

# Funkcionális programozás 9. gyakorlat

1. Definiáld újra a repeat függvényt!

```
take 3 (repeat 'a') == "aaa"
take 10 (repeat 'a') == replicate 10 'a'
take 2 (repeat "haskell") == ["haskell", "haskell"]
```

2. Definiáld újra a take függvényt!

```
take 4 "hello world" == "hell"
take 4 "" == ""
take (- 2) "hello world" == ""
take 8 [True, False] == [True, False]
take 0 [True, False] == []
```

3. Definiáld újra a drop függvényt!

```
drop 4 "hello world" == "o world"
drop 4 "" == ""
drop (- 2) "hello world" == "hello world"
drop 8 [True, False] == []
drop 0 [True, False] == [True, False]
```

4. Adott egy nyelvazonosító (például hu-HU vagy en-US), mely két részből, nyelvből és régióból tevődik össze. Bonts fel egy nyelvazonosítót nyelvre és régióra! Feltesszük, hogy jól formázott a bemenet. Mindkét rész két-két karakterből áll.

```
langAndRegion "hu-HU" == ("hu", "HU")
langAndRegion "en-US" == ("en", "US")
langAndRegion "fr-CA" == ("fr", "CA")
```

5. Definiáld újra a zip függvényt!

```
zip [] [True, False] == []
zip [1,2,3] [] == []
zip [1,2,3] [True, False] == [(1, True), (2, False)]
zip "abc" [1..] == [('a', 1), ('b', 2), ('c', 3)]
```

6. Definiáld újra az unzip függvényt!

```
unzip [('a', 1), ('b', 2), ('c', 3)] == ("abc", [1,2,3])
unzip [(1, True), (2, False)] == ([1,2], [True, False])
unzip [] == ([], [])
```

7. Adott egy szöveges fájl. Hanyadik sorai üresek? A sorszámozás 1-től induljon! Az utolsó sor nem számít üresnek.

Sorok feldarabolására a lines használható.

```
empty "elso\nmasodik\n\nnegyedik\n" == [3]
empty "theme=dark\n\ntoolbar=0\n\nicons=gnome" == [2,4]
empty "" == []
```

8. Bonts fel egy listát két részre egy adott pozíción!

```
splitAt 0 [True, False] == ([], [True, False])
splitAt 1 [True, False] == ([True], [False])
splitAt 2 [] == ([], [])
splitAt 2 "hello" == ("he", "llo")
splitAt (- 2) "hello" == ("", "hello")
```

9. Definiáld újra a nub függvényt! Ez minden elemet csak egyszer tart meg, annak is csak a legelső előfordulását.

```
nub' :: Eq a => [a] -> [a]
nub' [1,2,2,3,2,4] == [1,2,3,4]
nub' "hello world" == "helo wrd"
nub' [] == []
```

10. Definiáld újra a concat függvényt!

```
concat' [[1,2], [3,4,5]] == [1,2,3,4,5]
concat' [[1,2]] == [1,2]
concat' [] == []
concat' ["hello", "haskell"] == "hellohaskell"
```