

Csoportok

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

5. házi feladat

Kategória:	Házi feladatok
Elérhető:	2022. 10. 13. 23:59
Pótolható határidő:	2022. 10. 21. 23:59
Végső határidő:	2022. 10. 23. 23:59
Kiírta:	Erdei Zsófia

Leírás:

Házi feladat

A házi feladatot egy Homework5 nevű modulként kell beadni. Minden definiálandó függvényhez adjuk meg a hozzá tartozó típuszignatúrát is!

Fontos: a feltöltött tömörített zip állományban egyetlen .hs kiterjesztésű fájl szerepeljen (*hazi.txt*, *Homework.hs.txt* nem felel meg)!

Tipp: Ha szükség van rá, a feladatokat bontsd fel részfeladatokra és definiáljatok segédfüggvényeket!

A bónusz feladat megoldása nem kötelező (és azt nem teszteli az automatikus tesztelő).

Jelszó

Implementáld a `password :: [Char] -> [Char]` függvényt, ami egy karakterlánc minden karakterét a `*` karakterre cseréli ki.

Teszt:

```
password "akacfa2" == "*****"
password "hunter1234" == "*****"
password ['a'] == ['*']
password [] == []
```

Szótár

Implementáld a `lookup' :: Eq a => a -> [(a, b)] -> b` függvényt, ami kulcs-érték párok listájából kikeresi azt az értéket, ami egy adott kulcshoz tartozik. (A kulcs érték párokat egyszerű tuple-ökkel definiáljuk.)

Teszt:

```
lookup' 2 [(3,"xy"),(2,"abc"),(4,"qwe")] == "abc"
lookup' 42 [(1,2),(3,4),(42,42)] == 42
lookup' 42 [(a,a) | a <- [1..]] == 42
```

Bináris számok

Implementáld a `toBin :: Integer -> [Int]` függvényt, amely visszaadja egy nemnegatív egész szám kettes számrendszerbeli számjegyeit fordított sorrendben!

Teszt:

```
toBin 0 == []
toBin 1 == [1]
toBin 2 == [0,1]
toBin 10 == [0, 1, 0, 1]
```

Duplikátumok eltávolítása

Implementáld a `remdup :: Eq a => [a] -> [a]` függvényt, amely egy listából eltávolítja az egymás után előforduló azonos elemeket!

Teszt:

```
remdup [] == []
remdup [1,1,2,2,3,3,4,4] == [1,2,3,4]
remdup [1,1,1,1,1,2,2,2,2] == [1,2]
remdup [1,2,1,2,1] == [1,2,1,2,1]
```

Bónusz feladat

Definiáljuk a `mergeToSet :: Ord a => [a] -> [a] -> [a]` függvényt, amely összefésül két rendezett listát. Az eredmény lista az eredeti listák minden eleméből csak 1 darabot tartalmazzon!

```
mergeToSet [] [1..5] == [1,2,3,4,5]
mergeToSet [1..5] [] == [1,2,3,4,5]
mergeToSet [1,3..9] [2,4..10] == [1,2,3,4,5,6,7,8,9,10]

mergeToSet [1..10] [5..15] == [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
mergeToSet [1,1,1] [1,1,1,1,1,1,1] == [1]
mergeToSet [1,1,1,1,1] [1..5] == [1,2,3,4,5]
```

Megoldás

 Letöltés

Név:	Homework5.zip
Feltöltés ideje:	2022. 10. 22. 16:34
Értékelés:	
Státusz:	Elfogadva
Feltöltések száma:	1
Értékelte:	Erdei Zsófia
Megjegyzések:	

Automatikus tesztelés eredményei

A megoldás átment a teszteken