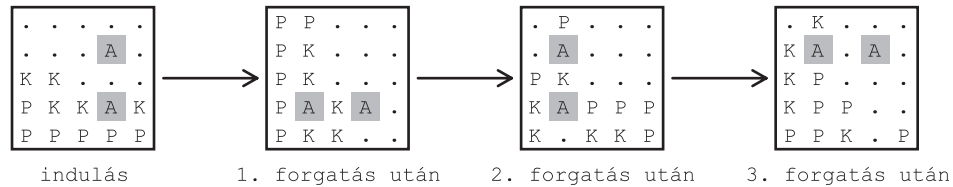
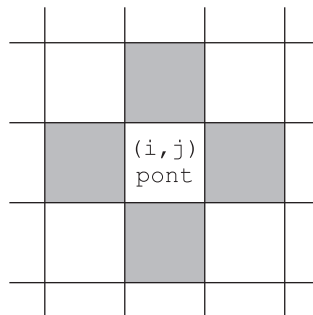


Egy $N \times N$ -es ($4 \leq N \leq 50$) tábla cellái üres vagy színes négyzetek az **I. 343.** feladatban leírtakhoz hasonlóan. A táblát függőlegesre állítva a színes négyzetek a legalacsonyabb üres helyig egymásra vagy az előre behelyezett és rögzített akadályig csúsznak a gravitáció miatt. Háromféle (piros, kék és zöld színű) négyzet lehetséges, amelyeket P, K és Z betűk jelölnek. Az üres cellát a . karakter, a rögzített helyű akadályokat az A betű jelöli. A táblát szegély veszi körbe, ami megakadályozza, hogy a színes négyzetek kiessenek.

A táblát csak jobbra lehet forgatni. A forgatás hatására a színes négyzetek újra a legalacsonyabb szabad cellába csúsznak, de egymásról nem esnek le. A rögzített akadályok és a forgatás keverő hatását vizsgáljuk.



Számoljuk meg induláskor és forgatás után, hogy hány olyan színes négyzet van a táblában, amelynek egynél több vele azonos színű szomszédja van. Szomszéd cellákon most négyet értünk, az *ábrának* megfelelően.



Készítsünk programot `i346` néven, amely a tábla forgatását, a színek keverését szimulálja.

A program első parancssori argumentuma a táblát leíró adatállomány neve legyen. A fájl első sora N ($4 \leq N \leq 50$) a tábla méretét és K ($1 \leq K \leq 100$) a forgatások számát adja meg. Az ezt követő N sor a tábla indulás előtti cellánkénti tartalmát adja meg.

A parancssor második argumentumaként megadott kimeneti állomány első sorában forgatásonként az olyan színes négyzetek számát írjuk ki szóközzel elválasztva, amelyeknek egynél több velük azonos színű szomszédjuk van. Az ezt követő N sorban a legutolsó forgatás utáni tábla tartalmát adjuk meg soronként.

Bemenet	Kimenet
5 3	5 5 2 3
.....	.K...
...A.	KA.A.
KK...	KP...
PKKAK	KPP..
PPPPP	PPK.P

Beküldendő a program forráskódja (`i346.pas`, `i346.cpp`, ...) és rövid dokumentációja (`i346.txt`, `i346.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.