

# Csoportok

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

## 4. ZH - A csoport

<b>Kategória:</b>	Vizsgafeladatok
<b>Elérhető:</b>	2022. 11. 25. 8:20
<b>Pótolható határidő:</b>	
<b>Végső határidő:</b>	2022. 11. 25. 8:50
<b>Kiírta:</b>	Erdei Zsófia

Leírás:

## Funkcionális programozás 4. ZH - A csoport

### Előzetes tudnivalók

Használható segédanyagok:

- [Haskell könyvtárak dokumentációja](#),
- [Hoogle](#),
- [a tárgy honlapja](#), és a
- [Haskell szintaxis összefoglaló](#).

**Más segítőeszköz nem használható.**

Ha bármilyen kérdés, észrevétel felmerül, azt a gyakorlatvezetőnek kell jelezni, **nem** a diáktársaknak!

A feladatsor megoldására 30 perc áll rendelkezésre.

A feladatok tetszőleges sorrendben megoldhatóak. A pontozás szabályai a következők:

- Minden teszten átmenő megoldás ér teljes pontszámot.
- Funkcionálisan hibás (valamelyik teszteseten megbukó) megoldás nem ér pontot.
- Fordítási hibás vagy hiányzó megoldás esetén a teljes megoldás 0 pontos.

Ha hiányos/hibás részek lennének a feltöltött megoldásban, azok kommentben szerepeljenek.

*Tekintve, hogy a tesztesetek, bár odafigyelés mellett íródnak, nem fedik le minden esetben a függvény teljes működését, határozottan javasolt még külön próbálgatni a megoldásokat beadás előtt!*

Az elméleti kérdésekre adott választ a forráskódban kell elhelyezni, kommentben. Minden függvénynek meg kell adni a típuszignatúráját is. A függvények elvárt neve és típusa zárójelben meg van adva.

`Zartheyli4` néven kell deklarálni a modult. A `.hs` fájlt `.zip`-be tömörítve kell beadni.

### Elméleti kérdés (1 pont)

Mit jelent a parciális alkalmazás? Adj rá egy példát!

### Gyakorlati feladatok

#### Mindent kisbetűre (1 pont)

Definiáld a `upperCharToLower :: String -> String` nevű függvényt, mely minden nagybetűt kisbetűvé alakít, a többi karaktert eldobja! A megoldáshoz használd a `map` és a `filter` függvényeket.

```
upperCharToLower "" == ""
upperCharToLower "CSUPA NAGYBETU" == "csupanagybetu"
upperCharToLower "csupa kisbetu!!!!" == ""
upperCharToLower "Ez Egy AlmaFa." == "eeaf"
```

#### Párok cseréje (2 pont)

Adott egy rendezett párokat tartalmazó lista. Definiáld a `swapIfCond :: (a -> Bool) -> [(a,a)] -> [(a,a)]` függvényt, amely megcseréli egy párban az elemek sorrendjét, ha a

pár első elemére teljesül a paraméterként kapott feltétel.

```
swapIfCond even [] == []
swapIfCond even [(1,2),(3,4),(5,6)] == [(1,2),(3,4),(5,6)]
swapIfCond odd [(1,2),(3,4),(5,6)] == [(2,1),(4,3),(6,5)]
swapIfCond odd [(1,2),(2,3),(3,4),(4,5),(5,6)] == [(2,1),(2,3),(4,3),(4,5),(6,5)]
```

## Indexek kiválogatása (2 pont)

Definiáld a `findIndices :: (a -> Bool) -> [a] -> [Int]` függvényt, amely egy listából kiválogatja azon elemek indexeit, amelyek megfelelnek a az első paraméterként megadott feltételnek. A megoldáshoz nem használható a `Data.List` -ben található azonos nevű függvény.

```
findIndices (<10) [] == []
findIndices (<10) [10..20] == []
findIndices (<10) [1..15] == [0..8]
findIndices even [1..10] == [1,3,5,7,9]
findIndices (== 'e') "eszperente nyelven" == [0,4,6,9,13,16]
findIndices null [[],[1..],[42],[],[],[1,2,3]] == [0,3,4]
```