

Funkcionális programozás 8. gyakorlat

1. Definiáld újra a `sum` függvényt!

```
sum' [4,5,6] == 15
sum' []      == 0
sum' [1..10] == 55
```

2. Definiáld újra a `last` függvényt! A `last` üres listán nincs értelmezve.

```
last' [4,5,9] == 9
last' [5]     == 5
last' [1..100] == 100
```

3. Definiáld újra az `and` függvényt!

```
and' [True, True, True, True]
and' []
not (and' [True, False, True, True])
not (and' [False, False, False])
```

4. Definiáld újra az `or` függvényt!

```
or' [True, True, True, True]
or' [True, True, False, True]
not (or' [])
not (or' [False, False, False, False])
```

5. Definiáld újra a `repeat` függvényt!

```
take 3 (repeat 'a') == "aaa"
take 10 (repeat 'a') == replicate 10 'a'
```

6. Definiáld újra a `replicate` függvényt!

```
replicate' 5 'a' == "aaaaa"
replicate' 10 'p' == "pppppppppp"
```

7. Egészíts ki egy szöveget megadott szélességűre! Ha netán a megadott szélesség kisebb lenne, mint a szöveg, akkor ne vágj le belőle!

```
format 10 "haskell" == "  haskell"
format 3 "haskell"  == "haskell"
```

8. Szűrj be egy egész számot nagyság szerinti helyére!

```
insert 2 [4, 5, 9] == [2, 4, 5, 9]
insert 6 [4, 5, 9] == [4, 5, 6, 9]
insert 11 [4, 5, 9] == [4, 5, 9, 11]
insert 5 [4, 5, 9] == [4, 5, 5, 9]
insert 2 []        == [2]
```

9. Valósítsd meg a beszűrő rendezést! Ez egy üres listába szűrja be egyesével a rendezetlen elemeket nagyság szerint megfelelő helyre.

```
sort [5, -2, 3] == [-2, 3, 5]
sort [] == []
```

10. *Fésülj össze két rendezett, egészekből álló listát úgy, hogy az eredmény is rendezett maradjon!

```
merge [4, 5, 7, 10] [2, 3, 6] == [2, 3, 4, 5, 6, 7, 10]
merge [5] [1, 2, 8] == [1, 2, 5, 8]
merge [2, 3] [] == [2, 3]
```

11. *Valósítsd meg az összefésülési (merge sort) rendezést, mely egy listát kettébont középen, rendezi a két félt, majd összefuttatja a kettőt!

```
mergeSort [5, 2, -4, 9, 3] == [-4, 2, 3, 5, 9]
mergeSort [] == []
```

12. Definiálj egy (rekurzív) `breakOn` függvényt, mely egy szövegben egy megadott karakter előtti és utáni darabot visszaadja!

```
breakOn ' ' "haskell is cool" == ("haskell", " is cool")
breakOn ' ' "is cool" == ("is", " cool")
breakOn '/' "haskell/gyak/gyak.hs" == ("haskell", "/gyak/gyak.hs")
```

13. Definiálj egy `splitOn` függvényt, mely adott karakter mentén darabol egy Stringet!

```
splitOn ' ' "haskell is cool" == ["haskell", "is", "cool"]
splitOn '/' "haskell/gyak/gyak.hs" == ["haskell", "gyak", "gyak.hs"]
splitOn '/' "haskell" == ["haskell"]
splitOn '/' "" == []
```

14. Bonts fel cellákra egy csv fájl tartalmát! A fájl sorokra való bontásához érdemes használni a `lines` függvényt.

```
csv "nev,neptun,jegy\nEndre,ADG1K5,5\nAnna,KRJ25L,5"
== [["nev","neptun","jegy"],["Endre","ADG1K5","5"],["Anna","KRJ25L","5"]]
```