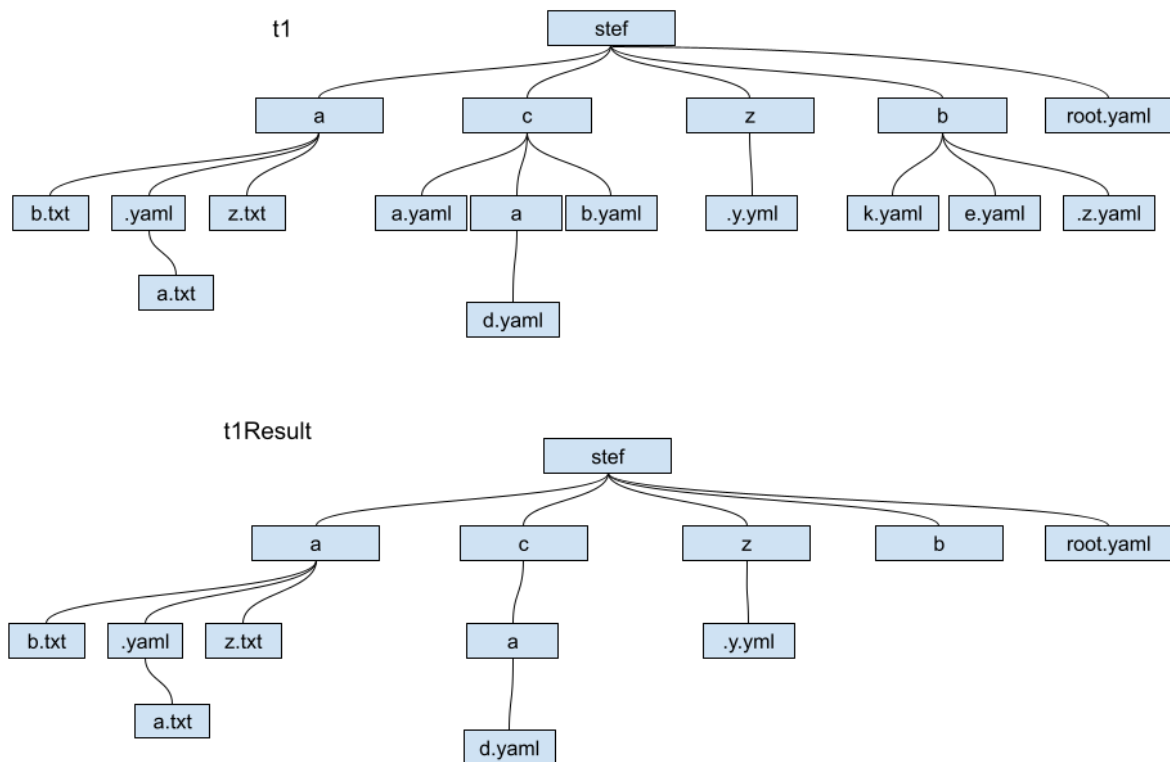
Нека е дефиниран следният алгебричен тип данни, описващ дървовидната структура на файлова система:

```
data NTree = Nil | Node String [NTree]
```

Да се дефинира функцията `storageMaintenance :: NTree -> Int -> String -> NTree`, която приема дърво от тип `NTree`, дълбочина (ниво) и разширение и премахва всички файлове в системата, които се намират на посоченото ниво и са с даденото разширение. Обектите, започващи с точка (напр. `.y.yaml`), са скрити. Функцията трябва да премахва скрити, както и нескрити файлове.

*Пример:*

```
storageMaintenance t1 2 ".yaml" → t1Result
```



#### Задача 4

Джон и Клайд играят игра. Клайд показва малка торбичка с няколко топчета, които са червени, зелени или сини. При едно теглене, Клайд взема шепа произволни топчета, показва ги на Джон и след това ги връща обратно в торбичката. В една игра се правят произволен брой тегления.

Двамата играят няколко игри и записват информация за всяка от тях. Всяка игра е посочена с нейния номер, последван от списък с цвета и количеството на изтеглените топчета. *Степен на набор от топчета* е произведението на броя на червените, зелените и сините топчета, изтеглени в една игра.

Дадени са следните типове, представящи теглене с връщане от торба с топчета:

```
type Number = Int
type Amount = Integer
data Sample = Blue Amount | Red Amount | Green Amount
data Game = Game Number [[Sample]]
```

Да се дефинира функция `minRequiredSum :: [Game] -> Amount`, която приема списък от игри и връща сумата на степените на наборите от топчета, състоящи се от минималния брой червени, зелени и сини топчета, необходими за провеждане на всяка от отделните игри.

*Примери:*

```
minRequiredSum [Game 1 [[Blue 3,Red 4],[Red 1,Green 2,Blue 6],[Green 2]],Game 2 [[Blue 1,Green 2],[Green 3,Blue 4,Red 1],[Green 1,Blue 1]],Game 3 [[Green 8,Blue 6,Red 20],[Blue 5,Red 4,Green 13],[Green 5,Red 1]],Game 4 [[Green 1,Red 3,Blue 6],[Green 3,Red 6],[Green 3,Blue 15,Red 14]],Game 5 [[Red 6,Blue 1,Green 3],[Blue 2,Red 1,Green 2]]] -> 2286
```

minRequiredSum [Game 1 [[Red 18,Green 8,Blue 7],[Red 15,Blue 4,Green 1],[Green 2,Red 17,Blue 6],[Green 5,Blue 1,Red 11],[Red 18,Green 1,Blue 14],[Blue 8]],Game 2 [[Blue 16,Green 12,Red 3],[Blue 13,Red 2,Green 8],[Green 15,Red 3,Blue 16]],Game 3 [[Green 6,Red 15],[Green 1,Red 4,Blue 7],[Blue 9,Red 7,Green 8]],Game 4 [[Red 8,Blue 2],[Red 11,Blue 5,Green 1],[Red 12,Green 1,Blue 5],[Blue 1],[Blue 2,Red 9]],Game 5 [[Blue 9,Red 3,Green 12],[Green 3,Red 4,Blue 17],[Blue 15,Green 2,Red 5],[Blue 3,Green 5,Red 6],[Red 6,Blue 4,Green 7],[Green 3,Blue 10]],Game 6 [[Red 11,Blue 2,Green 6],[Blue 2,Red 9,Green 4],[Blue 3,Red 12,Green 8],[Red 5,Green 11,Blue 4],[Blue 2,Red 9,Green 13],[Red 15,Blue 3,Green 7]],Game 7 [[Red 2,Green 9,Blue 12],[Blue 14,Green 1,Red 6],[Blue 7,Green 9],[Green 9,Red 8,Blue 4],[Red 5,Green 3,Blue 16],[Red 4,Green 8]],Game 8 [[Red 11,Green 12,Blue 1],[Red 4,Green 7],[Red 11,Green 6],[Green 17],[Green 15,Red 1]],Game 9 [[Red 1,Green 1,Blue 12],[Green 3,Red 12,Blue 6],[Red 14,Blue 1],[Blue 9,Red 1,Green 3]],Game 10 [[Red 1,Blue 4],[Blue 3,Green 4],[Green 3,Red 3,Blue 8],[Blue 2,Red 3],[Green 3,Red 4,Blue 3]],Game 11 [[Blue 8,Red 1],[Green 8,Red 1,Blue 1],[Green 13,Red 9,Blue 6]],Game 12 [[Red 2,Blue 2,Green 1],[Red 3,Green 1],[Blue 1,Green 3]],Game 13 [[Green 12,Blue 4],[Red 2,Blue 2,Green 8],[Green 6,Red 3],[Red 3,Green 5],[Green 9,Blue 7,Red 1]],Game 14 [[Red 1,Green 7],[Green 5,Red 12,Blue 10],[Red 9,Blue 11,Green 7],[Blue 7,Red 3,Green 9]],Game 15 [[Green 7,Blue 1],[Red 1,Green 2,Blue 1],[Green 7]],Game 16 [[Green 1,Blue 1],[Blue 2,Green 4,Red 2],[Green 2,Blue 2]],Game 17 [[Red 6,Green 11,Blue 7],[Blue 1,Green 13,Red 4],[Green 4,Blue 6,Red 7]],Game 18 [[Red 2,Blue 8],[Red 7,Blue 11],[Green 1,Blue 16,Red 7],[Blue 18,Green 1,Red 14]],Game 19 [[Red 2,Blue 2],[Green 1,Red 6],[Green 1,Red 3,Blue 2]],Game 20 [[Red 6,Blue 2,Green 5],[Red 4,Blue 1,Green 9],[Blue 3,Red 2,Green 9],[Red 8,Green 12,Blue 5]],Game 21 [[Red 6,Blue 7],[Blue 3,Red 16,Green 2],[Blue 2,Red 13],[Blue 3,Red 11,Green 3],[Green 1,Red 18,Blue 6],[Red 12,Blue 5,Green 2]],Game 22 [[Red 9,Blue 6,Green 14],[Blue 1,Green 5,Red 13],[Red 6],[Red 18,Green 4],[Blue 2,Green 10,Red 16],[Red 1,Green 18,Blue 1]]] → 13971