

## Домашна работа № 2 по Функционално програмиране

специалност „Информационни системи“, I курс, 2022/2023 учебна година

---

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран и да съдържа коментари на ключовите места. Предайте решенията на всички задачи в *един* файл с наименование *hw2\_<FN>.hs*, където *<FN>* е Вашият факултетен номер.

Домашните работи се предават като изпълнение на съответното задание в курса по ФП в Moodle (<https://learn.fmi.uni-sofia.bg/course/view.php?id=9029>) най-късно **до 23:55 ч. на 14.05.2023 г. (неделя)**.

*Приятна работа и успех!*

---

Нека е дефинирана следната структура на филмова база от данни:

```
type Title = String
type Year = Int
type Length = Int
type InColor = Char
type StudioName = String

type Name = String
type ProducerID = Int
type Networth = Integer

type Movie = (Title, Year, Length, InColor, StudioName)
type Studio = (Name, ProducerID)
type MovieExec = (Name, ProducerID, Networth)

type MovieDB = ([Movie], [Studio], [MovieExec])
```

Типът **Movie** представя данните за даден филм - неговото име, годината на излизането му на голям екран, продължителността му в минути, типът му и името на студиото, което го е заснело. Възможно е продължителността на филм да е неизвестна - тогава тя се кодира с произволно отрицателно число. Типът на даден филм е или черно-бял (и тогава стойността е `'N'`), или цветен (и тогава стойността е `'Y'`).

Типът **Studio** представя данните за дадено студио - неговото име и номер, който служи за идентифициране на неговия президент (който е и продуцент).

Типът **MovieExec** представя данните за даден продуцент - неговите име, номер и нетна стойност на активите.

Стойностите от тип **MovieDB** са вектори с три елемента - списъци, които представят базата от данни. Списъците съдържат съответно всички налични данни за филми, студиа и продуценти.

В задачите по-долу е използвана следната филмова база от данни:

**studios :: [Studio]**

```
studios = [("Disney", 199),("USA Entertainm.", 222),("Fox", 333),("Paramount", 123),("MGM", 555),("Buzzfeed Entertainm.", 42)]
```

**movieExecs :: [MovieExec]**

```
movieExecs = [("George Lucas", 555, 200000000),("Ted Turner", 333, 125000000),("Stephen Spielberg", 222, 100000000),("Merv Griffin", 199, 112000000),("Calvin Coolidge", 123, 20000000),("Christian Baesler", 42, 420000)]
```

**movies :: [Movie]**

```
movies = [("Pretty Woman", 1990, 119, 'Y', "Disney"),("The Man Who Wasn't There", 2001, 116, 'Y', "USA Entertainm."),("Logan's run", 1976, 120, 'Y', "Fox"),("Star Wars", 1977, -1, 'N', "Fox"),("Star Wars 2", 1977, 238, 'N', "Fox"),("Empire Strikes Back", 1980, 111, 'Y', "Fox"),("Star Trek", 1979, 132, 'Y', "Paramount"),("Star Trek: Nemesis", 2002, 116, 'Y', "Paramount"),("Terms of Endearment", 1983, 132, 'N', "MGM"),("The Usual Suspects", 1995, 106, 'Y', "MGM"),("Gone With the Wind", 1938, 238, 'Y', "MGM"),("Gone With the Wind 2", 1938, 238, 'Y', "MGM"),("The Fellowship of the Ring", 2001, -1, 'Y', "USA Entertainm.")]
```

**db :: MovieDB**

```
db = (movies, studios, movieExecs)
```

## Задача 1

Да се дефинира функция от по-висок ред **getNameLengthColor :: MovieDB -> ((Char -> Bool), (Int -> Bool)) -> [(Title, Length)]**.

Функцията приема филмова база от данни и вектор с два предиката: първият се отнася за типа на филм, а вторият - за неговата дължина. Върнатата от функцията стойност е списък от заглавията и дължините в минути на всички филми, които удовлетворяват двата предиката и имат дължина, която е известна и различна от тази на най-дългия цветен филм. Ако има няколко филма с максимална дължина, нито един от тях не трябва да бъде част от резултата. **Намирането на най-дългия цветен филм да се извърши чрез техниката за комбиниране / акумулиране на елементите на списък (folding)!**

*Примери:*

```
getNameLengthColor db ((== 'Y'), (> 106)) -> [("Pretty Woman",119),("The Man Who Wasn't There",116),("Logan's run",120),("Empire Strikes Back",111),("Star Trek",132),("Star Trek: Nemesis",116)]
```

```
getNameLengthColor db ((== 'Y'), (> 237)) -> []
```

```
getNameLengthColor db ((== 'Y'), (> 238)) -> []
```

```
getNameLengthColor db ((== 'N'), (< 133)) -> [("Terms of Endearment",132)]
```

```
getNameLengthColor db ((== 'N'), (< 300)) → [("Terms of  
Endearment",132)]
```

## Задача 2

Да се дефинира функция от по-висок ред

**getStudios :: MovieDB -> ([Year] -> [(StudioName, Name)]).**

Функцията приема филмова база от данни и връща нова функция, която приема списък от години и връща имената и президентите на тези студиа, които нямат филми или са снимали филми само в една единствена година и тя е част от приетия списък.

*Примери:*

```
(getStudios db) [2001] → [("USA Entertainm.", "Stephen  
Spielberg"), ("Buzzfeed Entertainm.", "Christian Baesler")]  
(getStudios db) [2002] → [("Buzzfeed Entertainm.", "Christian  
Baesler")]  
(getStudios db) [1990] → [("Disney", "Merv Griffin"), ("Buzzfeed  
Entertainm.", "Christian Baesler")]  
(getStudios db) [1990, 2001, 1976] → [("Disney", "Merv  
Griffin"), ("USA Entertainm.", "Stephen Spielberg"), ("Buzzfeed  
Entertainm.", "Christian Baesler")]  
(getStudios db) [1979, 2002] → [("Buzzfeed Entertainm.", "Christian  
Baesler")]
```