

## Problem Сапувара. Лифт

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

Като се разхождал неотдавна из гората, Вася решил да построи дървен лифт. Той иска, лифтът да бъде колкото може по-дълъг, но за съжаление трудно помни височината на дърветата в гората. За щастие, той е уверен, че правилно помни височината на всички дървета, с изключение на едно от тях.

Известно е, че гората се състои от  $n$  дървета, растящи в редици и номерирани отляво надясно с числата от 1 до  $n$ , височината на  $i$ -ото дърво по спомените на Вася е равна на  $h_i$ . Лифт с дължина  $k$  трябва да се опира на  $k$  ( $1 \leq k \leq n$ ) дървета  $i_1, i_2, \dots, i_k$  ( $i_1 < i_2 < \dots < i_k$ ), такива, че тяхната височина нараства, т. е.  $h_{i_1} < h_{i_2} < \dots < h_{i_k}$ .

Петя също ходил в гората, и той има  $q$  предположения за това, къде именно греша Вася. Неговото  $i$ -то предположение се задава с числата  $a_i$  и  $b_i$ , означаващи, че по мнението на Петя, височината на дървото с номер  $a_i$  всъщност е равна на  $b_i$ . Обърнете внимание, че Петините предположения са **независими** едно от друго.

Вашата задача се състои в това, за всяко от предположенията на Петя, да се намери максималната дължина на лифта, който може да се построи, опирайки се на тези дървета.

Да отбележим, че в рамките на дадената задача, за дължина на лифта, Вася приема броя опорни дървета в него.

### Input

Първият ред на входните данни съдържа две числа  $n$  и  $m$  ( $1 \leq n, m \leq 400\,000$ ) — броя дървета в гората и броя на предположенията на Петя.

На следващия ред са зададени  $n$  цели числа  $h_i$  ( $1 \leq h_i \leq 10^9$ ) — височината на дърветата по предположението на Вася.

Всеки от следващите  $m$  реда съдържа по две числа  $a_i$  и  $b_i$  ( $1 \leq a_i \leq n, 1 \leq b_i \leq 10^9$ ).

### Output

За всяко предположение на Петя изведете на отделен ред едно число — максималната дължина на лифта.

### 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

Да разгледаме първия пример. Първото Петино предположение съвпада с предположението на Вася.

По второто предположение, височините на дърветата са били (4, 2, 3, 4), по третото (1, 2, 3, 3), а по четвъртото (1, 2, 3, 5).

## Scoring

Тестовите към тази задача се състоят от седем групи. Точките за групите от 0 до 5 се дават само при преминаване на всички тестове в групата и всички тестове от **предходните** групи.

Тестовите в група 6 се оценяват независимо и струват 1 точка всеки. Решението се тества на тестовите от тази група само при преминаване на всички тестове от **предходните** групи.

Група	Тестове	Точки	Ограничения			Коментарии
			$n$	$m$	$h_i, b_i$	
0	1 – 2	0	–	–	–	Тестовите от условието.
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>Offline-проверка</b>
6	–	20	$n \leq 400\,000$	$m \leq 400\,000$	$h_i \leq 10^9$	<b>Offline-проверка</b>

## Problem Echidna. Пазители

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

Пазителите са в опасност, и Доктор Манхатън и неговият приятел Даниел Драйберг трябва спешно да ги предупредят. В отбора на пазителите има точно  $n$  човека,  $i$ -тия от които се намира в точка от равнината с координати  $(x_i, y_i)$ .

Настъпило е време за съставяне на план, но за това има няколко пречки. Както на всички е известно, доктор Манхатън изчислява разстоянието между двама пазители  $i$  и  $j$  по формулата  $|x_i - x_j| + |y_i - y_j|$ . Даниел пък, като обикновен човек, приема, че разстоянието е равно на  $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ .

Сега успехът на операцията зависи от това, колко двойки  $(i, j)$  ( $1 \leq i < j \leq n$ ) има, такива че разстояние между пазителя  $i$  и пазителя  $j$ , изчислено от Доктор Манхатън, е равно на разстоянието между тях, изчислено от Даниел. Да изчислите тази величина са помолили именно вас.

### Input

На първия ред на входните данни е записано число  $n$  ( $1 \leq n \leq 200\,000$ ) — броя на пазителите.

На всеки от следващите  $n$  реда са записани две цели числа  $x_i$  и  $y_i$  ( $|x_i|, |y_i| \leq 10^9$ )

### Output

Изведете броя на двойките пазители, такива че разстоянието между тях, изчислено от доктор Манхатън, е равно на разстоянието, изчислено от Даниел.

### 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

В първия пример разстоянието между пазител 1 и пазител 2 е равно на  $|1 - 7| + |1 - 5| = 10$  според доктор Манхатън и  $\sqrt{(1 - 7)^2 + (1 - 5)^2} = 2 \cdot \sqrt{13}$  според Даниел. За двойките  $(1, 1)$ ,  $(1, 5)$  и  $(7, 5)$ ,  $(1, 5)$  разстоянията, изчислени от доктор Манхатън и Даниел, съвпадат.

### Scoring

Тестовите към тази задача се състоят от три групи. Точките за всяка група се дават само при преминаване на всички тестове в групата и всички тестове от **предходните** групи.

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

---

Група	Тестове	точки	Допълнителни ограничения		Коментари
			$n$	$x_i, y_i$	
0	1 – 2	0	–	–	Тестовете от условието
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. Часовников механизъм

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

Казвам се Джим ди Гриз и съм най-ловкия мошеник и авантюрист в цялата Галактика. По мотиви от моите похождения са написани множество книги, а грабежите, извършени от мен, нямат брой. Обаче, вие можете да ме спипате в много пикантна ситуация.

Не намирайки пред себе си камери, надхитрявайки десетки охранители и заобикаляйки множество клопки, аз мога да достигна до желаня сандък със съкровища. Повдигайки капака му, аз съм активирал бомба с часовников механизъм, който вече отчита секундите на неизбежния взрив! За щастие, вече съм се сблъсквал с бомби от подобен модел и знам, че часовниковият механизъм може да бъде спрян, свързвайки по определен начин контактите в панела за управление на бомбата с жички.

Пред мен има  $n$  контакти, свързани с  $n - 1$  жички. Контактите са номерирани с целите числа от 1 до  $n$ . Бомбата е устроена така, че ако някой набор от  $k \geq 2$  контакти  $c_1, c_2, \dots, c_k$  е свързан в цикъл, т. е. между двойките контакти  $c_1$  и  $c_2$ ,  $c_2$  и  $c_3$ ,  $\dots$ ,  $c_k$  и  $c_1$  има  $k$  различни жички, то сработва проверка за безопасност, и зарядът мигновено се взривява, без да остави от неудачния взломажия ни следа. В това число, ако два контакта са свързани с повече от една жичка, от тях се образува цикъл с дължина 2, и бомбата също се взривява. Свързането на контакт със себе си с една жичка също се забранява.

От друга страна, ако отрежа едновременно няколко жички (с други думи, в някакъв момент от време ще бъдат включени  $n - 2$  жички), то сработва друга проверка за безопасност, която ще доведе до същия плачевен резултат. По този начин, всичко, което ми остава, е последователно да издърпвам жички и да ги слагам на ново място, за да не се образува цикъл, свързващ контактите.

Аз знам, как трябва да се разположат жичките, за да се спре часовниковият механизъм. Но ми остава все по-малко и по-малко време да го направя! Помогнете ми да да се измъкна от положението, като намерите най-кратката последователност от операции, всяка, от които представлява последователно изключване на определена жичка и включването и на ново място и, която подрежда жичките по искания начин.

### Input

На първия ред на входните данни се намира числото  $n$  ( $2 \leq n \leq 500\,000$ ).

По нататък в  $n - 1$  реда се намират  $n - 1$  двойки числа  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ), означаващи контакти, свързани с последователни жички в даден момент от време.

В следващите  $n - 1$  реда в аналогичен формат се задава схемата на включване на жичките, спираща часовниковия механизъм.

### Output

На първия ред изведете числото  $k$  ( $k \geq 0$ ), минималния брой жички, които аз трябва да издърпам и да свържа отново.

На следващите  $k$  реда изведете  $k$  четворки числа  $a_i, b_i, c_i, d_i$ , означаващи, че на  $i$ -тата стъпка аз трябва да отсъединя жичката, свързваща контактите  $a_i$  и  $b_i$ , и да свържа с нея контактите  $c_i$  и  $d_i$ . Разбира се, към този момент от времето жичката между контактите трябва да присъства в схемата.

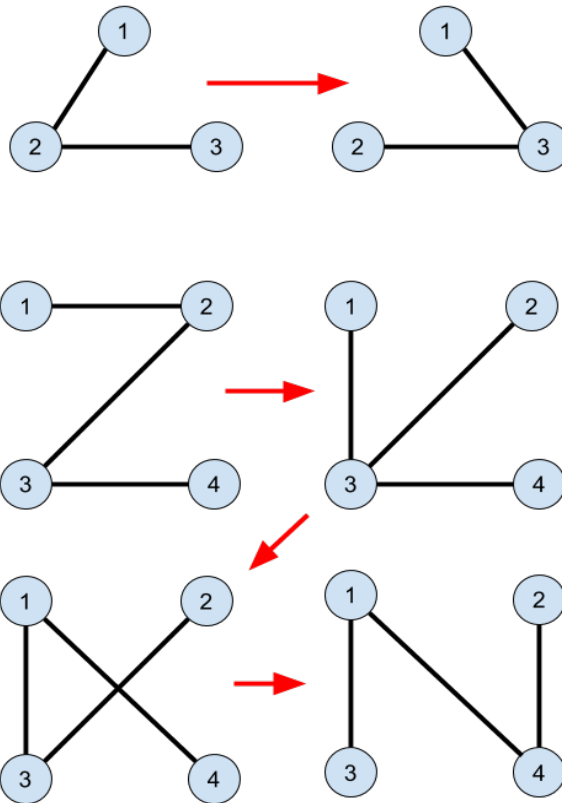
Ако търсената последователност от операции не съществува, изведете едно число -1.

## 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

Картинка с пояснение към тестовете от условието:



## Scoring

Тестовете към тази задача се състоят от шест групи. Точките за всяка група се дават само при преминаване на всички тестове в групата и всички тестове от **предходните** групи с изключение, евентуално, тестовете от условието.

Група	Тестове	Точки	Ограничения	Коментари
			$n$	
0	1 – 2	0	—	Тестове от условието
1	3 – 18	20	$n \leq 50$	Гарантирано е, че отговор съществува и изисква не повече от една операция
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>Offline-проверка</b>

## Problem Unicorn. Компресиране на таблици

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

Малкия Петя се увлича от алгоритми за компресиране на данни. Той вече е изучил форматите *gz*, *bz*, *zip* и няколко други. Въодушевен от новите си знания, Петя е решил да разработи свой формат на компресия и да го нарече *dis*.

Петя решил да компресира таблици. Той има таблица от  $n$  реда и  $m$  колони, запълнена с цели положителни числа. Той иска да замени стойностите на елементите на таблицата с цели положителни числа така, че редът на елементите във всеки ред и всяка колона да не се измени. С други думи, ако в някои от редовете на началната таблица  $a_{i,j} < a_{i,k}$ , то и в компресираната таблица  $a'_{i,j} < a'_{i,k}$ , и ако  $a_{i,j} = a_{i,k}$ , то  $a'_{i,j} = a'_{i,k}$ . Аналогично ако в някоя колона на началната таблица  $a_{i,j} < a_{p,j}$ , то и в компресираната таблица  $a'_{i,j} < a'_{p,j}$ , и ако  $a_{i,j} = a_{p,j}$ , то  $a'_{i,j} = a'_{p,j}$ .

Тъй като големите стойности изискват големи места за съхранение, максималната стойност на елемента на получената матрица трябва да бъде колкото може по-малка.

На теория Петя е професионалист, но не обича да пише код. Помогнете му да реализира своя формат на компресия *dis*.

### Input

На първия ред на входните данни са зададени две числа  $n$  и  $m$  ( $1 \leq n, m$  и  $n \cdot m \leq 1\,000\,000$ ) — броя редове и броя колони на таблицата.

На следващите  $n$  реда са зададени по  $m$  цели числа  $a_{i,j}$  ( $1 \leq a_{i,j} \leq 10^9$ ) — стойностите на елементите на таблицата.

### Output

Изведете компресираната таблица:  $n$  реда, съдържащи по  $m$  числа.

Ако съществуват няколко отговора, минимизиращи максималното число, се изведете кой да е от тях.

В първият пример  $a_{1,2} \neq a_{2,1}$ , но, тъй като те не са разположени в един ред или в една колона, при компресирането може да се направят равни.

### 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

### Scoring

Тестовите към тази задача се състоят от шест групи. Точките за всяка група се дават само при преминаване на всички тестове в групата и всички тестове от **предходните** групи. **Offline-проверка**



означава, че резултатите от тестването на вашето решение на дадената група ще станат достъпни едва след завършването на състезанието.

Група	Тестове	Точки	Допълнителни ограничения			Необх. групи	Коментари
			$n$	$m$	$a_{i,j}$		
0	1 – 2	0	–			–	Тестовете от услов
1	3 – 19	10	$n \leq 1000$	$m = 1$	–	–	
2	20 – 39	15	$n, m \leq 100$		Всички $a_{i,j}$ са различни	–	
3	40 – 74	15	$n, m \leq 100$		–	0, 2	
4	75 – 84	15	$n, m \leq 400$		Всички $a_{i,j}$ са различни	2	
5	85 – 102	15	$n, m \leq 400$		–	0, 2, 3, 4	
6	103 – 112	15	–		Всички $a_{i,j}$ са различни	2, 4	
7	–	15	Допълнителни ограничения няма			0 – 6	<b>Offline-провер</b>