

3. házi feladat

Kategória:

Házi feladatok

Végső határidő:

3/7/2025, 11:59 PM (Beadva)

Próbálkozások száma:

Korlátlan

Kiírta:

Erdei Zsófia

Leírás:

Mintaillesztés

A feladatokban tessék használni mintaillesztést. Elágazást, if-then-else-et nem szabad használni, mert nem tanultuk és nincs is rájuk szükség.

0. Modul

Definiálj egy modult `Hazi3` néven!

1a. Pontosan egy elemű

Írj egy olyan függvényt `isSingleton` néven, amely megállapítja egy listáról, hogy pontosan 1 elemű-e.

```
isSingleton :: [a] -> Bool
```

1b. Kettő vagy legalább négy elemű

Írj egy olyan függvényt `exactly20rAtLeast4` néven, amely megállapítja egy listáról, hogy pontosan 2 vagy legalább 4 elemű-e.

```
exactly20rAtLeast4 :: [a] -> Bool
```

2. Első két elem

Definiáld a `firstTwoElements` függvényt, amely visszaadja egy lista első két elemét listaként, ha van legalább két elem. Ha nincs két elem, akkor adj vissza üres listát.

```
firstTwoElements :: [a] -> [a]
```

3. Harmadik nélkül

Definiáld a `withoutThird` függvényt, amely egy legalább 3 elemű listából kihagyja a harmadik elemet. Ha listában nincs 3 elem, akkor amúgy sincs benne a harmadik elem, így az eredmény csak a paraméterül kapott lista legyen.

```
withoutThird :: [a] -> [a]
```

4. Négyesből lista

Definiáld a `quadrupleToList` függvényt, amely egy rendezett négyesből listát készít!

```
quadrupleToList :: (a,a,a,a) -> [a]
```

5. Második elem beszúrása

Definiáld az `insertSecond` függvényt, amely egy elemet a lista második helyére szúr be. Ha a lista üres, adjon vissza üres listát.

```
insertSecond :: a -> [a] -> [a]
```

6. Első párból lista

Definiáld a **firstPairToList** függvényt, amely egy rendezett párokból álló lista első elemét egy listává alakítja. Ha a lista üres, adjon vissza üres listát.

```
firstPairToList :: [(a,a)] -> [a]
```

Tesztetek

```
isSingleton [1]
isSingleton "x"
not (isSingleton [])
not (isSingleton ['x','a'])
not (isSingleton [1..])
not (isSingleton [5,4,6])
not (isSingleton [6,5])
exactly20rAtLeast4 "alma"
exactly20rAtLeast4 "te"
exactly20rAtLeast4 [1,10]
exactly20rAtLeast4 [1.5,9.25]
exactly20rAtLeast4 [1..]
not (exactly20rAtLeast4 [])
not (exactly20rAtLeast4 [1])
not (exactly20rAtLeast4 [1,2,3])
not (exactly20rAtLeast4 "nem")
not (exactly20rAtLeast4 "a")
firstTwoElements [1,2] == [1,2]
firstTwoElements "alma" == "al"
firstTwoElements [20,9,8,7,6,5,4,3,2,1,0] == [20,9]
firstTwoElements [2,9,5,-3,-7,6,10,2,2,3,3] == [2,9]
firstTwoElements [10..] == [10,11]
firstTwoElements [1.5] == []
firstTwoElements [2] == []
firstTwoElements "a" == []
firstTwoElements ([] :: [Integer]) == []
withoutThird [1,2] == [1,2]
withoutThird [1,2,3] == [1,2]
withoutThird "alma" == "ala"
withoutThird [20,9,8,7,6,5,4,3,2,1,0] == [20,9,7,6,5,4,3,2,1,0]
withoutThird [2,9,5,-3,-7,6,10,2,2,3,3] == [2,9,-3,-7,6,10,2,2,3,3]
take 5 (withoutThird [10..]) == [10,11,13,14,15]
withoutThird [1.5] == [1.5]
withoutThird [2] == [2]
withoutThird "a" == "a"
withoutThird ([] :: [Int]) == []
quadrupleToList (1,2,3,4) == [1,2,3,4]
quadrupleToList ('a','b','c','d') == "abcd"
insertSecond 1 [] == []
insertSecond 1 [2] == [2,1]
insertSecond 42 [1..10] == [1,42] ++ [2..10]
insertSecond 'b' "acd" == "abcd"
firstPairToList [] == []
firstPairToList [(1,2)] == [1,2]
firstPairToList [(1,2),(3,4),(5,6)] == [1,2]
firstPairToList [('a','b'),('c','d')] == "ab"
```

Git tároló

Útvonal:

<https://tms.inf.elte.hu/git/8067/vl2l98/wce51fa9dabe7346fdb33c8f44>

Használat:

`git clone https://tms.inf.elte.hu/git/8067/vl2l98/wce51fa9dabe7346fdb33c8f44`

Megoldás



Név:

solution.zip

Feltöltés ideje:

3/3/2025, 12:32 PM

Értékelés:

Státusz:

Elfogadva

Feltöltések száma:

2

Értékelte:

Szávó Tamás

Megjegyzések:

Automatikus tesztelés eredményei

✓ #1

Mellékelt fájlok

tests3.txt

