



2026

Task Sheet

XIII Maratona Mineira de Programação

Sponsored by



Hosted by



Organized by



Problem A. Ana, Hooray for Mariana!

Hooray for Mariana!



Mariana Counts 1 is a song often used while children are learning to count, since it presents counting in a didactic and playful way. For older children, the song was generalized into Mariana Counts X (where X is a natural number), and goes as follows:

Mariana counts 1

Mariana counts 1: it is 1, it is Ana

Hooray for Mariana, hooray for Mariana!

Mariana counts 2

Mariana counts 2: it is 1, it is 2, it is Ana

Hooray for Mariana, hooray for Mariana!

...

Mariana counts X

Mariana counts X : it is 1, it is 2, ..., it is X , it is Ana

Hooray for Mariana, hooray for Mariana!

To measure its didactic value, print how many numbers appear in the song.

Input

The input consists of a single integer X ($1 \leq X \leq 1000$).

Output

The output should contain a single integer: how many numbers were counted in the song.

Examples

Sample input 1	Sample output 1
2	7

Explanation of sample 1

Mariana counted 7 numbers: four times the number 1 and three times the number 2, as can be seen in the first two stanzas of the statement.

Sample input 2	Sample output 2
5	25

Sample input 3	Sample output 3
10	75

Problem B. Bario World

It's-a-Me, Bario!



Because of its fun mechanics, Super Bario World is a hit among gamers from Minas Gerais. In the game, Bario moves through levels made of N blocks, each of which may be either a solid block or a hole. The goal is to get from the start of the level (block 1) to the end (block N) without falling into any hole. Bario can run over solid blocks and, to avoid holes, he can jump over them.

Bario wears boots with an energy charge, initially equal to 1. While running, he charges the boots so he can jump farther – each solid block Bario runs over adds one unit of charge to his equipment. A jump with charge v lets Bario go from block b to any block from $b + 1$ to $b + v$; after landing, the boots lose their energy and return to their initial charge of 1.

Compute the minimum number of jumps Bario needs to reach position N , or report that it is impossible.

Input

The first line contains an integer T ($1 \leq T \leq 5 \times 10^5$), the number of test cases. It is guaranteed that the sum of the values of N is less than 5×10^5 .

Then T pairs of lines follow. The first line of each pair contains N ($2 \leq N \leq 5 \times 10^5$), the length of the level, and the second contains a string of N characters representing the blocks of the level: `x` for a solid block and `.` for a hole. It is guaranteed that blocks 1 and N are solid.

Output

Print the minimum number of jumps Bario needs to reach the end of the level, or -1 if it is impossible.

Examples

Sample input 1	Sample output 1
<pre>2 16 xxxxxxxx.xxxx.x.x 3 x.x</pre>	<pre>2 -1</pre>

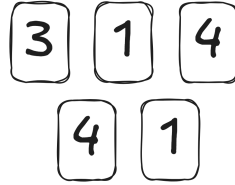
Explanation of sample 1

In the first test case, one possible solution with only two jumps has Bario:

- Run from position 1 to position 5, charging the boots to energy 5, and jump, landing at position 9. After landing, his boots go back to energy 1.
- Run from position 9 to position 12, charging the boots to energy 4, and jump to the end.

xxxxxxxx.xxxx.x.x

Problem C. Cards



One day, Rened decided to invite his friends to play a card game. The game consists of a sequence of numbered cards on a table, initially empty. The players take turns adding new cards to the end of this sequence; the i -th player plays the card with value C_i .

The main rule is simple: **there cannot be cards with repeated numbers on the table.**

Whenever a player plays a card whose number does not appear in the current sequence, it is simply added to the end, extending the sequence normally.

However, if a player plays a card with the same value as another card already on the table, an elimination occurs: all cards from the beginning of the current sequence up to and including the previous card with this value are removed. Then, the newly played card is placed at the end of the sequence. **Notice that after this process no repeated values remain on the table, and the sequence is never empty.**

To make the game faster, Rened asked you and your team to develop a system that tracks the state of the game and, after each play, reports the largest-value card still on the table and the index of the player who played it.

Input

The input has two lines. The first contains the value of N ($1 \leq N \leq 2 \times 10^5$), the number of players. The second contains the card values C_i ($1 \leq C_i \leq N$).

Output

The output should contain N lines, each containing two values separated by a space: the value C of the largest card still on the table and the index i of the player who played that card.

Examples

Sample input 1	Sample output 1
6	3 1
3 1 4 1 6 6	3 1
	4 3
	4 3
	6 5
	6 6

Explanation of sample 1

1st play:

Player 1 plays 3

Cards on the table: [3]

Largest card: 3 (player 1)

2nd play:

Player 2 plays 1

Cards on the table: [3, 1]

Largest card: 3 (player 1)

3rd play:

Player 3 plays 4

Cards on the table: [3, 1, 4]

Largest card: 4 (player 3)

4th play:

Player 4 plays 1

The prefix up to the previous 1 is removed

Cards on the table: [4, 1]

Largest card: 4 (player 3)

5th play:

Player 5 plays 6

Cards on the table: [4, 1, 6]

Largest card: 6 (player 5)

6th play:

Player 6 plays 6

Cards on the table: [6]

Largest card: 6 (player 6)

Sample input 2

```
12
1 1 4 6 5 8 4 6 8 2 5 2
```

Sample output 2

```
1 1
1 2
4 3
6 4
6 4
8 6
8 6
8 6
8 9
8 9
8 9
8 9
5 11
```

Problem D. Dubious Dates



NeoSpace is an innovative startup that uses advanced Artificial Intelligence models to process massive volumes of corporate information. In the company’s hallways, engineers often tell a classic joke: despite being experts at handling Big Data, the system’s greatest nightmare is still the small dates entered incorrectly by users.

Recently, the platform started integrating data from new international clients. During data ingestion, the team noticed a serious problem in the interface: while the system expects the Brazilian Day/Month (DD/MM) format, many foreign users fill out the forms in the Month/Day (MM/DD) format.

To ensure the quality of the AI model, the pipeline must identify when an entry may cause confusion. For example, the value 1/6 is problematic, since it may be interpreted as June 1st or January 6th. The value 28/2, on the other hand, is safe, since there is no month 28 in the calendar. Also, a value such as 9/9 is safe: both interpretations refer to the same date.

Your task is to create an algorithm that analyzes the input provided by the user and determines whether the given numbers can lead to this double interpretation.

Input

The input consists of a single line containing two integers D and M ($1 \leq D \leq 28, 1 \leq M \leq 12$), representing, respectively, the day and the month of the event.

Output

Print `DATA INCERTA` if the date can be interpreted in Month/Day format as a different valid date, and `DATA SEGURA` otherwise.

Examples

Sample input 1	Sample output 1
28 2	DATA SEGURA
Sample input 2	Sample output 2
1 6	DATA INCERTA
Sample input 3	Sample output 3
9 9	DATA SEGURA

Explanation of sample 3

The date can only be interpreted as September 9.

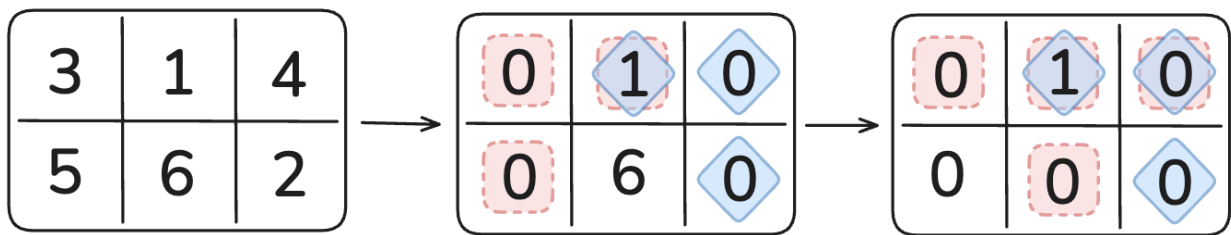
Problem E. Exploring the Terrain



Giovana and Arthur have just been hired by ArcelorMittal Sistemas to develop cutting-edge technology for mineral extraction. As their first project, both implemented algorithms and heuristics to create an exploration plan for a drilling machine, a machine commonly used in mines, with the goal of maximizing the ore extracted after T minutes of operation. The mine where Giovana and Arthur are working is represented by an $N \times M$ matrix, where each cell contains the value obtained if that position is drilled by the machine. Each exploration plan consists of T pairs of coordinates $(i_1, j_1), \dots, (i_T, j_T)$, where (i_t, j_t) is the row and column where the drilling machine will be at time t .

Very pleased with the projections of Giovana's and Arthur's algorithms, the company's directors decided that both exploration plans will be executed simultaneously, each with its own drilling machine. The problem is that Giovana's and Arthur's exploration plans were not designed with the presence of the other drilling machine in mind, which could cause conflicts during ore extraction!

Fortunately, the control software developed by ArcelorMittal Sistemas is quite sophisticated and can handle this kind of situation! Besides the mechanical control of the drilling machines, it establishes a communication network between the pieces of equipment, preventing the machines from exploring the same position simultaneously. Thus, when a drilling machine is at position (i, j) , it can extract ore from cell (i, j) and from cells $(i - 1, j)$, $(i, j - 1)$, $(i, j + 1)$, $(i + 1, j)$, but it does so only if the other drilling machine would not extract that same cell at that same time. Extraction is 100% efficient: after its ore has been extracted, a cell has zero units of ore. In the example in the figure below, if at time t Giovana's drilling machine is at position $(1, 1)$ and Arthur's is at position $(1, 3)$, **neither drilling machine will extract ore from cell $(1, 2)$, leaving all the ore that was there intact**. On the other hand, all other cells of interest to each drilling machine will have their ore extracted.



Visual representation of Example 1. On the left, the matrix with the amounts of ore in each cell. In the center, the cells explored by each drilling machine at time $t = 1$: red squares are cells explored by Giovana, and blue diamonds are cells explored by Arthur. On the right, the cells explored by each drilling machine at time $t = 2$;

Curious to know how their plans will fare, Giovana and Arthur asked for your help. Write a program that, given the matrix representing the mine and the exploration plans of each of them, prints the total ore extracted by each drilling machine. It is guaranteed that both exploration plans have the same duration T .

Input

The first line contains three integers, N , M , and T , ($1 \leq N, M \leq 500$, $1 \leq T \leq N \times M$), the number of rows and columns of the matrix and the duration of the exploration plans, respectively.

The next N lines contain M integers each, ($1 \leq v_1, \dots, v_M \leq 1000$). The j -th number on the i -th line contains the ore value present at position (i, j) .

The next T lines contain two integers each. The t -th of these lines contains $(i_t, j_t)_G$ ($1 \leq i \leq N$, $1 \leq j \leq M$), the position where Giovana's drilling machine will be at time t .

Finally, the last T lines contain two integers each. The t -th of these lines contains $(i_t, j_t)_A$ ($1 \leq i \leq N$, $1 \leq j \leq M$), the position where Arthur's drilling machine will be at time t .

Output

Print two numbers G and A separated by a space – the total sum of the values extracted by Giovana and Arthur, respectively.

Examples

Sample input 1

```
2 3 2
3 1 4
5 6 2
1 1
1 2
1 3
1 3
```

Sample output 1

```
14 6
```

Sample input 2

```
3 1 3
10
1
10
2 1
3 1
1 1
2 1
2 1
2 1
```

Sample output 2

```
0 20
```

Sample input 3

```
1 2 2
6 7
1 2
1 1
1 1
1 2
```

Sample output 3

```
0 0
```

Problem F. Food Blockade



Among the many delicacies of Minas Gerais, cheese and doce de leite are the most famous. Although both are made from the same raw material, milk, their production and storage processes are quite different. In particular, all the cheese in the state is produced in city A_1 and stored in city A_2 , while all the doce de leite is produced in city B_1 and stored in city B_2 . To move the stocks from one city to another, each product is transported along some of the M two-way roads, each of which connects exactly two of the N cities in Minas Gerais.

Besides their culinary importance, cheese and doce de leite are fundamental to the state's political and social stability: a shortage of both delicacies, even for a short time, is enough to turn our characteristic hospitality into deep animosity. Concerned about this possibility, the Movement of Minas Pacifists (MMP) needs to know: what is the minimum number of roads that, if closed, would make it impossible to transport the cheese from A_1 to A_2 and the doce de leite from B_1 to B_2 ? Write a program to help the MMP find this answer.

Input

The first line of the input contains two integers N and M ($4 \leq N \leq 2000$, $0 \leq M \leq 2000$), indicating the number of cities and roads in Minas Gerais, respectively.

Each of the next M lines contains two integers u and v ($1 \leq u, v \leq N$, $u \neq v$), indicating that there is a road between cities u and v . It is guaranteed that there is at most one road between the same pair of cities.

After these M lines, there is one last line containing four distinct integers A_1 , A_2 , B_1 , and B_2 , indicating the production and storage cities of the cheese and the doce de leite, respectively.

Output

The output should contain a single integer, indicating the minimum size of a set of roads that, after being closed, makes both transports impossible.

Examples

Sample input 1	Sample output 1
<pre>6 5 1 3 2 3 3 4 4 5 4 6 1 5 2 6</pre>	<pre>1</pre>

Explanation of sample 1

In this example, if the road connecting cities 3 and 4 is closed, there will no longer be a path between cities 1 and 5, nor between cities 2 and 6.

Sample input 2

```
6 4
1 2
2 3
4 5
5 6
1 3 4 6
```

Sample output 2

```
2
```

Explanation of sample 2

In this example, if the roads between 1 and 2, and between 4 and 5, are closed, this would prevent both the existence of a path between 1 and 3 and of a path from 4 to 6. It can be proven that this is the minimum number of roads that must be closed for this to happen.

Sample input 3

```
5 4
1 3
3 4
3 5
1 5
1 2 4 5
```

Sample output 3

```
1
```

Explanation of sample 3

In this example, it is already impossible to go from city 1 to city 2. Therefore, if the road from 3 to 4 is closed, it becomes impossible both to go from 1 to 2 and to go from 4 to 5.

Problem G. Garment Groups

After each contest, it is up to us to contemplate everything we experienced: past problems, balloons earned, friendships made along the way, and the shirts worn during the competition.



In the figure above, there are four shirts, using seven clothespins, forming three groups of shirts.

To hang a shirt on the clothesline, we must use two clothespins (one at each end of the shirt). On the other hand, with two shirts, we can use only three clothespins by fastening the ends of two consecutive shirts with the same clothespin. When this happens, we say that a group of shirts has been formed, and that the shirts fastened by the same clothespin belong to the same group. Being in a group is a transitive property: if shirt A belongs to the same group as shirt B, and shirt B belongs to the same group as shirt C, then shirts A and C also belong to the same group. A shirt that does not share a clothespin with any other shirt forms a group by itself.

Dilson wants to hang N shirts on the clothesline and, at the same time, wants exactly G groups of shirts to be formed. What is the minimum number of clothespins Dilson needs to use to achieve his goal?

Input

The only line of input contains two integers N ($1 \leq N \leq 10^6$) and G ($1 \leq G \leq N$), the number of shirts Dilson wants to hang and the number of groups he wants to form, respectively.

Output

The output should contain a single integer: the number of clothespins Dilson needs to use to hang N shirts in G groups.

Examples

Sample input 1	Sample output 1
4 3	7

Explanation of sample 1

As shown in the figure, Dilson must use seven clothespins to form three groups.

Sample input 2	Sample output 2
2 1	3

Explanation of sample 2

Dilson must use three clothespins: one at one end of the first shirt, one fastening the other end of the first shirt and one end of the second shirt, and one at the other end of the second shirt. Thus, the two shirts are fastened by the same clothespin, forming a group of shirts.

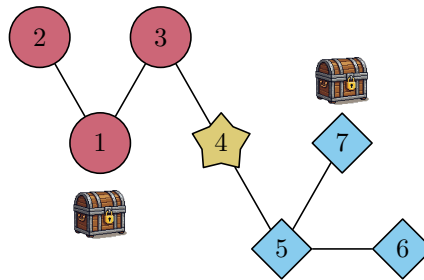
Problem H. Heritage Hunt



The Maratona Mineira team is organizing a treasure hunt. The treasure will be two chests of the most delicious doce de leite in the world, a cultural heritage of Minas Gerais, which will be hidden on the game map.

The map consists of N vertices, numbered from 1 to N , and $N - 1$ edges, in such a way that it is possible to go from any vertex to any other. In other words, the map forms a tree. The distance between two vertices is defined as the number of edges in the unique simple path between them.

The team already has the map, but still needs to choose in which vertices to hide the two chests. A choice is considered valid if the chests are placed on distinct vertices and the number of vertices closer to the first chest is equal to the number of vertices closer to the second chest. Vertices at the same distance from both chests do not count towards either one.



The figure above shows a valid choice for Example 1. The chests were placed at vertices 1 and 7. Thus, vertices 1, 2, and 3 (red circles) are closer to the chest at vertex 1, while vertices 5, 6, and 7 (blue diamonds) are closer to the chest at vertex 7. Vertex 4 (yellow star) is at the same distance from both chests.

Determine the number of valid ways to choose the positions of the chests. Two ways are considered distinct if the set of vertices where a chest was placed is different.

Input

The first line of the input contains an integer N ($2 \leq N \leq 2 \times 10^5$), the number of vertices in the map.

Each of the next $N - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$), indicating that there is an edge between vertices u_i and v_i . It is guaranteed that the edges form a tree.

Output

Print a single integer: the number of distinct valid ways to choose the positions of the two chests.

Examples

Sample input 1	Sample output 1
7 1 2 1 3 3 4 4 5 5 6 5 7	4

Explanation of sample 1

The 4 valid choices, represented by the positions of the chests, are: $\{1, 7\}$, $\{1, 6\}$, $\{3, 5\}$, and $\{6, 7\}$.

Sample input 2	Sample output 2
7 1 2 1 3 3 4 1 5 5 6 6 7	0

Explanation of sample 2

In this example, there are no valid choices for the positions of the chests.

Problem I. Interlingual Intermediaries

Over the ages, each people developed its own way of speaking, and each language carried the culture and way of life of its speakers. The Elves created Elvish, a softer and more elegant language; the dragons created Draconic, representing their dominance and ferocity; and, little by little, all the other peoples created their own languages as they pleased.



A great meeting among diplomats from N peoples will be held to discuss the future of the continent. The event has a total of M cataloged languages. The problem is that not all diplomats share a common language. For a message to get from one point to another, it may be necessary to use other diplomats as intermediaries.

However, every time the message must be translated from one language to another (when an intermediary hears it in one language and speaks it in another), a **language switch** occurs. To minimize misunderstandings, the conclave organizers need to find the minimum number of switches required for two diplomats to communicate.

Given the sets of languages spoken by each of the N diplomats and Q queries (a, b) , determine the minimum number of language switches needed for diplomats a and b to communicate. If communication is impossible, print -1 .

Input

The first line contains three integers N , M , and Q ($2 \leq N \leq 10^4$, $1 \leq M \leq 30$, $1 \leq Q \leq 2 \times 10^5$), the number of diplomats, the total number of languages, and the number of queries, respectively.

Then N lines follow, each composed first of an integer m_i ($1 \leq m_i \leq M$) – how many languages the i -th diplomat speaks – followed by m_i distinct values $\ell_1, \dots, \ell_{m_i}$ ($1 \leq \ell \leq M$) – the languages spoken by the diplomat.

Finally, the input contains Q more lines, each containing two integers a and b ($1 \leq a, b \leq N$, $a \neq b$), the initial and final diplomats in the query.

Output

The output should contain Q lines, each answering how many language switches are necessary for a and b to communicate; if it is not possible, print -1 .

Examples

Sample input 1	Sample output 1
6 7 3 2 3 6 3 3 2 7 2 1 2 2 1 5 1 4 1 1 4 1 2 3 2 5	2 0 -1

Explanation of sample 1

- In the first query, one possible way to transmit the message is:

$4 \xrightarrow{\text{language 1}} 6 \xrightarrow{\text{language 1}} 3 \xrightarrow{\text{language 2}} 2 \xrightarrow{\text{language 3}} 1$ – with two language switches, 1/2 and 2/3.

- In the second query, the message can be transmitted with no language switches: $2 \xrightarrow{\text{language 2}} 3$.
- In the third query, diplomat 5 speaks only language number 4, which is not spoken by anyone else, so it is not possible to transmit the message to them.

Problem J. Just the Betas



A group of N friends is going to play a game in which each player chooses an integer P_i . Then an integer X greater than 1 was supposed to be drawn, but Pedro found a way to cheat and can choose this value.

Each player's score is given by $P_i \bmod X$, that is, the remainder of the division of P_i by X .

Any player whose score is equal to 0 is called a *beta*, because nothing is left from the division.

Help Pedro choose a value of X that maximizes the number of beta players.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 10^5$), representing the number of players.

The second line contains N integers P_i ($2 \leq P_i \leq 10^6$), representing the value chosen by the i -th player.

Output

Print a single integer X ($2 \leq X \leq 10^6$) that maximizes the number of beta players.

If there are multiple values of X that produce the same maximum number of beta players, print any of them.

Examples

Sample input 1	Sample output 1
5 2 5 4 5 3	2
Sample input 2	Sample output 2
2 19 20	19

Examples

Sample input 1

2 500000003

Sample output 1

7

Explanation of sample 1

The possible games are:

- Field BB - 1 possible game; the player already starts winning.
- Field B. - 2 possible games: the player clicks the left position and loses, or the right position and wins.
- Field .B - 2 possible games: the player clicks the left position and wins, or the right position and loses.
- Field .. - 2 possible games: the player clicks either the left or the right position and wins.

Sample input 2

5 500000003

Sample output 2

205

Sample input 3

3000 999999733

Sample output 3

842327137

Problem L. Loading the Dish Rack

After an excellent family breakfast with cheese bread, Beto was put in charge of washing the dishes. After soaping everything, Beto remembered how annoying it is to put the cups in his home's linear dish rack, especially the work needed to place a cup between two others that are already drying. He estimated that the effort to place a cup in a given position of the dish rack is equal to the sum of the number of consecutive cups immediately to the left and the number of consecutive cups immediately to the right of that position at the time of placement.



In the figure above, we have the final state of a solution for Example 1 – 4 cups in a dish rack with 6 positions.

Given the size N of the dish rack and the number C of cups, compute the minimum total effort Beto must spend to place all cups in the dish rack and provide an insertion sequence that achieves this minimum effort. The dish rack positions are numbered from 1 to N , and each cup must be placed in an empty position. If there is more than one way to achieve the minimum effort, print any of them.

Input

The only line of input contains two integers N ($2 \leq N \leq 10^6$) and C ($1 \leq C \leq N$), the size of the dish rack and the number of cups, respectively.

Output

The first line of the output should contain a single integer, the minimum total effort Beto must spend to place all cups in the dish rack.

The second line of the output should contain C integers separated by spaces, where the i -th integer represents the dish rack position where the i -th cup should be placed.

Examples

Sample input 1	Sample output 1
6 4	1 3 4 1 6
Sample input 2	Sample output 2
7 5	2 1 2 4 6 7

Explanation of sample 2

One way to place the cups is [1 2 – 3 – 4 5]. Cups 1, 3, and 4 have placement cost 0, while cups 2 and 5 have cost 1, for a total effort of 2.

Sample input 3

Sample output 3

15 14	20 1 3 2 5 7 6 4 9 11 10 13 15 14 12
-------	---

Problem M. Moving Sticks

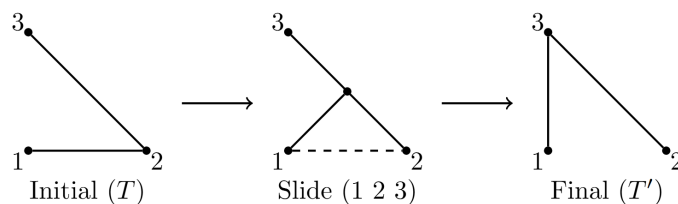


The renowned creators of pick-up sticks have just released the sliding-sticks puzzle. The puzzle consists of $N - 1$ sticks connected to N pins placed in the plane in **convex position**, forming a **planar tree**, that is, a connected and acyclic structure with no crossings.

The puzzle comes assembled in an initial configuration T , and the goal is to transform it into the final configuration T' illustrated on the game box.

However, there is an important mechanical limitation: a stick cannot be moved freely. The only valid move is called a **slide** and works as follows:

- Consider two sticks ab and bc . It is allowed to detach the common endpoint at b of stick ab and move it along stick bc , until connecting it to c .
- At the end of the move, no two sticks are allowed to cross.



Example: sliding edge 12 along edge 32.

We say that two sticks **cross** if their segments have a point in common that is not a shared endpoint; see the explanation of the second example below.

Since you have a dinner coming up and are short on time, you decided to add an extra restriction: your goal is to find a sequence of at most $2N$ slides that transforms T into T' .

Input

The first line of the input contains an integer N ($3 \leq N \leq 10^5$), indicating the number of pins.

Each of the next N lines contains two integers x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$), the coordinates of the pins in the plane. The pins are given in counterclockwise order along the convex hull.

Then $N - 1$ lines follow, each containing integers a and b ($1 \leq a, b \leq N, a \neq b$), indicating the sticks of the initial configuration T .

Finally, the next $N - 1$ lines contain integers a' and b' ($1 \leq a', b' \leq N, a' \neq b'$), indicating the sticks of the final configuration T' .

It is guaranteed that the pins are in convex position, that is, no three pins are collinear and, in the order in which they are given, they form a convex polygon (so no point lies inside this polygon). Additionally, the configurations T and T' are planar trees, that is, they have no crossings.

Output

Print an integer M ($0 \leq M \leq 2N$), representing the number of slides. **It is not necessary to minimize the number of slides**; it can be proven that there is always a solution with at most

$2N$ slides.

In the next M lines, print three integers a , b , and c , indicating a slide in which edge ab is modified by detaching the endpoint at b and sliding it along edge bc , resulting in edge ac .

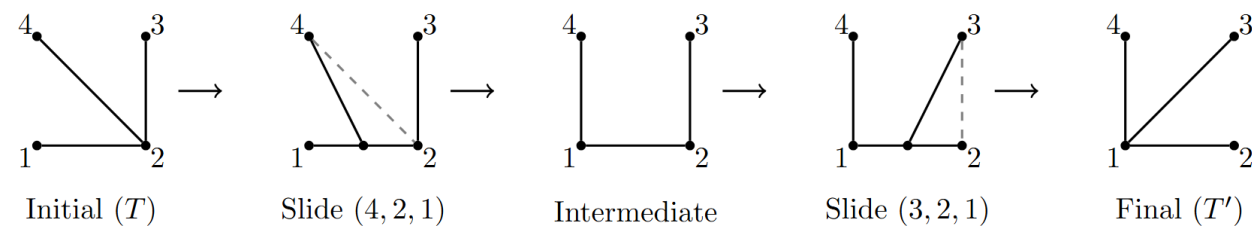
Examples

Sample input 1	Sample output 1
3	2
0 0	3 2 1
1 0	2 1 3
0 1	
1 2	
2 3	
1 3	
2 3	

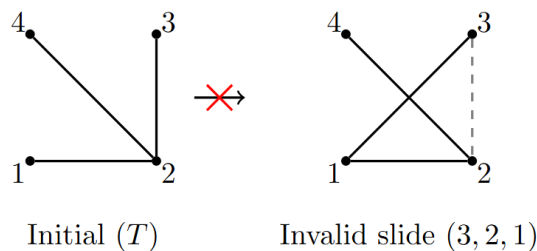
Sample input 2	Sample output 2
4	2
0 0	4 2 1
1 0	3 2 1
1 1	
0 1	
1 2	
2 3	
2 4	
1 2	
1 3	
1 4	

Explanation of sample 2

The figure below shows the sequence of operations from the initial state to the final state in Example 2. Notice that there is more than one solution.



The figure below shows an invalid slide. In configuration T , sliding edge 32 along edge 21 is invalid, because it creates a crossing between sticks 31 and 42 .



Problem N. N-Checkers

Few people know this, but according to a local legend the Portuguese name ‘damas’ is actually a linguistic corruption of ‘dez-amas’, a subset of the legendary n-amas.



The n -checkers board is an $n \times n$ board with squares alternating between white and black, with the square in the top-left corner being white. A square may be empty or, if it is black, contain a piece.

On their turn, a player must choose one of their pieces to move. The chosen piece may capture an opponent's piece if that piece is on a diagonally adjacent square and the next square in the same diagonal direction is empty. In this case, the chosen piece “jumps” over the opponent's piece, moves to the immediately following empty square, and removes the opponent's piece from the board. After a capture, the player may continue the move with the same chosen piece.

In the official rules of n -checkers, there is a majority-capture rule: if on their turn a player has the opportunity to capture opponent pieces, they **must perform a sequence of moves that results in the largest possible number of captured pieces**.

During the final of the Minas Gerais n -checkers Tournament, a dispute arose between the finalists. Alice, playing with the black pieces, accused Bob, playing with the white pieces, of having made an illegal move. Bob claims he captured as much as he could, but Alice says there was a longer path. Write a program that determines the maximum number of pieces Bob can capture in one move sequence.

Input

The first line contains a single integer n ($4 \leq n \leq 14$), the dimension of the board.

Then n lines follow, each with n characters representing the board. Each line contains only the characters ‘B’, ‘P’, and ‘.’, representing white pieces, black pieces, and empty squares, respectively. The letters come from Portuguese: ‘B’ for white pieces and ‘P’ for black pieces.

The board will contain at least one piece of each color, and no piece will be on a white square.

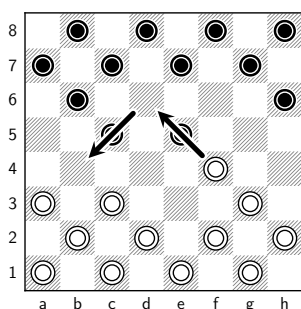
Output

Print the maximum number of pieces Bob can capture.

Examples

Sample input 1	Sample output 1
<pre> 8 .P.P.P.P P.P.P.P. .P.....P ..P.P...B.. B.B...B. .B.B.B.B B.B.B.B. </pre>	<pre> 2 </pre>

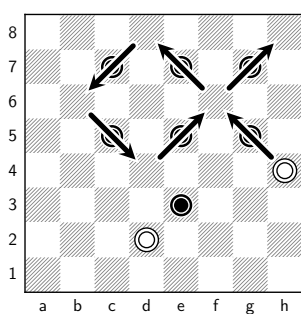
Explanation of sample 1



The move to be made, in which one of Bob's pieces captures two of Alice's pieces.

Sample input 2	Sample output 2
<pre> 8P.P.P.P.P.P.BP... ...B.... </pre>	<pre> 6 </pre>

Explanation of sample 2



One of the possible moves that can be made, in which one of Bob's pieces captures six of Alice's pieces.

Sample input 3**Sample output 3**

```
14
.B.P.B.B.P.B.P
B...B...B...B.
.P.B.P.P.B.P.B
..B...B...B...
.B.P.B.B.P.B.P
B...B...B...B.
.P.B.P.P.B.P.B
..B...B...B...
.P.B.P.P.B.P.B
B...B.B.B...B.
.B.P.B.B.P.B.P
..B...B...B...
.P.B.P.P.B.P.B
B...B.B.B...B.
```

```
12
```

Problem O. Operations in Order



During mathematics class, the students in the back row made a huge ruckus. Annoyed by the situation, the teacher did not let it slide and decided to assign a challenging task as punishment.

The task consists of a list, initially empty, of mathematical operations containing addition and multiplication. There are also N instructions. Each instruction has one of the following formats:

- $+ x$: Append an addition by x to the end of the list of operations.
- $* y$: Append a multiplication by y to the end of the list of operations.
- $? z$: Starting from value z , apply the list of operations **in order**, and output the result modulo $10^9 + 7$ – the remainder of the integer division by $10^9 + 7$.

Given the list of instructions, help the students with the assigned task.

Input

The input consists of several lines. The first line of the input contains the value of N ($2 \leq N \leq 5 \times 10^5$).

Each of the next N lines of the input contains an instruction as described in the statement – a character (+, * or ?) followed by a number $1 \leq x, y, z \leq 10^9$. It is guaranteed that the last instruction is of type '?' and that before the first instruction of type '?' there is at least one instruction of type '+' or '*'.

Output

The output consists of several lines. Each line should contain an integer representing the answer for each instruction of type "? z". Print the answer modulo $10^9 + 7$.

Examples

Sample input 1	Sample output 1
<pre>4 + 1 * 3 + 2 ? 2</pre>	<pre>11</pre>

Explanation of sample 1

The first query consists of the sequence of operations $2 \xrightarrow{+1} 3 \xrightarrow{\times 3} 9 \xrightarrow{+2} 11$.

Sample input 2	Sample output 2
<pre>6 * 1000000000 + 6 ? 1 + 1 ? 1 ? 998244353</pre>	<pre>1000000006 0 12289585</pre>