

Problema 1 – Șoricel

100 puncte

Într-un labirint sub forma unei matrice pătratică, într-o anumită poziție se află un șoricel înfometat, într-o altă poziție se găsește cașcaval, dar din păcate în labirint se găsesc și pisici care prind șoricelul în cazul în care ajunge în oricare dintre cele 8 pătrățele vecine cu ele. Șoricelul se poate deplasa în una din pozițiile din direcțiile N, S, E, V. Determinați numărul optim (minim) de pași prin care șoricelul ajunge la cașcaval fără să fie prins de pisici.

Date de intrare

Fișierul **soricel.in** conține pe prima linie 2 numere naturale separate printr-un spațiu: **n** dimensiunea matricei și **m** numărul pisicilor. Pe a doua linie a fișierului se dau 4 numere naturale reprezentând coordonatele șoricelului și a cașcavalului, iar pe următoarele **m** linii se dau câte 2 numere naturale reprezentând coordonatele celor **m** pisici.

Date de ieșire








Fișierul **soricel.out** va conține pe prima linie o valoare ce reprezintă numărul de pași minim prin care șoricelul ajunge la cașcaval fără să fie prins de pisici, sau mesajul **imposibil** dacă nu poate ajunge la cașcaval.

Restricții și precizări








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

Exemplu

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										

Timp maxim de execuție / test: 0.1 secunde; Memoria maximă disponibilă: 2MB din care 1 MB stivă;
Dimensiunea maximă a sursei: 10 KB.

PROBLEMA 2 – Mesaj

100 puncte

Maria și Dorel stabilesc un cod de transmitere a mesajelor între ei astfel încât să nu poată fi înțelese de alte persoane. Algoritmul de codificare pe care l-au stabilit este următorul: în fiecare cuvânt, o consoană urmată de o vocală se va înlocui cu litera P, după o vocală care are înaintea sa în textul original o consoană, se adaugă consoana precedentă și litera A.

În urma înțelegerii dintre ei și-au scris un algoritm de codificare a mesajelor transmise și un algoritm de decodificare a mesajelor primite.

Cerință

Dat fiind un text de trimis sa se obtina textul codificat, respectiv fiind dat un text primit să se obțină textul decodificat.

Date de intrare

Fișierul de intrare **mesaj.in** conține pe prima linie numărul **1** și pe linia a doua textul de trimis, respectiv conține pe prima linie numărul **2** și pe linia a doua textul primit.

Date de ieșire

Fișierul de ieșire **mesaj.out** va conține pe prima linie textul codificat sau textul decodificat.

Restricții și precizări

- Textul este format din cuvinte scrise cu litere mari din alfabetul englez, cifre, spații și alte caractere separatorii (, . ! ?). Șirul de vocale este AEIOU.
- În fișierul de intrare lungimea textului < 256

Exemplu

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

Timp maxim de execuție / test: 0.1 secunde;

Memoria maximă disponibilă: 2MB din care 1 MB stivă;

Dimensiunea maximă a sursei: 5 KB.