

Problema 1 – Graf

100 puncte

Se știe că într-un graf neorientat conex, între oricare două vârfuri există cel puțin un lanț iar lungimea unui lanț este egală cu numărul muchiilor care-l compun. Definim noțiunea *lanț optim între două vârfuri X și Y* ca fiind un lanț de lungime minimă care are ca extremități vârfurile X și Y. Este evident că între oricare două vârfuri ale unui graf conex vom avea unul sau mai multe lanțuri optime, depinzând de configurația grafului.

Cerință:

Fiind dat un graf neorientat conex cu N vârfuri etichetate cu numerele de ordine $1, 2, \dots, N$ și două vârfuri ale sale notate X și Y ($1 \leq X, Y \leq N, X \neq Y$), se cere să scrieți un program care determină vârfurile care aparțin **tuturor** lanțurilor optime dintre X și Y.

Date de intrare:

Fișierul **graf.in** conține

- pe prima linie patru numere naturale reprezentând: N (numărul de vârfuri ale grafului), M (numărul de muchii), X și Y (cu semnificația din enunț).
- pe următoarele M linii câte două numere naturale nenule A_i, B_i ($1 \leq A_i, B_i \leq N, A_i \neq B_i$, pentru $1 \leq i \leq M$) fiecare dintre aceste perechi de numere reprezentând câte o muchie din graf.

Date de ieșire:

Fișierul **graf.out** va conține

- pe prima linie, numărul de vârfuri comune **tuturor** lanțurilor optime dintre X și Y;
- pe a doua linie, numerele corespunzătoare etichetelor acestor vârfuri, dispuse **în ordine crescătoare**; între două numere consecutive de pe această linie se va afla câte un spațiu.

Restricții:

- $2 \leq N \leq 7500; 1 \leq M \leq 14000$
- pentru 50% din teste $N \leq 200$

Exemple:

graf.in	graf.out	graf.in	graf.out
6 7 1 4	3	3 2 1 3	2
1 2	1 4 5	1 2	1 3
1 3		3 1	
1 6			
2 5			
3 5			
5 6			
5 4			

Timp maxim de rulare/test: 1 secundă