

Feladat 1– Egér (Soricel)

100 pont

Egy négyzetes mátrix alakú labirintus egy meghatározott pozíciójában egy éhes egér tartozkodik, egy másik pozícióban pedig egy darab sajt található, de sajnos macskák is vannak a labirintusban, amelyek elkapják az egeret, amennyiben a macskák cellájával szomszédos 8 cella valamelyikére lép. Az egér az észak, dél, kelet vagy nyugat irányban található szomszédos cellák egyikére léphet. Határozd meg azt a minimális (optimális) lépést, amellyel eljut az egér a sajthoz anélkül, hogy a macskák elkapják.

Bemeneti adatok

A **soricel.in** állomány első sorában 2 természetes szám található egy szóközzel elválasztva: **n** a mátrix mérete és **m** a macskák száma. Az állomány második sorában 4 természetes szám van, amelyek az egér és a sajt pozícióját határozzák meg, míg a következő **m** sor mindegyikében 2 természetes szám mutatja az **m** darab macska pozícióját a mátrixban.

Kimeneti adatok








A **soricel.out** állomány első sora a legkevesebb lépést kifejező számot kell tartalmazza, ahogyan eljut az egér a sajthoz anélkül, hogy megfognák az macskák, vagy az **imposibil** szót, ha az egér nem tud eljutni a sajthoz.

Megkötések és pontosítások








- $1 \leq n \leq 100$
- $1 \leq m \leq 200$

Példa

Soricel.in
10 5
3 2 1 10
8 1
1 7
3 6
5 6
9 9

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Soricel.out
18

	1	2	3	4	5	6	7	8	9	10
1										18 
2										17
3			1					14	15	16
4			2					13		
5			3					12		
6			4					11		
7			5	6	7	8	9	10		
8										
9										
10										

Maximális futási idő / teszt: 0.1 másodperc; Maximális memória: 2MB amelyből 1 MB verem;
A forráskód maximális mérete: 10 KB.

Feladat 2– Üzenet (Mesaj)

100 pont

Mária és Dani gyakran üzennek egymásnak, de hogy senki más ne értse meg az üzeneteket, ezért kitalálnak egy kódolást. A kódolási algoritmus a következő szabályokat tartalmazza: mindegyik szóban, azt a mássalhangzót, amelyet egy magánhangzó követ a P betűre kell kicserélni, és ha egy magánhangzó előtt egy mássalhangzó volt az eredeti szövegben, akkor kódoláskor a magánhangzó után illesszük be azt a mássalhangzót ami előtte volt és még az A betűt.

A szabályok alapján, a két fiatal megírta a küldött üzenetek kódolási algoritmusát és egy algoritmust a kapott üzenetek dekódolásához, megfejtéséhez.

Követelmény

Egy adott szöveget ha el kell küldeni akkor határozd meg a kódolt megfelelőjét, míg ha a szöveg egy kapott üzenet, akkor határozd meg az eredeti üzenetet (dekódold).

Bemeneti adatok

A `mesaj.in` bemeneti állomány első sora ha az **1** számot tartalmazza akkor a második sorban lévő szöveget kell elküldeni vagyis kódolni kell, míg ha az első sorban lévő szám a **2** akkor a második sorban lévő szöveg a kapott üzenetet jelképezi és meg kell fejteni, dekódolni kell.

Kimeneti adatok

A `mesaj.out` kimeneti állomány első sora a kódolt vagy dekódolt szöveget fogja tartalmazni.

Megkötések és pontosítások

- A szöveg tartalmazhatja a következőket: az angol ábécé nagybetűit, számjegyet, szóközt és elválasztó karaktert (, . ! ?). Magánhangzók a következők: AEIOU.
- A bemeneti állományban található szöveg hossza < 256

Példa

mesaj.in	mesaj.out
1 TOAMNA ESTE PRIMAVARA IERNII	POTAAMPANA ESPETA PPIRAPAMAPAVAPARA IERPINAI
2 CPIRAPOTAGPARAPIFAA, SPITAINPATA PISA ARPATA	CRIPTOGRAFIA, STIINTA SI ARTA

Maximális futási idő / teszt: 0.1 másodperc;

Maximális memória: 2MB amelyből 1 MB verem;

A forráskód maximális mérete: 5 KB.