

Problema Virus

Fișier de intrare: **virus.in**
Fișier de ieșire: **virus.out**

Limită de timp: 0,4 secunde
Limită de memorie: 256 MB

O rețea de calculatoare este formată din N^2 calculatoare așezate într-un pătrat cu N linii și N coloane. Inițial toate calculatoarele sînt pornite. Fiecare calculator este conectat prin cabluri la vecinii săi din cele patru direcții, iar oricare două calculatoare pot comunica dacă între ele există o cale formată din calculatoare pornite și conectate prin cabluri. Un virus straniu începe să infecteze această rețea. Fiecare pas al infecției oprește calculatoarele aflate pe un dreptunghi (nu și pe cele din interiorul dreptunghiului). Dreptunghiurile de la oricare doi pași **nu se intersectează și nu coincid**, dar pot fi incluse unul într-altul.

Administratorul de sistem dorește să știe, pe parcursul infecției, dacă diverse perechi de calculatoare mai pot comunica. Concret, se dau Q operații de două tipuri posibile, în ordine cronologică:

- 1 L_1 C_1 L_2 C_2 : virusul oprește calculatoarele de pe dreptunghiul $(L_1, C_1) - (L_2, C_2)$.
- 2 L_1 C_1 L_2 C_2 : administratorul se întreabă dacă cele două calculatoare de la (L_1, C_1) și (L_2, C_2) **sînt pornite și pot comunica între ele**.

Se cere să afișați răspunsurile la toate operațiile de tipul 2.

Date de intrare

Prima linie conține întregii N și Q , separați prin spații. Fiecare dintre următoarele Q linii conține cinci întregi T , L_1 , C_1 , L_2 și C_2 care codifică o operație.

Date de ieșire

Pentru fiecare operație de tip 2, în aceeași ordine ca la intrare, afișați cîte o linie cu textul **DA** dacă cele două calculatoare **sînt pornite și pot comunica** sau **NU** în caz contrar.

Restricții

- $3 \leq N \leq 2.000$
- $1 \leq Q \leq 100.000$
- $1 \leq L_1, L_2, C_1, C_2 \leq N$
- Pentru operațiile de tip 1, $L_1 < L_2$ și $C_1 < C_2$.
- Pentru operațiile de tip 2, (L_1, C_1) și (L_2, C_2) sînt distincte.

#	Puncte	Restricții
1	30	$N \leq 100$, $Q \leq 1.000$
2	70	Fără restricții suplimentare.

Exemplu

virus.in	virus.out	configurație finală
10 8	DA	* * * * * * * * *
2 3 6 8 2	DA	*
1 2 2 8 7	NU	* . * * * * . . * .
2 5 5 4 3	NU	* . * * .
2 5 5 4 1	NU	* . * . * . . . * .
1 4 4 6 6		* . *
2 5 5 4 3		* . * * * * . * * *
1 2 8 6 10		* * * *
2 6 9 7 9		* * * * * * * * *
		* * * * * * * * *

Explicații

Inițial toate calculatoarele sînt pornite, deci (3,6) și (8,2) pot comunica. După oprirea dreptunghiului (2,2)-(8,7), calculatoarele (5,5) și (4,3) se află în interiorul dreptunghiului și pot comunica, dar (5,5) și (4,1) sînt separate de dreptunghi și nu pot comunica. După oprirea dreptunghiului (4,4)-(6,6), calculatoarele (5,5) și (4,3) nu mai pot comunica. După oprirea dreptunghiului (2,8)-(6,10), calculatorul (6,9) **este oprit**, deci implicit nu poate comunica cu (7,9). Configurația finală este descrisă pe ultima coloană (calculatoarele pornite sînt marcate cu *, cele oprite cu punct).