

# Csoportok

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

## Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

### 4. ZH - B csoport

|                            |                    |
|----------------------------|--------------------|
| <b>Kategória:</b>          | Vizsgafeladatok    |
| <b>Elérhető:</b>           | 2022. 11. 25. 8:20 |
| <b>Pótolható határidő:</b> |                    |
| <b>Végső határidő:</b>     | 2022. 11. 25. 8:52 |
| <b>Kiírta:</b>             | Erdei Zsófia       |

Leírás:

## Funkcionális programozás 4. ZH - B 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.

`Zarthelyi4` néven kell deklarálni a modult. A `.hs` fájlt `.zip`-be tömörítve kell beadni.

### Elméleti kérdés (1 pont)

Mit nevezünk magasabbrendű függvénynek? Adj egy példát, amely nem szerepel a feladatsorban!

### Gyakorlati feladatok

#### Mindent kisbetűre (1 pont)

Definiáld az `toAllCaps :: String -> String` nevű függvényt, mely minden kisbetűt nagybetűvé alakít, a többi karaktert eldobja! A megoldáshoz használd a `map` és a `filter` függvényeket.

```
toAllCaps "" == ""
toAllCaps "CSUPA NAGYBETU" == ""
toAllCaps "csupa kisbetu!!!!" == "CSUPAKISBETU"
toAllCaps "Ez Egy AlmaFa." == "ZGYLMAA"
```

#### Szétválogatás (2 pont)

Definiáld a `partition :: (a -> Bool) -> [a] -> ([a], [a])` függvényt, amely

szétválogat egy listát egy megadott feltétel szerint és az eredményt egy rendezett pár formájában adja vissza, ahol az első listában a feltételnek megfelelő, a másodikban a nem megfelelő elemek szerepelnek. A megoldáshoz nem használható a `Data.List`-ben található azonos nevű függvény.

```
partition even [] == ([],[])
partition (<=5) [1..10] == ([1,2,3,4,5],[6,7,8,9,10])
partition (<=10) [1..10] == ([1,2,3,4,5,6,7,8,9,10],[])
partition (>10) [1..10] == ([],[1,2,3,4,5,6,7,8,9,10])
partition even [1..10] == ([2,4,6,8,10],[1,3,5,7,9])
```

Kiválasztott indexű elemek leképezése (2 pont)

Definiáld a `mapAtIndices :: (a -> a) -> [Int] -> [a] -> [a]` függvényt, amely alkalmazza az első paraméterként megadott függvényt a listában megadott indexű elemekre.

```
mapAtIndices (*10) [1..10] [] == []
mapAtIndices (*10) [] [1..10] == [1,2,3,4,5,6,7,8,9,10]
mapAtIndices (*10) [0..9] [1..10] == [10,20,30,40,50,60,70,80,90,100]
mapAtIndices (*10) [0,2,5] [1..10] == [10,2,30,4,5,60,7,8,9,10]
mapAtIndices toUpper [0,2..10] "hello world" == "HeLLo WoRLD"
```

Megoldás

Letöltés

|                    |                    |
|--------------------|--------------------|
| Név:               | Zh4.zip            |
| Feltöltés ideje:   | 2022. 11. 25. 8:48 |
| Értékelés:         |                    |
| Státusz:           | Elfogadva          |
| Feltöltések száma: | 1                  |
| Értékelte:         | Erdei Zsófia       |
| Megjegyzések:      | 1 - 0              |

Automatikus tesztelés eredményei

Valamelyik tesztesetre hibásan futott le a beadott program.

Megbukott tesztek:

```
## 1.
## Kivétel:
Eval.hs:61:45: error:
    Variable not in scope: upperCharToLower :: [Char] -> [Char]
(deferred type error)

## Teszteset:
upperCharToLower "" == ""

## 2.
## Kivétel:
Eval.hs:61:71: error:
    Variable not in scope: upperCharToLower :: [Char] -> [Char]
(deferred type error)

## Teszteset:
upperCharToLower "CSUPA NAGYBETU" == "csupanagybetu"

## 3.
## Kivétel:
Eval.hs:61:124: error:
    Variable not in scope: upperCharToLower :: [Char] -> [Char]
(deferred type error)

## Teszteset:
upperCharToLower "csupa kisbetu!!!!" == ""

## 4.
## Kivétel:
Eval.hs:61:167: error:
    Variable not in scope: upperCharToLower :: [Char] -> [Char]
(deferred type error)

## Teszteset:
upperCharToLower "Ez Egy AlmaFa." == "eeaf"

## 5.
## Kivétel:
Eval.hs:61:211: error:
    Variable not in scope:
        swapIfCond :: (Integer -> Bool) -> [a0] -> [()]
(deferred type error)

## Teszteset:
swapIfCond even [] == []

## 6.
## Kivétel:
Eval.hs:61:236: error:
    Variable not in scope:
        swapIfCond
        :: (Integer -> Bool)
        -> [(Integer, Integer)] -> [(Integer, Integer)]
(deferred type error)

## Teszteset:
swapIfCond even [(1, 2), (3, 4), (5, 6)] ==
    [(1, 2), (3, 4), (5, 6)]

## 7.
## Kivétel:
Eval.hs:61:295: error:
    Variable not in scope:
        swapIfCond
        :: (Integer -> Bool)
        -> [(Integer, Integer)] -> [(Integer, Integer)]
(deferred type error)

## Teszteset:
swapIfCond odd [(1, 2), (3, 4), (5, 6)] == [(2, 1), (4, 3), (6, 5)]
```

```
## 8.
## Kivétel:
Eval.hs:61:353: error:
  Variable not in scope:
    swapIfCond
    :: (Integer -> Bool)
      -> [(Integer, Integer)] -> [(Integer, Integer)]
(deferred type error)

## Teszteset:
swapIfCond odd [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)] ==
  [(2, 1), (2, 3), (4, 3), (4, 5), (6, 5)]

## 9.
## Kivétel:
Eval.hs:61:435: error:
  • Variable not in scope:
      findIndices :: (Integer -> Bool) -> [a1] -> [()]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)

## Teszteset:
findIndices (< 10) [] == []

## 10.
## Kivétel:
Eval.hs:61:462: error:
  • Variable not in scope:
      findIndices :: (Integer -> Bool) -> [Integer] -> [()]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)

## Teszteset:
findIndices (< 10) [10 .. 20] == []

## 11.
## Kivétel:
Eval.hs:61:495: error:
  • Variable not in scope:
      findIndices :: (Integer -> Bool) -> [Integer] -> [Integer]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)

## Teszteset:
findIndices (< 10) [1 .. 15] == [0 .. 8]

## 12.
## Kivétel:
Eval.hs:61:531: error:
  • Variable not in scope:
      findIndices :: (Integer -> Bool) -> [Integer] -> [Integer]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)

## Teszteset:
findIndices even [1 .. 10] == [1, 3, 5, 7, 9]

## 13.
## Kivétel:
Eval.hs:61:571: error:
  • Variable not in scope:
      findIndices :: (Char -> Bool) -> [Char] -> [Integer]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)
```

```
## Teszteset:
findIndices (== 'e') "eszperente nyelven" == [0, 4, 6, 9, 13, 16]

## 14.
## Kivétel:
Eval.hs:61:632: error:
  • Variable not in scope:
      findIndices :: ([a2] -> Bool) -> [[Integer]] -> [Integer]
  • Perhaps you meant one of these:
      ‘List.findIndices’ (imported from Data.List),
      ‘List.findIndex’ (imported from Data.List)
(deferred type error)

## Teszteset:
findIndices null [[], [1 ..], [42], [], [], [1, 2, 3]] == [0, 3, 4]

## 15.
## Kivétel:
Eval.hs:61:690: error:
  Variable not in scope: toAllCaps :: [Char] -> [Char]
(deferred type error)

## Teszteset:
toAllCaps "" == ""

## 16.
## Kivétel:
Eval.hs:61:709: error:
  Variable not in scope: toAllCaps :: [Char] -> [Char]
(deferred type error)

## Teszteset:
toAllCaps "CSUPA NAGYBETU" == ""

## 17.
## Kivétel:
Eval.hs:61:742: error:
  Variable not in scope: toAllCaps :: [Char] -> [Char]
(deferred type error)

## Teszteset:
toAllCaps "csupa kisbetu!!!!" == "CSUPAKISBETU"

## 18.
## Kivétel:
Eval.hs:61:790: error:
  Variable not in scope: toAllCaps :: [Char] -> [Char]
(deferred type error)

## Teszteset:
toAllCaps "Ez Egy AlmaFa." == "ZGYLMAA"

## 19.
## Kivétel:
Eval.hs:61:830: error:
  • Variable not in scope:
      partition :: (Integer -> Bool) -> [a3] -> ([()], [()])
  • Perhaps you meant ‘List.partition’ (imported from Data.List)
(deferred type error)

## Teszteset:
partition even [] == ([], [])

## 20.
## Kivétel:
Eval.hs:61:859: error:
  • Variable not in scope:
      partition
      :: (Integer -> Bool) -> [Integer] -> ([Integer], [Integer])
  • Perhaps you meant ‘List.partition’ (imported from Data.List)
(deferred type error)

## Teszteset:
```

```
partition (<= 5) [1 .. 10] == ([1, 2, 3, 4, 5], [6, 7, 8, 9, 10])

## 21.
## Kivétel:
Eval.hs:61:913: error:
    • Variable not in scope:
      partition :: (Integer -> Bool) -> [Integer] -> ([Integer], [()])
    • Perhaps you meant ‘List.partition’ (imported from Data.List)
(deferred type error)

## Teszteset:
partition (<= 10) [1 .. 10] ==
  ([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], [])

## 22.
## Kivétel:
Eval.hs:61:969: error:
    • Variable not in scope:
      partition :: (Integer -> Bool) -> [Integer] -> ([()], [Integer])
    • Perhaps you meant ‘List.partition’ (imported from Data.List)
(deferred type error)

## Teszteset:
partition (> 10) [1 .. 10] == ([], [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

## 23.
## Kivétel:
Eval.hs:61:1024: error:
    • Variable not in scope:
      partition
      :: (Integer -> Bool) -> [Integer] -> ([Integer], [Integer])
    • Perhaps you meant ‘List.partition’ (imported from Data.List)
(deferred type error)

## Teszteset:
partition even [1 .. 10] == ([2, 4, 6, 8, 10], [1, 3, 5, 7, 9])

## 24.
## Kivétel:
Eval.hs:61:1077: error:
    Variable not in scope:
      mapAtIndices :: (Integer -> Integer) -> [Integer] -> [a4] -> [()]
(deferred type error)

## Teszteset:
mapAtIndices (* 10) [1 .. 10] [] == []

## 25.
## Kivétel:
Eval.hs:61:1113: error:
    Variable not in scope:
      mapAtIndices
      :: (Integer -> Integer) -> [a5] -> [Integer] -> [Integer]
(deferred type error)

## Teszteset:
mapAtIndices (* 10) [] [1 .. 10] == [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

## 26.
## Kivétel:
Eval.hs:61:1169: error:
    Variable not in scope:
      mapAtIndices
      :: (Integer -> Integer) -> [Integer] -> [Integer] -> [Integer]
(deferred type error)

## Teszteset:
mapAtIndices (* 10) [0 .. 9] [1 .. 10] ==
  [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

## 27.
## Kivétel:
Eval.hs:61:1239: error:
    Variable not in scope:
```

```
mapAtIndices
  :: (Integer -> Integer) -> [Integer] -> [Integer] -> [Integer]
(deferred type error)

## Teszteset:
mapAtIndices (* 10) [0, 2, 5] [1 .. 10] ==
  [10, 2, 30, 4, 5, 60, 7, 8, 9, 10]

## 28.
## Kivétel:
Eval.hs:61:1303: error:
  Variable not in scope:
    mapAtIndices :: (Char -> Char) -> [Integer] -> [Char] -> [Char]
(deferred type error)

## Teszteset:
mapAtIndices Char.toUpper [0, 2 .. 10] "hello world" ==
  "HeLlO WoRLD"
```