

## Problem Capybara. Zip-line

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **3 seconds**  
Memory limit:         **256 megabytes**

Vasya s-a hotarat sa construiasca o tiroliana in copacii din padurea din apropiere. El vrea ca traseul sa fie cat mai lung posibil dar nu isi mai aminteste exact inaltimele tuturor copacilor din padure. El este sigur ca isi aminteste corect inaltimele tuturor copacilor exceptand, posibil, unul dintre ei.

Se stie ca padurea este formata din  $n$  copaci asezati intr-un rand numerotati de la stanga la dreapta cu numere naturale de la 1 la  $n$ . Dupa Vasya, inaltimea pomului  $i$  este egala cu  $h_i$ . Tiroliana de lungime  $k$  trebuie agatata peste  $k$  ( $1 \leq k \leq n$ ) copaci  $i_1, i_2, \dots, i_k$  ( $i_1 < i_2 < \dots < i_k$ ) astfel incat inaltimele lor formeaza o secventa strict crescatoare, mai exact  $h_{i_1} < h_{i_2} < \dots < h_{i_k}$ .

Petya a fost impreuna cu Vasya in padure si acum are  $q$  presupuneri despre greseala in secventa lui Vasya  $h_i$ . Presupunerea cu numarul  $i$  este formata din doua numere intregi  $a_i$  si  $b_i$  insemnand, dupa Petya, ca inaltimea copacului numerotat cu  $a_i$  este de fapt egala cu  $b_i$ . Presupunerile lui Petya sunt **independente** intre ele.

Aflati care este lungimea maxima a tirolienei care poate fi construita peste copaci pentru fiecare dintre cele  $q$  presupuneri.

In probolema aceasta lungiea tirolienei este egala cu numarul copacilor peste care este agatata.

### Input

Prima linie din input contine doua numere intregi  $n$  si  $m$  ( $1 \leq n, m \leq 400\,000$ ) — numarul copacilor din padure si numarul presupunerilor facute de Petya.

Urmatoarea linie contine  $n$  numere intregi  $h_i$  ( $1 \leq h_i \leq 10^9$ ) — inaltimele copacilor stiute de Vasya.

Fiecare dintre cele  $m$  linii urmatoare contine doua numere intregi  $a_i$  and  $b_i$  ( $1 \leq a_i \leq n, 1 \leq b_i \leq 10^9$ ).

### Output

Pentru fiecare presupunere a lui Petya afisati un singur numar intreg pe o linie separata, lungimea maxima a tirolienei.

### Examples

standard input	standard output
4 4	4
1 2 3 4	3
1 1	3
1 4	4
4 3	
4 5	
4 2	4
1 3 2 6	3
3 5	
2 4	

### Note

Pentru primul test prima presupunere a lui Petya este identica cu ce a spus Vasya. A doua presupunere considera ca inaltimele copacilor sunt de fapt (4, 2, 3, 4), in a treia avem inaltimele (1, 2, 3, 3) iar cea de-a patra duce la (1, 2, 3, 5).

## Scoring

Testele pentru aceasta problema sunt impartite in sapte grupe. Pentru fiecare dintre grupele 1–5 castigati puncte numai daca solutia trece toate testele din aceea grupa si toate testele din toate grupele **anterioare**.

In grupa 6 toate testele sunt notate independent si fiecare dintre ele valoreaza 1 punct. Solutia este evaluata pentru aceasta grupa numai daca toate testele din toate grupele anterioare au fost trecute.

**Evaluare offline** inseamna ca solutia va fi punctata dupa incheierea concursului.

Grupa	Teste	Puncte	Restrictii			Comment
			$n$	$m$	$h_i, b_i$	
0	1 – 2	0	–	–	–	Sample tests.
1	3 – 32	10	$n \leq 15$	$m \leq 15$	$h_i, b_i \leq 100$	
2	33 – 51	10	$n \leq 500$	$m \leq 500$	$h_i, b_i \leq 500$	
3	52 – 70	20	$n \leq 2000$	$m \leq 3000$	$h_i, b_i \leq 100\,000$	
4	71 – 89	20	$n \leq 10\,000$	$m \leq 20\,000$	$h_i, b_i \leq 100\,000$	
5	–	20	$n \leq 75\,000$	$m \leq 75\,000$	$h_i \leq 10^9$	<b>Evaluare offline</b>
6	–	20	$n \leq 400\,000$	$m \leq 400\,000$	$h_i \leq 10^9$	<b>Evaluare offline</b>

## Problem Echidna. Watchmen

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Watchmen sunt in pericol iar Doctor Mahhattan impreuna cu prietenul sau Daniel Dreiberg trebuie sa ii avertizeze cat mai repede. In plan se gasesc  $n$  watchmen, al  $i$ -lea watchman se afla in punctul  $(x_i, y_i)$ .

Ei trebuie sa puna la punct un plan, dar sunt cateva probleme. Dupa cum stiti, Doctor Manhattan considera distanta dintre watchmen  $i$  si  $j$  to be  $|x_i - x_j| + |y_i - y_j|$ . Daniel, ca orice persoana normala, calculeaza distanta folosind formula  $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ .

Succesul operatiunii depinde de cate perechi  $(i, j)$  ( $1 \leq i < j \leq n$ ) exista, astfel incat distanta dintre watchman  $i$  si watchmen  $j$  calculata de Doctor Manhattan este egala cu distanta dintre ei calculata de Daniel. Tu trebuie sa calculezi numarul perechilor de tipul acesta.

### Input

Prima linie din input contine un singur numar intreg  $n$  ( $1 \leq n \leq 200\,000$ ) — numarul de watchmen.

Fiecare dintre urmatoarele  $n$  linii contine doua numere intregi  $x_i$  and  $y_i$  ( $|x_i|, |y_i| \leq 10^9$ ).

### Output

Afisati numarul de perechi de watchmen, astfel incat distanta dintre ei calculata de Doctor Manhattan este egala cu distanta calculata de catre Daniel.

### Examples

standard input	standard output
3 1 1 7 5 1 5	2
6 0 0 0 1 0 2 -1 1 0 1 1 1	11

### Note

In primul exemplu, distanta dintre watchman 1 si watchman 2 este egala cu  $|1 - 7| + |1 - 5| = 10$  pentru Doctor Manhattan si  $\sqrt{(1 - 7)^2 + (1 - 5)^2} = 2 \cdot \sqrt{13}$  pentru Daniel.

Pentru perechile  $(1, 1)$ ,  $(1, 5)$  si  $(7, 5)$ ,  $(1, 5)$  Doctor Manhattan si Daniel vor calcula aceleasi distante.

### Scoring

Testele pentru aceasta problema sunt impartite in trei grupe. Pentru fiecare grupa veti primi puncte numai daca solutia trece toate testele din acea grupa si toate testele din toate grupele **anterioare**.

Grupa	Teste	Puncte	Restrictii suplimentare		Comentariu
			$n$	$x_i, y_i$	
0	1 – 2	0	–	–	Exemple
1	3 – 23	50	$1 \leq n \leq 1000$	$-10\,000 \leq x_i, y_i \leq 10\,000$	–
2	–	50	$1 \leq n \leq 200\,000$	–	–

## Problem Python. Clockwork Bomb

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2.5 seconds  
Memory limit:         256 megabytes

Numele meu este James deGriz, sunt cel mai inteligent hot si vanator de comori din intreaga galaxie. Exista carti scrise despre aventurile si cantece despre loviturile mele, cu toate astea tu ai reusit sa ma gasesti intr-un moment destul de dubios.

Am reusit sa ma ascund de camere, pacalesc toti paznicii si sa scap de numeroase capcane dar cand, in sfarsit, am ajuns in sipet si l-am deschis am pornit din greseala o bomba cu ceas! Din fericire, am mai intalnit astfel de bombe inainte si stiu ca mecanismul poate fi oprit conectand cu fire anumite contacte de pe panoul de control.

Vad  $n$  contacte unite de  $n - 1$  cabluri. Contactele sunt numerotate cu numere intregi de la 1 la  $n$ . Bomba are un mecanism de siguranta care asigura urmatoare conditie: daca exista  $k \geq 2$  contacte  $c_1, c_2, \dots, c_k$  care formeaza un circuit, i. e. adica exista  $k$  cabluri **distincte** intre contactele  $c_1$  si  $c_2$ ,  $c_2$  si  $c_3$ ,  $\dots$ ,  $c_k$  si  $c_1$ , atunci bomba explodeaza instant si povestea mea ia sfarsit aici. Se considera ca daca doua contacte sunt conectate cu cel putin doua cabluri atunci ele formeaza un circuit de lungime 2. De asemenea este interzis a conecta un contact cu el insusi folosind un cablu.

Pe de alta parte, daca deconectez mai mult de un cablu (i. e. la un moment dat nu exista mai mult de  $n - 2$  fire in schema) atunci alt mecanism de securanta se activeaza si bomba explodeaza. Deci, singurul lucru pe care il pot face este sa deconectez un fir si apoi sa il conectez intr-un loc nou avand grija sa nu apara niciun circuit.

Stiu in ce ordine sa pun cablurile pentru a dezamorsa bomba dar timpul meu se scurge! Ajuta-ma sa scap: gaseste secventa de operatii constand in deconectarea unui cablu si apoi conectarea sa in alt loc astfel incat bomba va fi dezamorsata.

### Input

Prima linie din input contine  $n$  ( $2 \leq n \leq 500\,000$ ), numarul de contacte.

Fiecare dintre urmatoarele  $n - 1$  linii contine doua numere intregi  $x_i$  si  $y_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ) desemnand contactele unite de cablul  $i$ .

Urmatoarele  $n - 1$  linii contin descrierea schemei cautate in acelasi format.

### Output

Prima linie trebuie sa contina  $k$  ( $k \geq 0$ ) — numarul minim de mutari, formate din deconectarea si reconectarea unui cablu, pentru a dezamorsa bomba.

Pe fiecare din urmatoarele  $k$  linii afisati patru numere intregi  $a_i, b_i, c_i, d_i$  cu semnificatia ca la pasul al  $i$ -lea ar trebui sa deconectez cablul care uneste contactele  $a_i$  and  $b_i$  si sa il folosesc pentru a uni contactele  $c_i$  and  $d_i$ . Bineinteles, cablul care uneste contactele  $a_i$  and  $b_i$  trebuie sa fie prezent in schema.

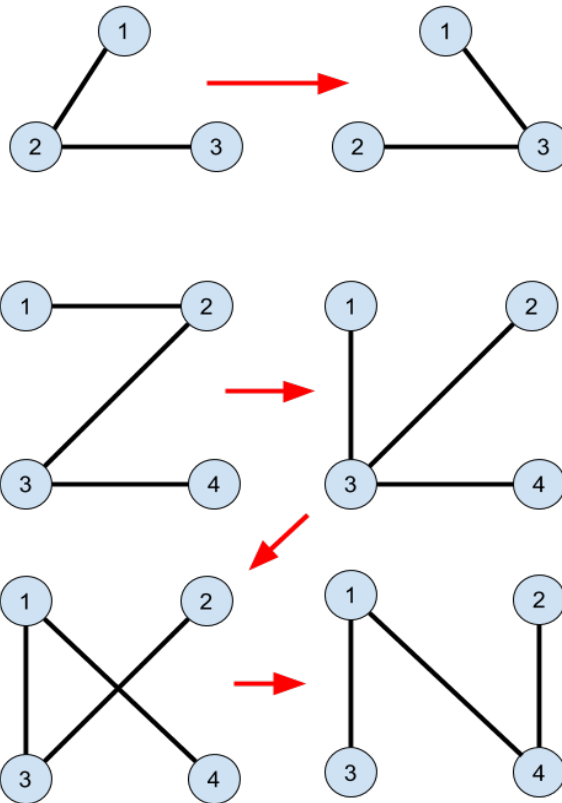
Daca nu exista o secventa corecta pentru a transforma schema initiala in schema cautata afisati -1.

## Examples

standard input	standard output
3 1 2 2 3 1 3 3 2	1 1 2 1 3
4 1 2 2 3 3 4 2 4 4 1 1 3	3 1 2 1 3 4 3 4 1 2 3 2 4

## Note

Detalii exemplu:



## Scoring

Testele pentru aceasta problema sunt impartite int trei grupe. Pentru fiecare grupa castiga puncte numai daca solutia trece toate testele din grupa respectiva si toate testele din toate grupele **anterioare**.

X Moscow Open Olympiad in Informatics (Romanian version)  
Moscow, March 7th, 2016

---

Grupa	Teste	Puncte	Restrictii suplimentare	Comment
			$n$	
0	1 – 2	0	—	Exemple
1	3 – 18	20	$n \leq 50$	Se garanteaza ca raspunsul exista si nu necesita mai mult decat o operatie
2	19 – 37	20	$n \leq 50$	
3	38 – 53	20	$n \leq 5000$	
4	54 – 69	20	$n \leq 100\,000$	
5	–	20	$n \leq 500\,000$	<b>Evaluare offline</b>

## Problem Unicorn. Table Compression

Input file:            standard input  
Output file:           standard output  
Time limit:            4 seconds  
Memory limit:         256 megabytes

Micutul Petya este acum atras de algoritmiile de compresie a datelor. El a studiat deja algoritmiile *gz*, *bz*, *zip* si multi altii. Inspirat de noile sale cunostinte, Petya dezvoltă un nou algoritm de compresie pe care vrea sa il numeasca *dis*.

Petya a decis sa comprime matrici. El primeste o matrice  $a$  formata din  $n$  linii si  $m$  coloane ce contine numere intregi pozitive si vrea sa construiasca o matrice  $a'$  formata din numere intregi pozitive astfel incat ordinea relativa a elementelor de pe fiecare linie si fiecare coloana ramane aceeasi. Mai exact, daca intr-o linie  $i$  din matricea initiala  $a_{i,j} < a_{i,k}$ , atunci in matricea rezultata  $a'_{i,j} < a'_{i,k}$ , daca  $a_{i,j} = a_{i,k}$  atunci  $a'_{i,j} = a'_{i,k}$ . Similar, daca intr-o coloana  $j$  din matricea initiala  $a_{i,j} < a_{p,j}$  atunci in matricea comprimata  $a'_{i,j} < a'_{p,j}$ , daca  $a_{i,j} = a_{p,j}$  atunci  $a'_{i,j} = a'_{p,j}$ .

Pentru ca valorile mari necesita mai mult spatiu de stocare, valoarea maxima din  $a'$  trebuie sa fie cat mai mica posibil.

Petya le are cu teoria, totusi, are nevoie de ajutorul tau ca sa implementeze algoritmul.

### Input

Prima linie din input contine doua numere intregi  $n$  and  $m$  ( $1 \leq n, m$  and  $n \cdot m \leq 1\,000\,000$ ), numarul de linii si de coloane ale tabloului initial.

Fiecare dintre cele  $n$  linii contine  $m$  numere intregi  $a_{i,j}$  ( $1 \leq a_{i,j} \leq 10^9$ ) reprezentand valorile din matrice.

### Output

Afisati matricea comprimata sub forma a  $n$  linii, fiecare continand  $m$  numere intregi.

Daca exista mai multe raspunsuri astfel incat numarul maxim din matricea comprimata este minim, puteti afisa pe oricare dintre ele.

### Examples

standard input	standard output
2 2 1 2 3 4	1 2 2 3
4 3 20 10 30 50 40 30 50 60 70 90 80 70	2 1 3 5 4 3 5 6 7 9 8 7

### Note

In primul exemplu, in ciuda faptului ca  $a_{1,2} \neq a_{2,1}$ , ele nu sunt aflate pe aceeasi linie sau coloana asa ca pot fi egale dupa compresie.

### Scoring

Testele pentru aceasta problema sunt impartite in sase grupe. Pentru fiecare grupa veti castiga puncte numai daca solutia trece toate testele din acea grupa si toate testele din cateva dintre grupele anterioare. **Evaluare offline** inseamna ca solutia va fi evaluata pentru testele din grupa aceea numai dupa incheierea concursului.



Grupa	Teste	Puncte	Restrictii suplimentare			Grupe necesare	Comment
			$n$	$m$	$a_{i,j}$		
0	1 – 2	0	–			–	Exemplu
1	3 – 19	10	$n \leq 1000$	$m = 1$	–	–	
2	20 – 39	15	$n, m \leq 100$	Toate $a_{i,j}$ sunt diferite			
3	40 – 74	15	$n, m \leq 100$	–		0, 2	
4	75 – 84	15	$n, m \leq 400$	Toate $a_{i,j}$ sunt diferite		2	
5	85 – 102	15	$n, m \leq 400$	–		0, 2, 3, 4	
6	103 – 112	15	–	Toate $a_{i,j}$ sunt diferite		2, 4	
7	–	15	Nicio restrictie suplimentara			0 – 6	<b>Evaluare offline</b>