

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

2. Levenshtein-távolság

13 pont

A következő algoritmus két, maximum 25 karakter hosszú karakterláncról megállapítja, hogy mekkora a Levenshtein-távolságuk, azaz minimálisan hány karakterenkénti művelet (beszúrás, törlés, csere) kell ahhoz, hogy az egyik karakterláncot a másikra átalakítsuk. Kódolja az algoritmust a választott programozási nyelven! Az elkészült program forráskódját mentse LD néven!

A megoldás során vegye figyelembe a következőket:

- A "Hossz()" függvény a karakterlánc hosszát adja meg.
- A választott programozási nyelvtől függően eltérő jelölésű operátorokat és függvényeket kell alkalmaznia.
- A "Térj vissza!" utasítás megszakítja a függvény futását, és meghatározza annak visszatérési értékét.

Függvény Min(a: Egész, b: Egész): Egész

Ha a<b akkor

Min:= a

különben

Min:= b

Elágazás vége

Függvény vége

Függvény LDTav(s1: Szöveg, s2: Szöveg): Egész

Változó h1, h2:Egész

Változó c:Egész

Változó tömb m[0..26, 0..26]:Egész

h1:= Hossz(s1)

h2:= Hossz(s2)

Ha h1= 0 akkor Térj vissza h2-vel

Ha h2= 0 akkor Térj vissza h1-el

Ciklus i:= 0-tól h1-ig (+1 lépésközzel)

m[i,0]:= i

Ciklus vége

Ciklus j:= 0-tól h2-ig (+1 lépésközzel)

m[0,j]:= j

Ciklus vége

Ciklus i:= 1-től h1-ig (+1 lépésközzel)

Ciklus j:= 1-től h2-ig (+1 lépésközzel)

c:= 1

Ha s2[j-1]= s1[i-1] akkor c:= 0

m[i,j]:= Min(Min(m[i-1,j]+1,m[i,j-1]+1),m[i-1,j-1]+c)

Ciklus vége

Ciklus vége

LDTav:= m[h1,h2]

Függvény vége

Program LD

Ki: LDTav("alma","halmaz")

Program vége.