

- 1) Definiáljuk újra a (++) függvényt (+++) néven! (A függvény két azonos típusú elemekből álló listát fűz össze)

(+++) :: [a] -> [a] -> [a]

Tesztesetek:

[1,2,3] +++ [4,5,6] == [1,2,3,4,5,6]

[] +++ [4,5,6] == [4,5,6]

[1,2,3] +++ [] == [1,2,3]

[] +++ [] == []

- 2) Definiáljuk újra a (!!) függvényt (!!!) néven! (A függvény az indexelésnek megfelelően adja vissza az annyiadik elemet)

(!!!) -> [a] -> Int -> a

Tesztesetek:

[1,2,3] !!! 2 == 3

[1,2,3] !!! 0 == 1

- 3) Írjunk rekurzívan egy függvényt, amely bekér egy Stringet és egy számot, majd megállapítja, hogy a String hossza nagyobb e mint a szám! (**length nem használható**)

longerThan :: Integral i => [a] -> i -> Bool

Tesztesetek:

longerThan [1,2,3] 1

not (longerThan [1,2,3] 3)

not (longerThan [1,2,3] 4)

- 4) Egy camelCase szöveget bontsunk fel szavakra! (a **Data.Char**- ban található **isUpper** függvény hasznokra lehet)

camelToWords :: String -> String

Tesztesetek:

camelToWords "ezTobbSzo" == "ez Tobb Szo"

camelToWords "" == ""

camelToWords "camelToWords" == "camel To Words"

- 5) Állítsuk elő egy lista összes lehetséges végződését!

tails' :: [a] -> [[a]]

Tesztesetek:

tails' [1,2,3] == [[1,2,3], [2,3], [3], []]

tails' [] == [[]]

tails' [1,2,3,4,5] == [[1,2,3,4,5], [2,3,4,5], [3,4,5], [4,5], [5], []]