

Feladat

Egy kép feldolgozása a feladat. Az egyszerűség kedvéért a képet egy karaktermátrix tárolja. A pixelek színét egy betű-karakter jelzi most. A kép több, nem háttérszínű foltból áll. E foltok „egyedi azonosítást” kell elvégezni. A kép felső és alsó pixel-sorai, baloldali és jobboldali szélső oszlopai háttérszínűek.

Bemenet (text-fájl: `kep.be`):

1. sor: N M, a kép méretei, azaz pixel-sorok és – oszlopok száma (N,M≥2; a példában: 5 5.
- 2-N+1. sor: a kép 1-N. pixel-sora, amelyek pontosan M darab, színeket kódoló jelből áll ('A'..'Z'); a példában előforduló 3 színt az 'S', a 'G' és az 'F' betű kódolja. Mivel a „ke-re-t” S színű, ezért tudjuk, hogy S a háttér-szín.

1						5	5
2						S	S
3						S	F
4						S	F
5						S	S
	1	2	3	4	5	S	S

Egy 5x5-ös példa-kép... ... és fájlbeli megfelelője.

1. ábra

Kimenet (text-fájl: `kep.ki`):

1. sor: db, a képet alkotó alakzatok száma; a példában: 2.
- 2-N+1. sor: a módosított kép pixel-sorai, amelyek most is pontosan M karakterből állnak; az alakzat határait a 'h', az egyes alakzatokhoz tartozó pixeleket rendre az '1', '2' ... '9' karakterek jelzik.

2	Első alakzat a 3 pixelből álló,
hhhhS	amely pixeleit az '1' színnel
h11hS	kódoltuk. A második egyet-
h1hhh	len pixele a '2' szint kapott.
hhh2h	Mindkét alakzatot 'h' színű
Sshhh	pixelek veszik körbe.

A fenti kimenete, valamint a magyarázat.

2. ábra

Megjegyzések:

1. egyszerű iteratív algoritmus nincs annak fel-
derítésére, hogy mely pixelek tartoznak ugyan-
ahhoz az alakzathoz, ezért célszerű *rekurzió*-
ban gondolkodni;
2. él-szomszédos pixelek alkothatnak csak egy
alakzatot;
3. feltehető, hogy 9-nél több alakzat nincs a ké-
pen, ezért használhatók egyedi színek gya-
nánt a számjegy-karakterek.

1									hhhhhhhhh
2									h1111h2h
3									h1h1hh2h
4									h1hh2h2h
5									hhh2222h
6									SShhhhhhh
	1	2	3	4	5	6	7	8	

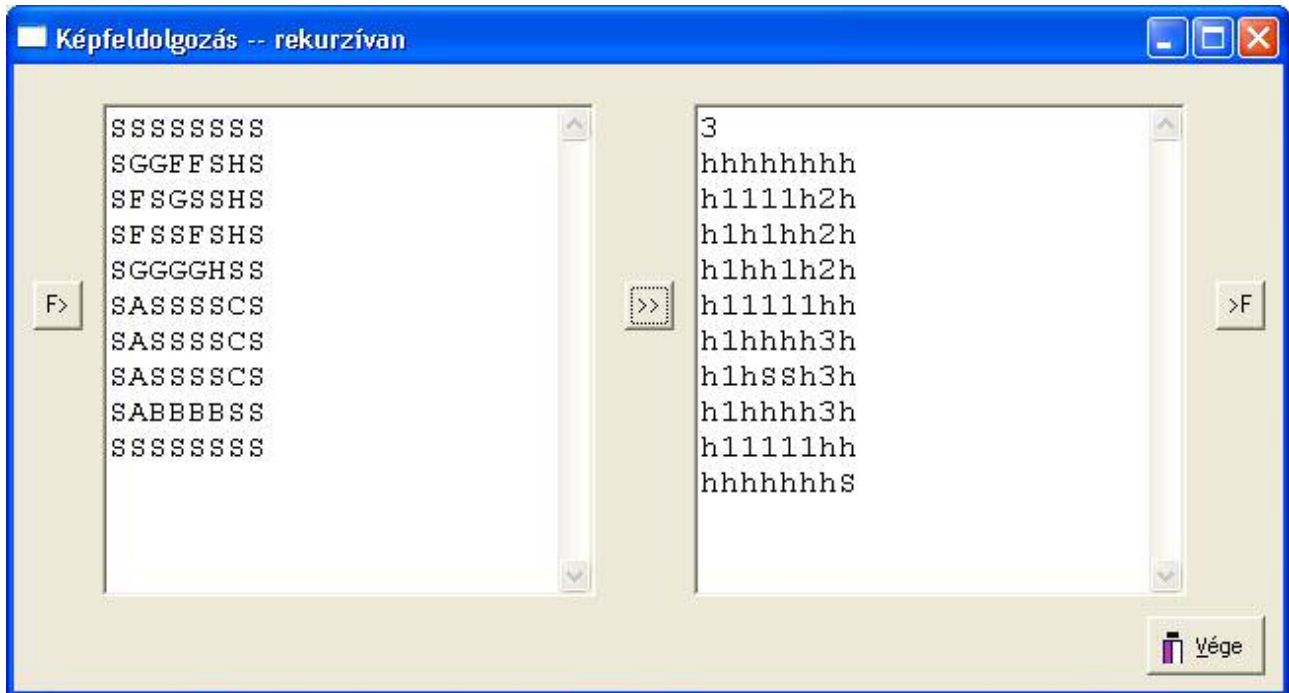
Egy további kép... ...és transzformációja.

3. ábra

Feladata olyan alkalmazás írása, amely

- beolvasza a felhasználó által megadott nevű (* .be), „szintaktikusan” helyes fájlból a képet,
- megjeleníti a színmátrixot valamilyen erre alkalmas grafikus komponensben (stringGrid, memo^{*}), azaz a fájl tartalmát az első sora nélkül,
- elvégzi a fenti feldolgozást,
- megjeleníti az eredményeket egy erre alkalmas grafikus komponensben (stringGrid, memo),
- kimentí egy ’.ki’ kiterjesztésű, a fent leírt szerkezetű text-fájlba;
- módot biztosít, hogy a bementi grafikus komponensben „kézzel” is módosításokat teghessen a felhasználó (a szintaktikus helyesség megőrzése mellett),
- gombokkal vagy menüvel vezérelhető a program: fájlkezelés, kódolás, kilépés.

* Memo estében szemmel jól érzékelhető képet csak valamely nem proporcionális font (pl. Courier New) kiválasztásával kapunk.

Egy GUI példa:**Pontozás:**

Tevékenység	Pont	Tevékenység	Pont
Minden funkcióhoz grafikus kellék a formon	6	Az input szerkeszthető kézzel (és feldolgozható)	2
Fájl betölthető az input komponensbe	2	Az eredmény (bármilyen) fájlba vihető	1
Valami átkerül az input komponensből az outba	1	Helyes az eredmény a fájlban	1
Az 1. próbafájl – alakzatszám OK (fájlban/ablakban)	2	A funkciók elérhetősége kezdetben megfelelő	1
Az 1. próbafájl – alakzatszínezés OK (f/a)	2	A funkciók elérhetősége jól módosul betöltés után	1
Az 1. próbafájl – él határpontok OK (f/a)	2	Pluszok	4
Az 1. próbafájl – sarok határpontok OK (f/a)	2		
A 2. próbafájl – alakzatszám OK. (f/a)	2	A 3. próbafájl – alakzatszám OK. (f/a)	2
A 2. próbafájl – alakzatszínezés OK. (f/a)	3	A 3. próbafájl – alakzatszínezés OK. (f/a)	3
A 2. próbafájl – él határpontok OK (f/a)	2	A 3. próbafájl – él határpontok OK (f/a)	2
A 2. próbafájl – sarok határpontok OK (f/a)	2	A 3. próbafájl – sarok határpontok OK (f/a)	2
Összesen:	45		
Értékelés			
Kettes, ha legalább:	15	Négyes, ha legalább:	29
Hármas, ha legalább:	22	Ötös, ha legalább:	36