Házi feladat

A házi feladatot egy LastHomework nevű modulként kell beadni. Ez egy remek gyakorlási lehetőség a vizsgára, ajánlom, hogy mérjétek le, hogy mennyire fértek bele a vizsga időtartalmába.

1. Melyik listában van az elem?

Adott három lista és egy elem. Döntsd el, melyikben található a keresett elem! Ha egyikben sem, az eredmény 0.

```
which :: ([Char], [Char], [Char]) -> Char -> Int

which (['a', 'b'], ['c','d'], ['e','f']) 'b' == 1
which (['a', 'b'], ['c','d'], ['e','f']) 'd' == 2
which (['a', 'b'], ['c','d'], ['e','f']) 'f' == 3
which (['a', 'b'], ['c','d'], ['e','f']) 'g' == 0
which (['a', 'b'], ['a'...'z'], ['e','f']) 'x' == 2
which (['a', 'b'], ['a'...'z'], ['e','f']) 'a' == 1
```

2. Dominók illeszthetősége

Állapítsd meg, hogy két dominólap illeszthető-e egymáshoz! Ez akkor áll fenn, ha a dominók két oldala közül van, amelyiken azonos számú pont van. Egy dominót a két oldalán lévő pontok számával jelölünk. Elég az első dominó jobb oldalát és a második bal oldalát összevetni.

```
matches :: (Int, Int) -> (Int, Int) -> Bool

matches (2, 4) (4, 3)

matches (1, 5) (5, 4)

not (matches (2, 4) (3, 6))

not (matches (1, 5) (3, 7))
```

3. Nagybetűsítés

Alakítsuk nagybetűvé egy szöveg első betűjét! Ha az első karakter nem betű, akkor hagyjuk a szöveget változatlanul!

4. Maybe-csere

```
Cserélj ki egy Maybe belsejében lévő értéket egy másikra! A Nothing-ból Nothing lesz.

swap :: Maybe a -> b -> Maybe b

swap (Just 'a') True == Just True
swap (Just "hello") 'b' == Just 'b'
swap Nothing False == Nothing
```

5. Fájljogosultságok

Unix rendszerekben a fájlok engedélyeit szokás szimbolikusan (pl. "rwx") és számszerűen (pl. 7) is jelölni.

Valósítsd meg a szimbolikus jelölésről számszerűre konverziót!

Feltesszük, hogy a bemenetben csak 'r', 'w' és 'x' betűk szerepelnek, mindegyik csak legfeljebb egyszer.

A betűk tetszőleges sorrendben lehetnek.

```
numeric :: String -> Int

numeric "r" == 4
numeric "w" == 2
numeric "x" == 1
numeric "rwx" == 7
numeric "xrw" == 7
numeric "xrw" == 6
numeric "wr" == 6
numeric "wr" == 6
numeric "rx" == 5
numeric "wx" == 3
```

6. Pitagoraszi számhármasok

Gyűjtsd egy listába a pitagoraszi számhármasokat! Ezek azok az a, b, c számhármasok, melyekre teljesül $a^2 + b^2 = c^2$.

Elegendő csak \$1 <= a, b, c <= 100\$ tartományban keresni.

```
Az ismétléseket kerüljük: (3, 4, 5) és (4, 3, 5) közül csak az egyik szerepeljen.

Megjegyzés: a tesztesetben használt sort függvény a Data.List modulból érhető el.

pythagoreans :: [(Int, Int, Int)]

take 5 (sort pythagoreans) == [(3,4,5), (5,12,13), (6,8,10), (7,24,25), (8,15,17)]
```

7. Van-e hosszú szó?

Vizsgáld meg, van-e legalább \$n > 0\$ betűből álló szó egy szövegben!

```
hasLongWord :: Int -> String -> Bool

hasLongWord 5 "hello world"
hasLongWord 4 "alma fa"
hasLongWord 3 "the car"
hasLongWord 9 "princess bubblegum"
not (hasLongWord 1 "")
not (hasLongWord 7 "king of ooo")
not (hasLongWord 5 "alma fa")
```

8. Minimális szélesség

Egészíts ki egy szöveget szóközökkel balról megadott hosszúságra! Ha a szöveg eleve hosszabb volt a meghatározottnál, ne vágj le belőle!

```
align :: Int -> String -> String
align 5 "alma" == " alma"
```

```
align 6 "alma" == " alma"
align 2 "alma" == "alma"
align (-8) "korte" == "korte"
align 10 "korte" == " korte"
```

9. Fejelem módosítása

Egy lista első elemét módosítsd egy Maybe-t adó f függvénnyel! Ha f egy Nothing-ot ad, töröld a lista első elemét! Ha Just x-et, akkor cseréld le az első elemet x-re!

10. Hosszabb-e a lista, mint \$n\$?

Írd meg a length egy olyan változatát, mely megvizsgálja, hogy hosszabb-e egy lista, mint egy előre megadott \$n >= 0\$ méret!

```
isLonger :: [a] -> Int -> Bool

isLonger [1..100] 50
isLonger [1..] 50
isLonger [1,2,3] 2
not (isLonger [1,2,3] 3)
not (isLonger [1,2,3] 4)
not (isLonger [] 10)
not (isLonger [] 0)
```

11. Ékezetes betűk cseréje

Cseréld ki az ékezetes betűket az ékezet nélküli párjukra! Elegendő csak a kisbetűkkel foglalkozni.

```
removeAccents :: String -> String

removeAccents "hétfő" == "hetfo"
removeAccents "időjárás" == "idojaras"
removeAccents "helyesírás" == "helyesiras"
removeAccents " == ""
removeAccents "telefon" == "telefon"
removeAccents "köpül" == "kopul"
removeAccents "hűvös" == "huvos"
```

12. Aláhúzásjelek levágása

Távolítsd el az aláhúzásjeleket egy szöveg elejéről és végéről!

13. Kő-papír-olló

Definiálj egy RPS adatszerkezetet a kő-papír-olló játékhoz! Három lehetőség a Rock, Paper, Scissors. Kérj legalább egyenlőségvizsgálatot is (deriving (Eq))! Definiálj egy függvényt, mely megmondja, melyik jel melyiket üti!

```
beats :: RPS -> RPS

beats Paper == Rock
map beats [Rock, Paper, Scissors] == [Scissors, Rock, Paper]
```

14. Kő-papír-olló játszma

Két játékos kő-papír-ollót játszik. Felírják, hogy ki milyen jelet mutatott. Segíts nekik megszámolni, hogy az első játékos hányszor nyert!

Feltesszük, hogy mindkét lista azonosan hosszú.

```
firstBeats :: [RPS] -> [RPS] -> Int

firstBeats [Rock] [Paper] == 0
firstBeats [Rock] [Scissors] == 1
firstBeats [Paper, Scissors] [Rock, Paper] == 2
firstBeats [Paper, Scissors, Paper] [Rock, Paper, Scissors] == 2
firstBeats (replicate 20 Paper) (replicate 20 Rock) == 20
```

15. Hőmérséklet mérése

Definiálj egy Temperature adatszerkezetet a levegőhőmérséklet mérésekhez! A hőmérsékletet mérjük nappal és éjszaka (Daytime és Night).

Döntsd el egy mérésről, hogy nappal történt-e vagy éjszaka!

```
isDaytime :: Temperature -> Bool

isDaytime (Daytime 15)
isDaytime (Daytime 0)
isDaytime (Daytime (-2))
not (isDaytime (Night (-4)))
not (isDaytime (Night 0))
not (isDaytime (Night 2))
```

16. Szélsőséges hőmérsékletek

Adott egynapi négyóránkénti méréssorozat. Állapítsd meg a legmagasabb nappali, és legalacsonyabb éjszakai hőmérsékletet!

```
extremes :: [Temperature] -> (Int, Int)

extremes [Night (-5), Night (-6), Daytime 0, Daytime 3, Daytime 5, Daytime 1, Night (-7)]
== (5, -7)
extremes [Night 5, Night 0, Daytime 1, Daytime 10, Daytime 8, Daytime 5, Night 2]
== (10, 0)
extremes [Night 3, Night 0, Daytime 1, Daytime 10, Daytime 8, Daytime 15, Night 7]
== (15, 0)
```