

- 1) Alakítsunk át egy listát rendezett párokból álló listává úgy, hogy az elemeket kettesével berakjuk egy párba majd befűzzük a listába, ha páratlan elemet tartalmaz az a lista akkor az utolsó elemet elhagyjuk.

pairs :: [a] -> [(a,a)]

Tesztesetek:

pairs [1,2,4,48,6,15,12,31,6516,4,6] == [(1,2),(4,48),(6,15),(12,31),(6516,4)]

pairs ['a','g','k','l','h'] == [('a','g'),('k','l')]

pairs [True, True, False, True, False] == [(True,True),(False,True)]

- 2) Írjunk függvényt, amely megadja, hogy egy String hányszor tartalmazza egy megadott másik Stringet.

countString :: String -> String -> Int

Tesztesetek:

countString "al" "alalaalmalal" == 6

countString "a" "aaababbbbbaaaabbaaa" == 11

countString "k" "aaababbbbbaaaabbaaa" == 0

- 3) Adjuk meg, hogy egy számokat tartalmazó lista azonos paritású számokat tartalmaz-e.

sameParity :: [Int] -> Bool

Tesztesetek:

sameParity [0,2,2,2,2,2,4,4,4,8,8,8,4,4,2,6]

sameParity [1,5,7,5,3,1,9,13]

not (sameParity [1,2,3,4,5,6,7,8,9])

- 4) Definiáljuk azt a függvényt, amely eldönti egy listáról, hogy az rendezett-e! Egy listát rendezettnek tekintünk, ha az elemei növekvő sorrendben vannak.

isSorted :: Ord a => [a] -> Bool

Tesztesetek:

isSorted [1..10]

not \$ isSorted [10,9,8]

isSorted []

isSorted [1]

not \$ isSorted [1,5,9,5,9,10]

- 5) Definiáljuk azt a függvényt, amely egy listából kiszűri az egymás melletti ismétlődő elemeket!

uniques :: Eq a => [a] -> [a]

Tesztesetek:

uniques [1,1,1,0,0,1,0,0,0] == [1,0,1,0]

uniques "aaaallllmmmmaaaaaa" == "alma"

uniques [True,True,False,True,False,False] == [True,False,True,False]

uniques [] == []

uniques [4] == [4]