

**1. feladat:** Azonosító kód (30 pont)

A processzorgyártó cégek megállapodtak abban, hogy milyen rendszert alkalmaznak az általuk gyártott processzorok azonosítására. Minden cég kap egy betűkészletet, és ezekből kell az azonosító kódot képeznie, úgy, hogy minden betű meghatározott számszor szerepeljen az azonosítóban. Például egy cég azt kapta, hogy minden azonosítója pontosan 3 db 'a' betűt, 2 db 'b' betűt és 1 db 'c' betűt tartalmazhat.

**Feladat:**

Készíts programot (KOD.PAS vagy KOD.C), amely adott kódra meghatározza a lexikografikus (ábécé szerinti) sorrendben rákövetkező szabályos kódot, ha van rákövetkező!

**Bemenet:**

A KOD.BE állomány első sorában a kódok száma ( $1 \leq N \leq 100$ ) van. A további  $N$  sorban az egyes kódok találhatóak. Minden kód legfeljebb 100 betűből állhat, csak az angol ábécé kis betűit tartalmazhatja.

**Kimenet:**

A KOD.KI állományba  $N$  sort kell írni! Az  $i$ -edik sorba a bemeneti állomány  $i+1$ -edik sorában lévő kód rákövetkezőjét kell írni, ha nincs rákövetkezője, akkor a NINCS szót.

**Példa:**

KOD.BE	KOD.KI
2	ababac
abaacb	NINCS
cbaaa	

2. feladat: Kirakó (40 pont)

Egy szókirakó játékban a játémező (egy négyzethálós téglalap) egyes négyzeteibe kell szavakat elhelyezni vízszintesen vagy függőlegesen. A szavak egymást keresztezhetik, ekkor természetesen a közös mezőben mindkét szóban azonos betűnek kell állnia! A játékban a szavakat csak a megadott sorrendben szabad lerakni, s minden lépésben úgy kell eljárni, hogy addig az átfedő betűk száma a lehető legnagyobb legyen! Feltehető, hogy minden újabb szónak legalább 1 átfedő betűje van a már letett szavakkal.

**Feladat:**

Készíts programot (KIRAKO.PAS vagy KIRAKO.C), amely megadja azt a szóelhelyezést, amelyben kirakás közben és a kirakás végén is a legtöbb átfedő betű található!

**Bemenet:**

A KIRAKO.BE állomány első sorában a kirakandó szavak száma ( $2 \leq DB \leq 15$ ) és a játémező mérete ( $1 \leq N$  sor,  $M$  oszlop  $\leq 30$ ), a második sorban pedig az első szó kezdőpozíciója ( $1 \leq OSZLOP \leq M$ ,  $1 \leq SOR \leq N$ ) és iránya (V vagy F – jelentése vízszintesen, illetve függőlegesen) szerepel egy-egy szóközzel elválasztva. A további **DB** sorban az egyes értelmes magyar szavak találhatóak (legfeljebb 10 karakteresek, ugyanaz a betű legfeljebb négyszer fordul elő bennük.).

**Kimenet:**

A KIRAKO.KI állományba **DB-1** sort kell írni, a KIRAKO.BE állomány második sorának megfelelő módon, azaz egyes szavak kezdőpozícióját és irányát, egy-egy szóközzel elválasztva.

Példa:

KIRAKO.BE	KIRAKO.KI
3 10 10	6 4 F
5 5 V	6 6 V
FA	
KAP	
PER	

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
1										
2										
3										
4						<b>K</b>				
5					<b>F</b>	<b>A</b>				
6						<b>P</b>	<b>E</b>	<b>R</b>		
7										
8										
9										
10										

**3. feladat:** Bányászrobot (30 pont)

Egy robotot helyezünk egy labirintusba, megjelölve a kezdő- és a célpozícióját. A robot el kell vezetni a célig, feltéve, hogy minden lépés 1 időegységbe kerül. A robot a labirintus járatain közlekedhet vízszintes vagy függőleges irányban a 4 szomszéd hely valamelyikére lépve, valamint képes a labirintus falait is kibontani. Egy egységnyi fal kibontása a robotnak  $F$  időegységbe kerül.

**Feladat:**

Készíts programot (BANYASZ.PAS vagy BANYASZ.C), amely megadja, hogy minimum mennyi idő múlva érhet a robot a kezdőpozícióból a célpozícióba!

**Bemenet:**

A BANYASZ.BE állomány első sorában a négyzetrács magassága és szélessége ( $1 \leq N, M \leq 100$ ), valamint a fal kibontásához szükséges idő ( $1 \leq F \leq 100$ ) van, egy-egy szóközzel elválasztva. A következő  $N$  sor mindegyike pontosan  $M$  karaktert tartalmaz, az üres helyeken szóközt, a foglalt helyeken \*-ot, a robot kezdőpozícióján K betűt, célpozícióján pedig C betűt.

**Kimenet:**

A BANYASZ.KI állományba egyetlen sort kell írni, azt a minimális időtartamot, ami alatt a robot a kezdőhelyről a célpozícióba érhet.

**Példa:**

BANYASZ.BE	BANYASZ.KI
4 6 5	8
*****	
*K *	
*****	
* C**	