

Csoportok

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 12

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 91

Eseményvezérelt alkalmazások

IP-18bEVALKEG | 92

1. ZH - P 8:00-10:00 - A csoport

Kategória:	Vizsgafeladatok
Elérhető:	2022. 10. 07. 8:20
Pótolható határidő:	
Végső határidő:	2022. 10. 07. 8:45
Kiírta:	Erdei Zsófia

Leírás:

1. ZH - péntek 8:00-10:00 időszáv - 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 20 perc áll rendelkezésre (+ 2 perc feltöltésre)

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 meg van adva. **Zartheyli1** néven kell deklarálni a modult. A **.hs** fájlt **.zip**-be tömörítve kell beadni.

Elméleti kérdések (1 pont / kérdés)

1. Miért nem fordul az alábbi kód?

```
f :: a -> a
f x = 2 * x
```

2. Adj példát egy függvény infix és prefix használatára!

Gyakorlati feladatok

Vektorok (1 pont)

Reprezentáljuk a 3-dimenziós térben vektorokat rendezett párokkal! A tuple első komponense legyen az egyik irányban a hossz, a második a második irányban a hossz, a harmadik a harmadik irányban a hossz. (A hosszok jelen esetben egész számok.) Készíts függvényt, amely kap két darab vektort, és megadja a két vektor skaláris szorzatát! Két vektor skaláris szorzatát úgy kapjuk meg, hogy koordinátáinként a hosszukat összeszorozzuk, és a szorzatok összegét vesszük.

```
scalar :: (Int, Int, Int) -> (Int, Int, Int) -> Int
scalar (57, -7, 74) (-25,52, 45) == 1541
scalar (-68, 41, -99) (-90,18, 16) == 5274
scalar (76, -69, -78) (-64,-96, 81) == -4558
scalar (-24, 51, 24) (82,57, 83) == 2931
scalar (-7, -15, 34) (-67,44, 73) == 2291
scalar (91, -14, 83) (7,-48, -16) == -19
scalar (96, 31, -86) (28,-29, -3) == 2047
```

Szép számok (2 pont)

Adjuk meg azon pozitív x számokat csökkenő sorrendben, amelyek 1000-nél nem nagyobbak, ötös maradékuk 3, háromszorosuk hetes maradéka 2!

```
numbers :: [Int]
numbers == [983,948,913,878,843,808,773,738,703,668,633,598,563,528,493,458,423,388]
```

Logikai függvény - A (1 pont)

Adjuk meg azt a függvényt, amely az alábbi logikai táblázat alapján 3 logikai értékből megad egy logikai értéket! Használjunk mintaillesztést!

A	B	C	logicalFunctionA(A, B, C)
Igaz	Igaz	Igaz	Igaz
Igaz	Igaz	Hamis	Hamis
Igaz	Hamis	Igaz	Igaz
Igaz	Hamis	Hamis	Igaz
Hamis	Igaz	Igaz	Hamis
Hamis	Igaz	Hamis	Igaz
Hamis	Hamis	Igaz	Igaz
Hamis	Hamis	Hamis	Hamis

```
logicalFunctionA :: Bool -> Bool -> Bool -> Bool
logicalFunctionA True False False == True
logicalFunctionA False False False == False
```