

- 1) Döntsük el a koordinátpárokról, hogy az origóban vannak-e vagy sem!
origo :: (Int,Int) -> Bool

Tesztesetek:

origo (0,0)
not (origo (2,1))
not (origo (0, 2))

- 2) Add meg a kizáró vagy (xor) függvénytörzsét a **legegyszerűbb** alakban!
xor :: Bool -> Bool -> Bool

Tesztesetek:

not (xor True True)
not (xor False False)
xor True False
xor False True

- 3) Adjuk meg azt a függvényt, amely eldönti egy számról, hogy az egy "kis" prím-e! Kis prím alatt az egyjegyű prímeket értjük (2,3,5,7). Ezekre a számokra igazat, minden egyéb számra hamisat adjon vissza!
isSmallPrime :: Int -> Bool

Tesztesetek:

isSmallPrime 2
isSmallPrime 3
isSmallPrime 5
not (isSmallPrime 11)
not(isSmallPrime 4)

- 4) Adjuk meg az a függvényt amely egy koordinátát középpontosan tükröz az origóra.
invertO :: (Int, Int) -> (Int, Int)

Tesztesetek:

invertO (2,3) == ((-2),(-3))
invertO (4,0) == ((-4),0)
invertO (0,0) == (0,0)
invertO ((-2),3) == (2,(-3))

- 5) Adjuk meg hogy két bekért szám közül a második osztható e az elsővel!
divides :: Integral a => a -> a -> Bool

Tesztesetek:

divides 0 0
divides 5 10
divides 6 0
not (divides 0 6)
not (divides 4 2)

- 6) Adjuk meg két bekért vektor összegét, különbségét és skaláris szorzatát külön függvényekben!

add :: (Num a, Num b) => (a,b) -> (a,b) -> (a,b)

sub :: (Num a, Num b) => (a,b) -> (a,b) -> (a,b)

scalar :: (Num a) => (a,a) -> (a,a) -> a

Tesztesetek:

add (2,6) (1, (-5)) == (3,1)
add (1,1) (2,2) == (3,3)
sub (2,6) (1,(-5)) == (1,11)
sub (1,1) (2,2) == ((-1),(-1))
scalar (2,6) (1,(-5)) == (-28)
scalar (1,1) (2,2) == 4
scalar (100,100) (0,0) == 0