

# Plemiona (plemiona)

Memory limit: 32 MB

Time limit: 7.00 s

Po dwóch stronach rzeki żyją dwa plemiona, a każde z nich posługuje się swoim lokalnym dialektem. Wodzowie obu plemion zgodnie chcieliby wybudować na rzece most, który połączy ich plemiona (ta zgodność wymagała wielu dni rozmów “na migi” prowadzonych z dwóch różnych brzegów rzeki). Powstał jednak poważny problem – wspólne wybudowanie mostu wymaga stabilnej i sprawnej komunikacji między plemionami.

Szaman Twojego plemienia, po wielu dniach wyteżonej obserwacji, rozpracował już różnicę między dialektami. Zdanie z jednego dialektu można przetłumaczyć na drugi dialekt, odwracając kolejność słów w tym zdaniu. Przykładowo zdanie `sele sinazo iinkuni zebhulorho` z jednego dialektu tłumaczy się do `zebhulorho iinkuni sinazo sele` w drugim dialekcie.

Wódz Twojego plemienia powierzył Ci zadanie przetłumaczenia jego szczegółowej koncepcji mostu na dialekt drugiego plemienia. Pomóż swojemu Wodzowi, a otrzymasz przywilej przecięcia wstęgi na nowopowstałym moście!

## Wejście

W pierwszym wierszu wejścia znajduje się jedna dodatnia liczba całkowita  $N$ , określająca liczbę zdań, które trzeba przetłumaczyć.

Po pierwszym wierszu wejścia następuje  $N$  opisów zdań.

Każdy opis zdania składa się z dwóch wierszy: w pierwszym z nich znajduje się jedna dodatnia liczba całkowita  $L_i$ , określająca liczbę słów w zdaniu, a w drugim z nich znajduje się  $L_i$  słów pooddzielanych pojedynczymi odstępami, które składają się na jedno zdanie.

## Wyjście

Dla każdego podanego zdania należy w osobnym wierszu wypisać jego tłumaczenie na drugi dialekt.

## Ograniczenia

Słowa złożone są z małych liter alfabetu łacińskiego, sumaryczna długość wszystkich podanych słów nie przekracza 500 000,  $1 \leq N, L_i \leq 500\,000$ .

## Przykład

### Input

```
2
1
siyaqala
4
sele sinazo iinkuni zebhulorho
```

### Output

```
siyaqala
zebhulorho iinkuni sinazo sele
```

### Input

```
2
9
borokho bo lokela ho ba boholo joalo ka mammoth
6
le bophara bo lekanang le matsale
```

### Output

```
mammoth ka joalo boholo ba ho lokela bo borokho
matsale le lekanang bo bophara le
```

**Input**

3

2

deras sungai

6

kita pondok sekuat mestilah itu jambatan

4

terjemahan dengan berjaya semoga

**Output**

sungai deras

jambatan itu mestilah sekuat pondok kita

semoga berjaya dengan terjemahan

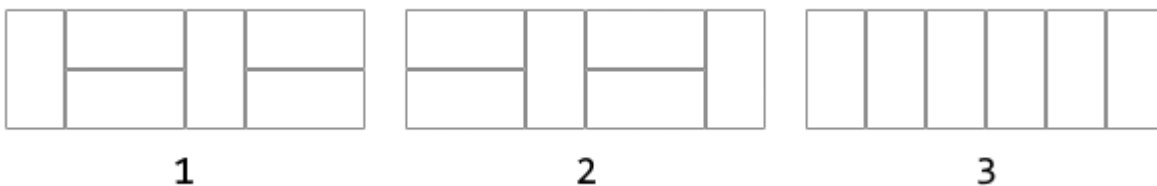
# Zamek (zamek)

Memory limit: 128 MB

Time limit: 1.00 s

Zapewne pierwszą rzeczą, jaka kojarzy się ludziom ze średniowieczem, jest stojący wśród lasów (bądź innych terenów zielonych) zamek. W końcu przecież, w zasadzie przez całą tę epokę historyczną, ludzie tylko i wyłącznie budowali zamki. Tak też będzie w tym zadaniu.

Pewien władca średniowiecznego imperium zwany Karolem Wspaniałym postanowił wybudować sobie zamek absolutnie niemożliwy do zdobycia; twierdzę, która przetrwa absolutnie każdy najazd. Oczywiście twierdza taka musi mieć solidnie zbudowane mury. Władca Karol wspaniały postanowił, że jego ściany będą miały nie jeden metr grubości, a dwa. Precyzyjniej mówiąc, mury zamkowe buduje się z bloków o wymiarach  $2 \times 1 \times 1$ . Władca Karol Wspaniały wybudował już pierwszą warstwę muru w kształcie prostopadłościanu o wymiarach  $N \times 1 \times 2$ . Oczywiście brakuje jeszcze drugiej warstwy o dokładnie takich samych wymiarach, w końcu mur miał mieć dwa metry grubości. Jednak tutaj pojawia się pewien kłopot. Biorąc pod uwagę właściwości obronne muru, żaden z bloków z warstwy pierwszej nie może pokrywać się z żadnym z bloków z warstwy drugiej.



Przykładowo, jeśli w widoku od przodu warstwa pierwsza wygląda jak na rysunku pierwszym, to układ na rysunku drugim byłby poprawny – żaden blok z warstwy pierwszej nie pokrywa się z żadnym blokiem z warstwy drugiej. Natomiast układ na rysunku trzecim byłby niepoprawny, ponieważ bloki pionowe pokrywałyby się z blokami pionowymi z układu z rysunku pierwszego.

Twoim zadaniem będzie więc napisanie programu, który wczyta opis pierwszej warstwy muru i wyznaczy poprawny układ bloków w drugiej warstwie, czyli taki, w którym żaden z bloków z pierwszej warstwy nie pokrywa się z żadnym z bloków z drugiej warstwy.

## Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba naturalna  $N$  oznaczająca długość pierwszej warstwy muru. W drugim wierszu znajduje się ciąg znaków  $S$ , składający się z liter H i V. Litera H odpowiada dwóm poziomym blokom, z których jeden postawiony jest na drugim, natomiast litera V odpowiada jednemu blokowi, ustawionemu pionowo. Przykładowo ciąg znaków VHVH reprezentuje układ z rysunku pierwszego, natomiast ciąg znaków VVVVVV reprezentuje układ z rysunku trzeciego.

## Wyjście

W pierwszym wierszu standardowego wyjścia powinno znaleźć się jedno słowo TAK lub NIE w zależności od tego, czy istnieje poprawny układ bloków w warstwie drugiej. Jeśli odpowiedź jest twierdząca, to w drugim wierszu wyjścia należy wypisać ciąg znaków reprezentujący poprawny układ warstwy drugiej, w postaci opisanej w sekcji wejście. Jeśli istnieje wiele poprawnych odpowiedzi, możesz wypisać dowolną z nich.

## Ograniczenia

$2 \leq N \leq 100\,000$ . Ciąg znaków  $S$  reprezentuje mur długości  $N$ .

## Przykład

Input	Output
6	TAK
VHVH	HVHV

**Input**

6

VVVVVV

**Output**

TAK

HHH

# Wyprawy morskie (wyprawy-morskie)

Memory limit: 512 MB Time limit: 3.00 s

Nastaly czasy różnorodnych wypraw morskich – wielkie odkrycia geograficzne, przygody, morskie potwory i piraci. Na świecie powstają wielkie imperia mające swoje terytoria we wszelkich zakątkach naszego globu. Zostałeś właśnie najważniejszym kartografem (obraźliwie zwanym mapoludem) jednego z największych imperiów na świecie. Twoim zadaniem będzie zaplanowanie odpowiednich handlowych szlaków morskich, które pozwolą skutecznie rozprowadzać różnorakie towary na obszar całego imperium. W końcu pracownicy bezpłatni sami nie dopłyną na plantacje, a egzotyczne owoce same nie wyrosną na stole władcy.

Widzisz przed sobą mapę całego świata, a na niej  $N$  zaznaczonych ważnych portów, numerowanych kolejnymi liczbami naturalnymi od 1 do  $N$ , a każdy z nich dostarcza reszcie świata ważne surowce. Ponadto na mapie zaznaczonych jest  $M$  dwukierunkowych tras morskich, a każda z tras morskich łączy ze sobą dwa różne porty. Twoim zadaniem jest zaplanowanie handlowych szlaków morskich.

Handlowy szlak morski to taki ciąg portów, że pomiędzy każdą parą portów sąsiednich w ciągu istnieje trasa morska je łącząca. W szczególności handlowy szlak morski może wielokrotnie przepływać przez ten sam port, jednak nigdy nie może przepływać dwukrotnie tą samą trasą morską.

Twoim zadaniem jest zaplanowanie minimalnej liczby handlowych szlaków morskich, które będą korzystały z każdej zaznaczonej na mapie trasy morskiej, jednak każda trasa morska może być używana przez dokładnie jeden handlowy szlak morski.

## Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby naturalne  $N$  i  $M$ , oddzielone pojedynczym odstępem i oznaczające odpowiednio liczbę portów i tras morskich pomiędzy nimi. W kolejnych  $M$  wierszach znajdują się po dwie liczby naturalne  $A_i$  i  $B_i$  oddzielone pojedynczym odstępem i oznaczające istnienie trasy morskiej pomiędzy portami o numerach  $A_i$  i  $B_i$ . Pomiedzy każdą parą portów istnieje co najwyżej jedna trasa morska.

## Wyjście

W pierwszym wierszu wyjścia należy wypisać jedną liczbę naturalną  $K$  – minimalną liczbę handlowych szlaków morskich, które spełnią założenia zadania. W kolejnych  $2 \cdot K$  wierszach powinny znaleźć się opisy proponowanych handlowych szlaków morskich.

Opis handlowego szlaku morskiego powinien składać się z dwóch wierszy. W pierwszym z nich powinna znajdować się jedna liczba naturalna  $X_i$  – liczba portów w proponowanym handlowym szlaku morskim. W drugim z nich powinno znaleźć się  $X_i$  liczb naturalnych pooddzielanych pojedynczymi odstępami i oznaczających numery kolejnych portów morskich w proponowanym handlowym szlaku morskim.

Jeśli istnieje wiele poprawnych odpowiedzi, możesz wypisać dowolną z nich.

## Ograniczenia

$2 \leq N \leq 100\,000$ ,  $2 \leq M \leq 200\,000$ ,  $1 \leq A_i, B_i \leq N$ ,  $A_i \neq B_i$ .

## Przykład

Input	Output
9 8	3
1 2	3
2 3	1 2 5
2 4	3
2 5	4 2 3
6 7	5
7 8	9 6 7 8 9
8 9	
9 6	

# Mozaika (mozaika)

Memory limit: 64 MB

Time limit: 10.00 s

Kilka lat temu zaczęła się budowa perły Cesarstwa – niemal doskonałej świątyni z olbrzymią kopułą. Budowa powoli dobiega już końca, jednak do pełnej doskonałości budowli brakuje tylko jednego: mozaiki na posadzce. By mozaika była dobra, musi być zbudowana z kwadratowych kamyczków tego samego rozmiaru, cała musi mieć kształt kwadratu i musi się składać z kamyczków w dokładnie dwóch kolorach.

Twoim zadaniem będzie kupienie na lokalnym targowisku materiałów na tę finalną mozaikę. W sekcji kamieniarskiej targowiska są tylko dwaj kupcy: Kwadratus i Liniusz. Kwadratus sprzedaje tylko białe kamyczki i za  $K$  talarów otrzymasz u niego  $K^2$  kamyczków. Liniusz natomiast sprzedaje droższe, połączane kamyczki i za  $K$  talarów możesz od niego dostać  $K$  kamyczków.

Targowisko jest już bliskie zamknięcia, kupcy chcą już zamykać stanowiska, więc każdemu z dwóch kupców możesz zapłacić tylko jedną monetą. Na ten zakup kwesor przyznał Ci  $N$  monet o nominałach  $A_1, A_2, \dots, A_N$ . Nominały w Cesarstwie nie są przypadkowe – dla wygody nominały nie mają dzielników pierwszych większych od 5.

Sytuacja staje się napięta, bo kupcy już zamykają stanowiska, mozaikę trzeba wykonać jak najszybciej, a możliwych zakupów jest dużo. Ale właściwie to jak dużo? Na ile sposobów możesz wręczyć Kwadratusowi i Liniuszowi dwie monety tak, by otrzymane od nich kamyczki mogły wszystkie złożyć się na dobrą mozaikę?

## Wejście

W pierwszym wierszu wejścia znajduje się jedna dodatnia liczba całkowita  $N$ , określająca liczbę Twoich monet. W drugim wierszu wejścia znajduje się  $N$  liczb całkowitych  $A_1, A_2, \dots, A_N$  pooddzielanych pojedynczymi odstępami i określających nominały kolejnych monet w talarach.

## Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna nieujemna liczba całkowita, określająca liczbę sposobów wręczenia kupcom monet tak, by z uzyskanych w ten sposób kamyczków dało się stworzyć dobrą mozaikę.

## Ograniczenia

$1 \leq N \leq 100\,000$ ,  $1 \leq A_i \leq 10^{18}$ , liczby  $A_i$  są postaci  $2^{n_i} 3^{m_i} 5^{k_i}$  dla nieujemnych i całkowitych  $n_i, m_i, k_i$ .

## Przykład

Input	Output	Explanation
5 1 1 16 3 5	3	Są 3 sposoby: Kwadratus dostaje monetę $A_1$ , Liniusz dostaje monetę $A_4$ i powstaje mozaika $2 \times 2$ ; Kwadratus dostaje monetę $A_2$ , Liniusz dostaje monetę $A_4$ i powstaje mozaika $2 \times 2$ ; Kwadratus dostaje monetę $A_4$ , Liniusz dostaje monetę $A_3$ i powstaje mozaika $5 \times 5$ .
4 2 3 5 5	2	Są 2 sposoby: Kwadratus dostaje monetę $A_1$ , Liniusz dostaje monetę $A_3$ i powstaje mozaika $3 \times 3$ ; Kwadratus dostaje monetę $A_1$ , Liniusz dostaje monetę $A_4$ i powstaje mozaika $3 \times 3$ .

**Input**

5  
3 3 5 3 12

**Output**

0

**Explanation**

Nie da się ułożyć dobrej mozaiki.

# Prawidłowa odbudowa (prawidlowa-odbudowa)

Memory limit: 128 MB

Time limit: 1.50 s

Stolica Cesarstwa spłonęła, więc niezwłocznie trzeba zaprojektować miasto na nowo i je odbudować. Pierwszy element projektu jest inicjatywą samego Cesarza – w punkcie centralnym miasta musi powstać pałac, ogromny i przestronny. Twoim zadaniem jest sprawdzenie i ewentualne poprawienie projektu jednej z ulic odchodzących od środka miasta.

Projekt ulicy to ciąg dodatnich liczb całkowitych oznaczających wysokości kolejnych budynków. By pałac był godzien Cesarza, w jego okolicy nie powinno być wysokich budynków, a najlepiej by było, gdyby najbliżej pałacu stały budynki najniższe, a najdalej najwyższe. Innymi słowy projekt ulicy jest dobry wtedy i tylko wtedy, gdy wysokości budynków są posortowane niemalejąco.

Lokalny artysta-architekt przygotował już swoją propozycję projektu, jednak istnieje szansa, że pracując pośpiesznie popełnił błąd i wysokości budynków nie są posortowane. By nie urazić uczuć artystycznych lokalnego artysty-architekta, możesz zamienić miejscami co najwyżej jedną parę budynków z projektu.

Czy projekt lokalnego artysty-architekta jest dobry lub czy da się go naprawić jedną zamianą?

## Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna dodatnia liczba całkowita  $N$ , określająca liczbę budynków w projekcie. W drugim wierszu wejścia znajduje się  $N$  dodatnich liczb całkowitych  $H_1, H_2, \dots, H_N$  pooddzielanych pojedynczymi odstępami i określających wysokości kolejnych budynków w projekcie.

## Wyjście

Jeżeli projekt jest dobry, to wypisz wyłącznie słowo TAK.

Jeżeli projekt nie jest dobry, ale można go naprawić jedną zamianą budynków, to w pierwszym wierszu wyjścia wypisz słowo NAPRAWA, a w drugim wierszu wyjścia dwie dodatnie liczby całkowite określające pozycje do zamienienia. Pierwsza z wypisanych liczb musi być mniejsza od drugiej.

Jeżeli projekt nie jest dobry i nie da się go naprawić jedną zamianą, to wypisz wyłącznie słowo NIE.

## Ograniczenia

$1 \leq N \leq 500\,000$ ,  $1 \leq H_i \leq 10^{18}$ .

## Przykład

Input	Output	Explanation
4 2 5 19 23	TAK	Projekt jest dobry.
6 3 2 2 1 4 4	NAPRAWA 1 4	Projekt nie jest dobry, ale zamienienie miejscami pierwszego i czwartego budynku sprawia, że jest już dobry.
5 100 39 42 21 3	NIE	Projekt nie jest dobry i można pokazać, że jedna dowolna zamiana nie sprowadzi go do dobrego projektu.



# Kosmiczna ochrona (kosmiczna-ochrona)

Memory limit: 128 MB Time limit: 5.00 s

*To jest zadanie interaktywne.*

Podczas rutynowego patrolu Sił Ochrony Kosmosu, ujawniono niezidentyfikowany wielokąt wypukły. Wywiad donosi, że wielokąt znajduje się w całości wewnątrz kwadratu o rogach  $(0, 0)$ ,  $(1000, 0)$ ,  $(0, 1000)$ ,  $(1000, 1000)$  włącznie z brzegiem (punkt  $(0, 0)$  może być wierzchołkiem wielokąta), a ponadto wierzchołki tego wielokąta wszystkie leżą w punktach kratowych. Twoim celem będzie wyznaczenie rozmiaru tego incydentu, czyli pola niezidentyfikowanego wielokąta wypukłego.

W tym celu możesz skorzystać z wielu statków zwiadowczych. Każdy statek zwiadowczy przy starcie otrzymuje konkretne docelowe miejsce stacjonowania, a po osiągnięciu celu przysyła raport – odległość do najbliższego punktu wielokąta.

W tym roku budżet SOK pozwala na użycie 8 000 tego rodzaju statków. Czy sprostasz wyzwaniu i wypełnisz misję?

## Interakcja

Dostępne są dwa rodzaje zapytań:

- ?  $x$   $y$  wyśle statek zwiadowczy w punkt  $(x, y)$  i wypisze na standardowe wejście raport z tego statku, czyli jeden wiersz zawierający jedną liczbę rzeczywistą, określającą odległość z punktu  $(x, y)$  do najbliższego punktu szukanego wielokąta;
- !  $p$  jest sposobem zwrócenia przez Ciebie wyniku całej misji, czyli  $p$  – pola szukanego wielokąta.

**Należy pamiętać o opróżnianiu bufora wypisywania.**

## Ograniczenia

Możesz zadać co najwyżej 8 000 zapytań typu ?, musisz zadać dokładnie jedno zapytanie typu ! i bezpośrednio po nim zakończyć program.

Punkty z zapytań typu ? muszą się znajdować w kwadracie  $[-10\,000, 10\,000]^2$ . Błąd względny zwracanych odpowiedzi na zapytania nie przekroczy  $10^{-5}$ .

Twoja odpowiedź zostanie zaakceptowana, jeżeli błąd względny nie przekroczy  $10^{-6}$ .

## Przykładowa interakcja

Input	Output
	? 2.00 3.00
0.000000	
	? 3.00 2.00
1.000000	
	? 5.00 7.00
1.414213	
	? 3.00 4.00
0.000000	
	! 6.00

Szukany wielokąt to prostokąt o wierzchołkach  $(2, 3)$ ,  $(4, 3)$ ,  $(4, 6)$  i  $(2, 6)$ .

# Kameleony (kameleony)

Memory limit: 128 MB

Time limit: 1.00 s

“Egipcjanie dawno już wybudowali swoje piramidy, wprawiając cały świat w podziw, dziś ich dzieło przyćmiewane jest przez nasze, nasze babilońskie wiszące ogrody, spójrz tylko. Widzisz tę idealną kompozycję. Czy dostrzegasz to niesamowite arcydzieło? Tutaj u nas każde drzewko współgra z każdym innym, tworząc idealne układy, sekwencje i schematy...” ~ tak rozwodziła się nad pięknem swych ogrodów legendarna władczyni Semiramida. Ty jako wierny swej królowej oczywiście musiałeś wysłuchiwać jej żmudnej przemowy. Udało Ci się już uciec daleko myślami, gdy nagle usłyszałeś swoje imię. Władczyni nagle stwierdziła, że jej ogrodom czegoś brakuje, że brakuje im ciągu  $N$  zieloniutkich kameleonów.

Oczywiście to Tobie Semiramida powierzyła misję sprowadzenia  $N$  kameleonów. Jako że nie jesteś w ciemę bity, szybko zapłaciłeś komu trzeba i w ogrodzie pojawiło się  $N$  leżących w rzędzie, dwukolorowych kameleonów. No właśnie dwukolorowych, a nie zieloniutkich. Kameleony były nieodpowiednio przechowywane w transporcie i część z nich zmieniła swój kolor.

Masz więc przed sobą ciąg  $N$  kameleonów (numerowanych kolejnymi liczbami naturalnymi od 1 do  $N$ ), część z nich ma kolor zielony, dokładnie taki jaki powinny mieć, a część z nich ma kolor czerwony, dokładnie taki jakiego nie powinny mieć. Zrezygnowany podchodzisz do jednego z kameleonów i głaszciesz go po grzbiecie. Ku Twojemu zdziwieniu kameleon i dwa kameleony obok (ten po prawej od głaskanego i ten po lewej od głaskanego) zmieniły swój kolor na przeciwny. Eksperyment powtórzyłeś wielokrotnie i ku Twojemu zdziwieniu efekt był dokładnie taki sam. Kameleony zielone zmieniały swój kolor na czerwony, a kameleony czerwone na zielony. Zastanawiasz się teraz, które spośród kameleonów należy pogłaskać, by osiągnąć ciąg  $N$  zielonych kameleonów. Postanowiłeś więc napisać program, który wczyta opis kolorów poszczególnych kameleonów i wyznaczy, które spośród kameleonów należy pogłaskać. Biorąc pod uwagę, że kameleony to groźne bestie, obawiasz się głaskać jakiegokolwiek z nich więcej niż jeden raz, ponadto spośród wszystkich możliwych rozwiązań postanowiłeś wybrać to najmniejsze leksykograficznie, bowiem właśnie takie wydaje Ci się najbezpieczniejsze.

## Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba naturalna  $N$ , oznaczająca liczbę kameleonów na wejściu. W drugim wierszu wejścia znajduje się ciąg  $N$  znaków złożony z liter C i Z,  $i$ -ta z nich odpowiada kolorowi kameleona  $i$ -tego. Litera C odpowiada kolorowi czerwonemu, a litera Z odpowiada kolorowi zielonemu.

## Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać jedno słowo – TAK lub NIE, w zależności od tego, czy istnieje sekwencja ruchów, sprowadzająca kolor wszystkich kameleonów do koloru zielonego. Jeśli odpowiedź jest twierdząca, to w drugim wierszu wyjścia należy wypisać najmniejszy leksykograficznie ciąg numerów głaskanych kameleonów sprowadzający wszystkie kameleony do koloru zielonego.

## Ograniczenia

$1 \leq N \leq 100\,000$ .

## Przykład

Input	Output
5	NIE
CZCCC	

Input	Output
6	TAK
CZCCZC	2 3 4 5

**Input**

3

ZZZ

**Output**

TAK

# Uprzemysłowienie (uprzemysłowienie)

Memory limit: 512 MB Time limit: 3.00 s

W zasadzie każdy człowiek, wchodząc na ulicę Implementacyjnych Dyrzymałów, staje z osłupieniem, podziwiając najwspanialszą na świecie Fabrykę. To arcydzieło cywilizacji dziewiętnastego wieku jest największą fabryką, jaka tylko powstała – dziesiątki tysięcy metrów kwadratowych powierzchni, wiele kondygnacji. Z zewnątrz ściany Fabryki zdobione są czerwoną cegłą, która w połączeniu z ogromnymi połączeniami przyciemnianego szkła sprawia niesamowite wrażenie. Gra światła wewnątrz budynku szokuje swym pięknem każdego odwiedzającego. Fabryka . . .

Fabryka powinna być najnowocześniejszym dziełem ludzkim, jakie tylko istnieje. Dziś opatentowany został nowy system posadzkowy. Oczywiście Fabryka musi zostać w taki system wyposażona. Zgodnie z tym systemem podłoga Fabryki powinna składać się z dwóch warstw, z których każda składa się z klocków w kształcie litery "L", złożonych z trzech mniejszych kwadratowych fragmentów o wymiarach  $1 \times 1$ . Ponadto, żaden klocek z warstwy drugiej nie może znajdować się nad tylko jednym klockiem z warstwy pierwszej.

Znamienici inżynierowie ułożyli już pierwszą warstwę posadzki na prostokątnym obszarze fabryki o wymiarach  $N \times M$ . Twoim zadaniem będzie wyznaczenie układu klocków warstwy drugiej, w którym żaden z klocków nie będzie znajdował się nad tylko jednym klockiem z warstwy pierwszej – każdy klocek z warstwy drugiej musi znajdować się na co najmniej dwoma różnymi klockami z warstwy pierwszej.

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne  $N$  i  $M$ , oddzielone pojedynczym odstępem i oznaczające wymiary posadzki Fabryki. W następnych  $N$  wierszach znajduje się po  $M$  liczb naturalnych  $A_{ij}$ , pooddzielanych pojedynczymi odstępami i oznaczającymi numer klocka na pozycji  $i, j$ .

Gwarantowane jest, że na wejściu podany będzie poprawny opis posadzki Fabryki.

## Wyjście

Na wyjście należy wypisać  $N$  wierszy, w każdym z nich powinno znaleźć się po  $M$  liczb z przedziału od 1 do  $N \cdot M$ , pooddzielanych pojedynczymi odstępami i reprezentującymi numery klocków, którymi zostało pokryte dane pole. Dane testowe zostały tak przygotowane, że istnieje poprawne rozwiązanie. Jeśli istnieje wiele poprawnych odpowiedzi, możesz wypisać dowolną z nich.

## Ograniczenia

$2 \leq N, M \leq 1000, 1 \leq A_{ij} \leq N \cdot M$ .

## Przykład

Input	Output
2 3	1 2 2
1 1 2	1 1 2
1 2 2	
Input	Output
5 6	1 1 3 3 5 5
1 1 3 3 5 5	2 1 3 4 6 5
1 2 4 3 5 6	2 2 4 4 6 6
2 2 4 4 6 6	7 8 8 9 9 10
7 7 8 9 10 10	7 7 8 9 10 10
7 8 8 9 9 10	

**Input**

6 6  
1 1 2 3 3 4  
1 2 2 3 4 4  
5 5 6 7 7 8  
5 6 6 7 8 8  
9 9 10 11 11 12  
9 10 10 11 12 12

**Output**

1 2 2 3 4 4  
1 1 2 3 3 4  
5 6 6 7 8 8  
5 5 6 7 7 8  
9 10 10 11 12 12  
9 9 10 11 11 12