

2. házi feladat

Kategória:

Házi feladatok

Végső határidő:

3/2/2025, 11:59 PM (Beadva)

Próbálkozások száma:

Korlátlan

Kiírta:

Erdei Zsófia

Leírás:

Helyvektorok műveletei

Megjegyzés: A helyvektor kezdőpontja minden esetben a $(0,0)$ pont, tehát például az $(1,4)$ rendezett pár írja le a $(0,0)$ pontból az $(1,4)$ pontba mutató vektort a Descartes-féle koordinátarendszerben.

0. Modul

Definiálj egy modult `Hazi2` néven! A feladatok megoldásához **használd mintaillesztést!**

1. Vektorok összeadása, kivonása

Definiáld az `addV` és `subV` függvényeket, amik segítenek két helyvektort összeadni, kivonni.

```
addV :: (Double,Double) -> (Double,Double) -> (Double,Double)
subV :: (Double,Double) -> (Double,Double) -> (Double,Double)
```

2. Skalárral való szorzás

Definiáld a `scaleV` függvényt, aminek segítségével a vektor mérete szabadon változtatható, tehát szorozz meg egy vektort egy számmal.

```
scaleV :: Double -> (Double,Double) -> (Double,Double)
```

3. Skaláris-szorzás

Definiáld a `scalar` függvényt, ami kiszámolja két helyvektor skaláris szorzatát.

```
scalar :: (Double,Double) -> (Double,Double) -> Double
```

4. Oszthatóság

Készíts egy olyan függvényt `divides` néven, ami paraméterül vár két Integer típusú egész számot és megmondja, hogy az első osztója-e a másodiknak. A matematikai definíció szerint a 0 osztója a 0 -ának! A negatív számokkal nem szükséges foglalkozni, feltehető, hogy negatív számot paraméterként ez a függvény soha nem fog kapni.

Megjegyzés: Ez a függvény szeretne lenni a matematikából ismert $2 \mid 4$ jelölés. Pl. $2 \mid 4$ jelenti matematikában azt, hogy "Kettő osztója négynek".

```
divides :: Integer -> Integer -> Bool
```

5. Különböző típusú egész számok

Készítsünk egy `add` függvényt, amely két különböző típusú egész számot össze tud adni.

```
add :: Integer -> Int -> Integer
```

Tesztetek

FIGYELEM: a tesztek mindig tájékoztató jellegűek; ha a kód ezeknek megfelel, még nem jelenti azt, hogy jó és el lesz fogadva, érdemes tovább tesztelni, próbálgatni, legfőképpen a szélsőséges eseteket.

```
addV (1,2) (4,5) == (5,7)
addV ((-1),5) (7,8) == (6,13)
abs (fst (addV (1.22, 3.2) (5.48, 3.8)) - 6.7) < 0.00001
abs (snd (addV (1.22, 3.2) (5.48, 3.8)) - 7.0) < 0.00001
subV (1,2) (4,5) == ((-3),(-3))
subV (10,9) (7,8) == (3,1)
abs (fst (subV (1.22, 3.2) (5.48, 3.8)) + 4.26) < 0.00001
abs (snd (subV (1.22, 3.2) (5.48, 3.8)) + 0.6) < 0.00001
scaleV 6 (5,0) == (30,0)
scaleV (-2) (6,7) == ((-12),(-14))
scaleV 0 (4.5, (-7.897)) == (0.0,0.0)
abs (fst (scaleV 1.764 (3.45, (-0.23))) - 6.0858) < 0.00000001
abs (snd (scaleV 1.764 (3.45, (-0.23))) + 0.40572) < 0.00000001
scalar (2,4) ((-2), 1) == 0
scalar (5, 13) (4, 10) == 150
abs (scalar (3.25, 4.46) ((-0.75), 0.5) + 0.2075) < 0.00000001
0 `divides` 0
not (0 `divides` 12)
not (0 `divides` 100)
12 `divides` (0 :: Integer)
10 `divides` 0
120 `divides` 0
not (12 `divides` 3)
3 `divides` 12
1 `divides` 2
1 `divides` 3
1 `divides` 4
1 `divides` 13
not (13 `divides` 1)
add (2 :: Integer) (3 :: Int) == (5 :: Integer)
add (2 :: Integer) (3 :: Int) == (5 :: Integer)
add (2 :: Integer) (3 :: Int) == (5 :: Integer)
add (2 :: Integer) (3 :: Int) == (5 :: Integer)
add (5 :: Integer) (4 :: Int) == (9 :: Integer)
add (5 :: Integer) (4 :: Int) == (9 :: Integer)
```

Megoldás



Név:
solution.zip

Feltöltés ideje:
2/24/2025, 8:25 AM

Értékelés:

Státusz:
Elfogadva

Feltöltések száma:
4

Értékelte:
Szávó Tamás

Megjegyzések:
Az utolsó feladatnál az első fromIntegral felesleges, amúgy jó a megoldás

Automatikus tesztelés eredményei	
✔ #1	