

FIGYELEM!! Ha valahol a továbbiakban a feladatokban olyat láttok, hogy `<u>"Bónusz feladat"</u>` és nem számozott, akkor azt a feladatot *nem kötelező* megcsinálni. Azon fajta feladatok valami érdekes dolgot mutatnak be, esetleg olyan koncepciót gyakoroltatnak, amik az 5-öst segíthetik könnyebben elérni.

FIGYELEM!! A továbbiakban nem lesz kiírva a feladatokhoz, hogy "adjátok meg a típust", ez implicit minden egyes háziban feladat, hogy a függvény lehető legáltalánosabb típusát adjátok meg, amennyiben maga a feladat szövege nem rendelkezik másképp (pl. a feladatban meg van adva, hogy mi kell legyen a függvény típusa).

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 koordináta-rendszerben.

0. Modul

Definiálj egy modult `Hazi2` néven!

1. Vektorok összeadása, kivonása

Definiáld az `addV :: (Double,Double) -> (Double,Double) -> (Double,Double)`, `subV :: (Double,Double) -> (Double,Double) -> (Double,Double)` függvényeket, amik segítenek két helyvektort összeadni, kivonni.

2. Skalárral való szorzás

Definiáld a `scaleV :: Double -> (Double,Double) -> (Double,Double)` függvényt, aminek segítségével a vektor mérete szabadon változtatható, tehát szorozz meg egy vektort egy számmal.

3. Skaláris-szorzás

Definiáld a `scalar :: (Double,Double) -> (Double,Double) -> Double` függvényt, ami kiszámolja két helyvektor skaláris szorzatát.

Bónusz feladat: Vektor hossza

Adjuk meg a helyvektorok hosszát a `vecLength :: (Double,Double) -> Double` függvényt megírva.

Segítség: Ez a feladat egy kis önálló utánanézést igényel, a megoldásban szükséges egy olyan függvény, amit nem mutattam. Utánanézni esetleg a `Floating` nevű osztályra rákeresve lehet ghci-ben vagy `hoogle`-ön.

Tesztesetek

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
```