

2019 ICPC Asia Tehran Regional Contest

Problem A : Petrol

The government of Neverland has recently announced a new petrol rationing plan with an unexpected price hike. According to the new plan, each person receives a quota of 60 liters per month in a fuel card. Each liter of petrol costs 1500 Oshloobs if it is within quota. Any extra fueling costs 3000 Oshloobs per liter.

After recovering from the shock, Mahya is trying to figure out how dark is the future. The current month is coming to an end, and Mahya has some quota left in her fuel card, remaining available for the next month. A quota of 60 liters will also be added to her fuel card just at the beginning of the next month. She also has a prediction of the amount of petrol that will be used in the next month. She now wants to know how much she should pay for petrol in the next month. However, she is too lazy to do that on her own. So, she needs your help to calculate the cost for her.

Input

The input consists of two lines. The first line contains an integer n ($0 \leq n \leq 200$), specifying the amount of petrol that will be used in the next month. The second line contains an integer k ($0 \leq k \leq 360$), showing the quota left in Mahya's fuel card at the end of current month.

Output

Print the amount of money (in Oshloobs) that Mahya will pay for petrol in the next month.

Example

Standard Input	Standard Output
41 0	61500
Standard Input	Standard Output
125 40	225000

2019 ICPC Asia Tehran Regional Contest

Problem B : Gold Rush

Again, Neverland has experienced a very bad economic condition over the past few months. The value of Oshloob, the national currency of Neverland, changes against one unit of gold very rapidly. People in Neverland, all wondering about their savings, are trying to exchange their savings with gold coins.

Dr. Predictman who is a data scientist, has obtained a prediction of the price (in Oshloobs) of a gold coin for the next n days based on the existing data over the past 40 years. He believes his prediction, and now he wants to increase his savings based on it. He was wondering how much savings he has at the end of the n -th day assuming that he has c Oshloobs at the beginning of the first day. Since Dr. Predictman is not a programmer, he asks you to help to find his answer.

Input

The first line of the input contains two integers c ($0 \leq c \leq 3000$), Dr. Predictman's initial savings in Oshloobs, and n ($0 \leq n \leq 30$), the period of his prediction. Each of the next following n lines contains an integer p_i ($1000 \leq p_i \leq 2000$) denoting the price of a gold coin at day i ($1 \leq i \leq n$) in Oshloobs.

Output

The output contains just an integer, which indicates the maximum savings he can obtain at the end of the n -th day assuming that Dr. Predictman exchanges all his remaining gold coins (if there is any) to Oshloobs at the end of the n -th day.

Example

Standard Input	Standard Output
1000 3 1000 1100 1200	1200

Standard Input	Standard Output
2000 4 1000 2000 1500 1800	4600

Problem C : Valid Emails

This year, many people registered for the internet contest with several email addresses. We want to see how many valid and distinct email addresses registered.

A valid email address consists of a username and a domain name separated by a character '@'. A username is a string containing letters (a-z and A-Z), digits (0-9), underscores (_), and periods (.). Usernames cannot begin or end with a period and cannot contain two consecutive periods. Other than this rule, periods do not matter in email addresses (they can be removed without changing the address). Uppercase and lowercase letters in the usernames are considered the same. So, usernames AliBaba and ali.baba are considered the same. Usernames should contain 6 to 30 characters, after removing all of its periods.

A valid domain name is a string of length between 3 and 30 (inclusive), consisting of domain parts separated by periods (.). A domain name must not start or end with a period. Each domain part is a non-empty string of letters (a-z and A-Z), digits (0-9), and dash (-). Uppercase and lowercase letters in the domain names are also considered the same. So, Foo.bar is the same as foo.Bar, but not the same as Foo-Bar or FooBar.

Input

The first line of the input contains a positive integer n ($1 \leq n \leq 1000$), the number of the registered email addresses. Each of the next n lines contains one email address of length at most 100 and consisting of alphabets, digits, '@', '.', '_', and '-'.

Output

Print a single integer which is the number of distinct email addresses that are valid.

Example

Standard Input	Standard Output
5 acmacm@icpc.ir acmacm..a@icpc.ir .ala1.abc@icpc.ir acma.c.m@icpc.ir acmacm@icpc.com	2

2019 ICPC Asia Tehran Regional Contest

Problem D : Cafebazaar's Chess Tournament

Ali hosts a yearly chess tournament for CafeBazaar's Shab-e Yalda festival. In a chess tournament, each pair of participants play a game against each other exactly once. Moreover, players are granted one point for a win, a half point for a draw, and no points for a loss toward their tournament score.

Danial has built a system to predict the result of Ali's tournament. Based on experience, he has assigned an opening skill and an ending skill to each of n participants in the tournament. For the i -th participant, let us denote the former with o_i and the latter with e_i . In a game between the i -th and j -th participants, Danial decides the result of the game according to the following rules:

1. If $o_i > o_j$ and $e_i > e_j$, then the i -th participant wins the game.
2. If $o_j > o_i$ and $e_j > e_i$, then the j -th participant wins the game.
3. Otherwise, the game ends in a draw.

To make the tournament more exciting, Ali wants to invite Danial to join the other n participants in the tournament. Since Danial has no prior experience in chess, he decides to practice for the tournament. Based on the amount of training, Danial can end up with any opening and ending skill. However, Danial has promised Ali that he will train in such a way that his opening skill will be **different** from the opening skill of the other participants. He will also keep his ending skill **different** from the ending skill of the other participants.

For his advertisement campaign, Ali wants to know the number of distinct possible final scores that Danial might get based on Danial's rules mentioned above. For example, Danial can achieve the scores 0, 1.5, 2.5, 3, 4, and 5 in the sample. For instance, the score 3 is obtained by setting the opening and ending skills of Danial to be 1.5. Since Ali and all other CafeBazaar programmers are busy planning the event, he has turned to you for help. Write a program to calculate this value.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 200\,000$), the number of participants. The i -th line of the next n lines contains two integers o_i and e_i ($1 \leq o_i, e_i \leq n$), the opening and ending skills of the i -th participant, respectively. Note that the limits for opening and ending skills do not apply to Danial's opening and ending skills. More specifically, Danial's opening and ending skills can be any real numbers.

Output

In the only line of the output, print the number of distinct possible final scores for Danial.

Example

Standard Input	Standard Output
5 1 1 1 2 1 1 2 1 2 2	6

2019 ICPC Asia Tehran Regional Contest

Problem E : The Big Surprise

“A big surprise is coming on the next Thursday!”, the young mayor of TetrisCity announced in the social media. TetrisCity is the most populous and modern city in Neverland constructed on a flat area with endless clusters of high-rise buildings packed so closely together that they resemble a game of Tetris. Buildings look like axis-parallel boxes constructed on the ground and they are disjoint (they do not even touch each other).

The big surprise announced by the mayor is going to be a special delivery service using drones. The drones used in this service are a generation of quadcopters which can physically move only in one of x , y , and z directions. So, the distance traveled by a drone is the sum of distances traveled by it in each axis. The young mayor now has ordered to make the drones smart by equipping them with a software that computes the shortest path from any source to any destination avoiding the buildings. Your job is to develop this software.

Input

The first line of the input contains an integer n ($0 \leq n \leq 100$), specifying the number of buildings in the TetrisCity. Each of the next n lines contains 5 space-separated integers x , y , x' , y' , and h specifying a building: the coordinates (x, y) and (x', y') respectively specify the west-south corner and the east-north corner of the building, and h determines its height. It is guaranteed that the volume of the building is not zero. The source and destination appear at the end of the input in two separated lines; each containing x , y , and z coordinates. All numbers in the input are non-negative integers being at most 100 000. It is guaranteed that the source and destination are outside the buildings (they can be on the boundary of buildings). The shortest path can touch buildings and it is assumed that a drone looks like a point.

Output

In the output, print the length of the shortest path from the source to the destination avoiding the buildings.

Example

Standard Input	Standard Output
1 1 1 11 21 40 5 0 5 6 23 8	35

Problem F : Lets Burn and Rob Manhootan

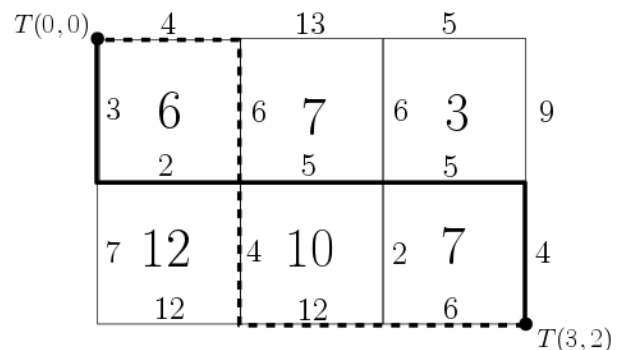
There are two types of angry people in this world, those who burn and those who rob. But we programmers know that there is a third type; those who counter their anger, by both burning and robbing.

Bob lives in Manhootan. The city of Manhootan is like a grid of n rows and m columns, containing $n \times m$ blocks. The rows are numbered from 0 to $n - 1$ from north to south and the columns are numbered from 0 to $m - 1$ from west to east. The j -th block on i -th row is worth A_{ij} . Before the first row, between every two consecutive rows, and after the last row, there is a west-east street. The $n + 1$ west-east streets are numbered from 0 to n from north to south. Similarly, before the first column, between every two consecutive columns, and after the last column, there is a north-south street. The $m + 1$ north-south streets are numbered from 0 to m from west to east. The part of a street that is between two adjacent blocks is called a *street segment*. Each west-east street contains m street segments, numbered from 0 to $m - 1$ from west to east. Similarly, each north-south street contains n street segments, numbered from 0 to $n - 1$ from north to south. Since Manhootan is an expensive city, passing through street segments costs money. Passing through the j -th segment of the i -th west-east street costs H_{ij} and passing through the j -th segment of the i -th north-south street costs V_{ij} .

**This is Bob.
Bob is Angry.
Bob wants to
burn and rob.
Don't be like Bob.**



After a recent crisis in Manhootan, Bob got angry. He pierced his car's fuel tank to make it leak on the streets he passes. Let's call the intersection of i -th west-east street and j -th north-south street, $T(i, j)$. At first, Bob is at $T(0, 0)$. He is planning to drive to $T(n, m)$ only going east and south, then returning to $T(0, 0)$ only going west and north. Then, he is going to light the leaked fuels and put the streets on fire. After that, Bob will rob all the blocks that are caught inside the fire, i.e., any block that can not reach outside of Manhootan without crossing a burning street, will be robbed by Bob. The figure shows one possible plan for Rob in the sample.



Now, you can't be like Bob, but you can help him find the most profitable burn-and-rob plan. In other words, maximize the total value of the robbed blocks minus the total cost of the passed street segments. A street segment may be passed twice, which should be paid for each separately.

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 200$), the number of rows and columns, respectively. The next n lines describe the value of blocks; each containing m numbers, where the j -th number of the i -th line denotes A_{ij} ($1 \leq A_{ij} \leq 100$). The next $n + 1$ lines describe the cost of west-east street segments. Each line contain m numbers, where the j -th number of the i -th line denotes H_{ij} ($1 \leq H_{ij} \leq 1000$). Finally, the next $m + 1$ lines describe the cost of north-south street segments. Each line contains n numbers, where the j -th number of the i -th line denotes V_{ij} ($1 \leq V_{ij} \leq 1000$).

Output

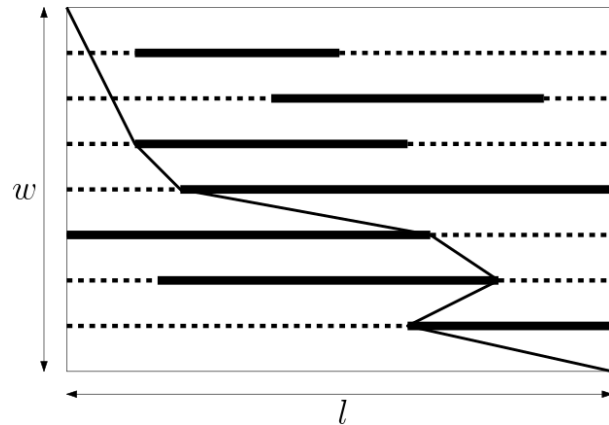
Print the profit of the most profitable plan. Note that the answer can be negative, zero, or positive.

Example

Standard Input	Standard Output
2 3 6 7 3 12 10 7 4 13 5 2 5 5 12 12 6 3 7 6 4 6 2 9 4	-28

Problem G : Gift Puzzle

Peyman’s birthday party is being held. Keivan, his old friend, has bought a puzzle as a birthday gift. The puzzle consists of a flat rectangular board with a length of l and a width of w , and a thread. The board has n horizontal rails of length l placed at different distances from the top horizontal side. On each rail, there is one obstacle which can *slide freely* on the rail. An example of the board is depicted in the figure. The rails are illustrated by dotted lines, and the obstacles are illustrated by thick segments.



In order to solve the puzzle, one must connect the top-left corner of the board to the bottom-right corner using the supplied thread. The thread must be inside the board and cannot pass through obstacles. In the figure, one possible way to do the puzzle is shown. Since Keivan believes in Peyman’s ability to solve hard puzzles, he wants to give Peyman the shortest thread while it is still possible to connect the two corners. So, kindly help Keivan to find the desired length of the thread.

Input

The first line of the input contains three integers l, w ($2 \leq l, w \leq 10^9$), the length and the width of the board, and n ($1 \leq n \leq \min(100\,000, w - 1)$), the number of the rails. Each of the next n lines contains two integers y_i ($1 \leq y_i \leq w - 1$), indicating the distance between the i -th rail and the top horizontal side, and l_i ($1 \leq l_i \leq l - 1$), length of the obstacle on the i -th rail. Note that all y_i ’s are distinct. You may assume that all obstacles and the thread have a width of zero.

Output

In the only line of the output, print the minimum t for which it is possible to configure obstacles such that the top-left corner can be connected to the bottom-left corner using a thread of length t while avoiding obstacles. Your answer is considered to be correct if it has a relative error of at most 10^{-9} .

Example

Standard Input	Standard Output
<pre>5 6 5 1 1 2 3 3 3 4 1 5 4</pre>	7.848191962583

Problem H : Passport Control Gates

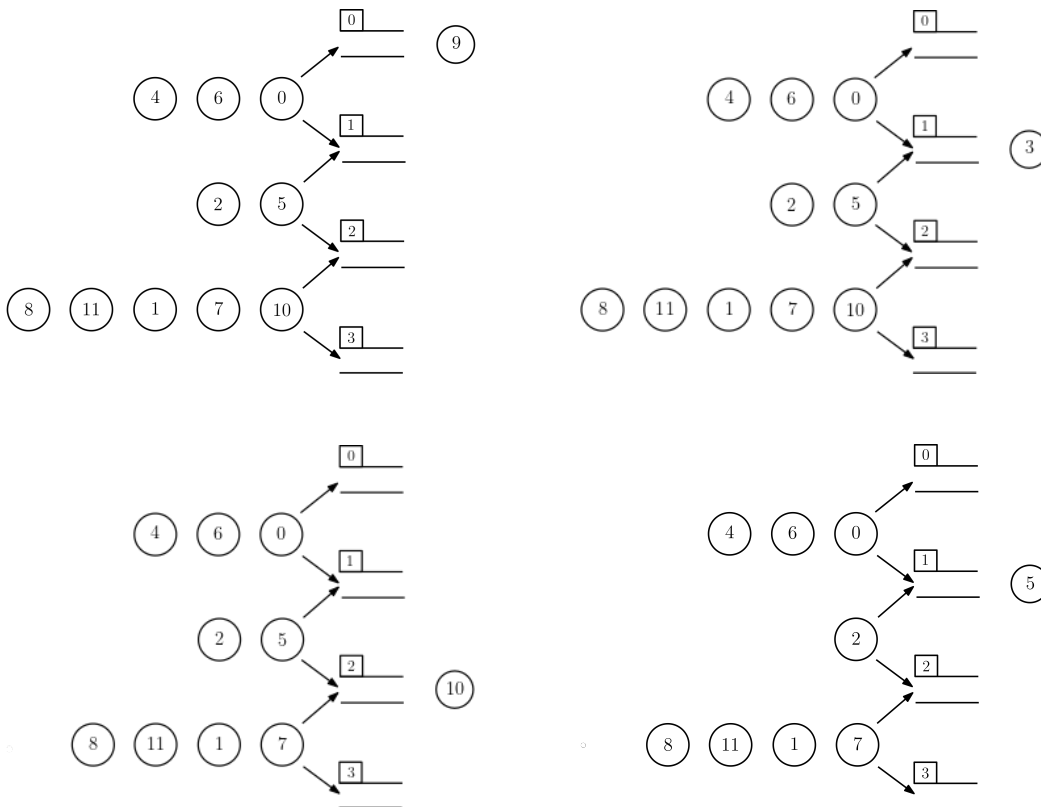
Everland and Neverland are two neighboring countries, and a huge number of Everlandian tourists visit Neverland every year. The governments want to analyze the passport control process in the border of Neverland and Everland, and need your help!

In the border, tourists stand in q queues for their passport check, and there are $q + 1$ passport control gates. If we number the gates from 0 to q and number the queues from 0 to $q - 1$, the tourists standing in queue i can only pass through gates i or $i + 1$.

Whenever gate i opens, if one of the queues i and $i - 1$ is empty or non-existent, the tourist at the front of the other queue passes through the gate. If both queues i and $i - 1$ are non-empty, the older tourist between the ones at the front of two queues passes through the gate. It is assumed that no two gates open at the same time.

We have a picture of n tourists standing in queues; waiting for the gates to open. Also, we have another picture that has been taken a while later, that some of the tourists from the first picture have passed through the gates. The tourists in the pictures are numbered from 0 to $n - 1$, in the order of their ages such that the person number 0 is the youngest and the person number $n - 1$ is the oldest. The picture below shows the first four configurations of the tourists in the first sample.

You are asked to find any valid sequence of gates' opening that might have happened between the times the two pictures were taken, or claim that it is impossible. A gate can be opened multiple times in the sequence.



2019 ICPC Asia Tehran Regional Contest

Input

The first line of the input contains two integers q ($1 \leq q \leq 100\,000$), the number of queues, and n ($0 \leq n \leq 100\,000$), the number of tourists in the first picture. The i -th line of the next following q lines describes queue $i - 1$ in the first picture. Each queue description starts with a number k ($0 \leq k \leq n$) that shows the size of the queue, followed by k integers that indicate the tourist numbers in that queue, from the back to the front. The tourist numbers are unique and non-negative integers less than n . In the next following q lines the description of the second picture appears in the same format.

Output

If there is no valid sequence, print *Impossible*. If there are valid sequences, output any of them in the following format. Print the length of the sequence in the first line and the sequence itself in the second line. In your sequence, every time any gate opens, there must be at least one tourist waiting for it.

Example

Standard Input	Standard Output
<pre>3 12 4 4 6 0 9 3 2 5 3 5 8 11 1 7 10 3 4 6 0 1 2 2 8 11</pre>	<pre>6 0 1 2 1 2 3</pre>

Standard Input	Standard Output
<pre>3 3 1 2 1 0 1 1 1 2 0 1 1</pre>	<pre>Impossible</pre>

Standard Input	Standard Output
<pre>1 2 2 0 1 2 1 0</pre>	<pre>Impossible</pre>

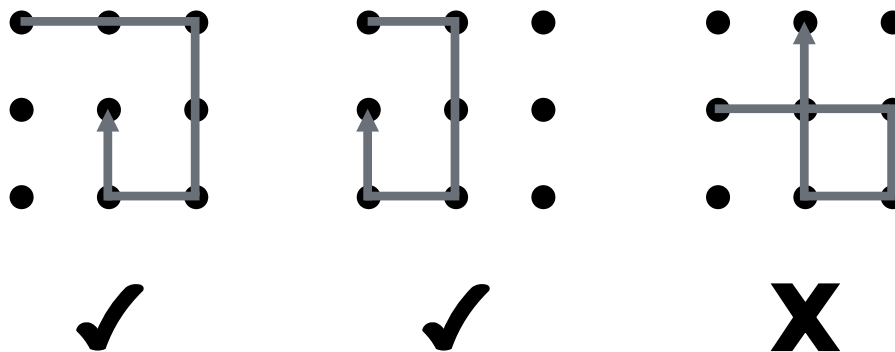
Standard Input	Standard Output
<pre>1 2 2 0 1 2 0 1</pre>	<pre>0</pre>

Problem I : Password

After working for several months at Cafebazaar, Farhad became rich enough to buy a house in the valley of the rich. There he met Shirin several times. Now, he is considering proposing to her whether she would marry him. To surprise her, he wants to install an application on her phone that pops up at the exact right time and asks if she would marry him.

However, to install the application secretly, he needs her password which he unfortunately does not have. He knows her password is a poly-line consisting of vertical or horizontal line segments. Each line segment connects the center of two cells in a 3×3 grid. Looking at her hand while she unlocked her phone, Farhad learned the direction of each line segment. However, he was too distracted to also learn the length of each segment. He also knows that her phone's operating system does not allow the poly-line to intersect with itself even in one point.

Farhad wants to distract Shirin long enough to try all possible patterns given what he already knows. Unfortunately, he has no idea how long that will take. That is why, he has now turned to you for help. Help him by writing a program that calculates the total number of possible password patterns given the direction of the line segments. The following figure depicts two valid and one invalid patterns given the line segments were directed towards right, down, left, and up in order.



Input

In the only line of the input, a single string is given consisting of characters R, U, L, and D which represent a line segment toward right, up, left, and down, respectively. The length of this string is at most 10. Every two consecutive characters is guaranteed to be different.

Output

In the only line of the output, print the number of patterns satisfying Farhad's knowledge of the password. Note that this number might be zero.

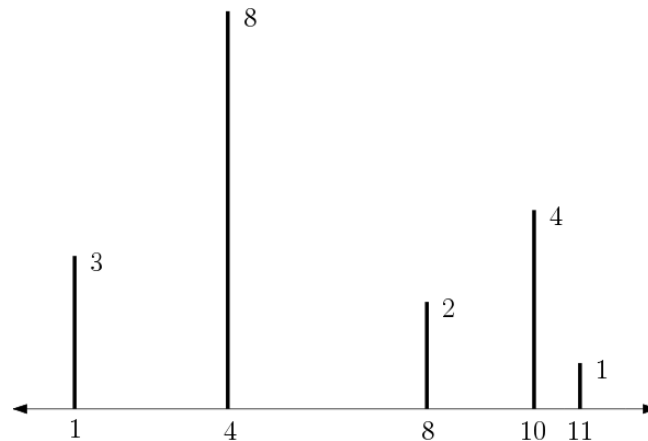
Example

Standard Input	Standard Output
DRU	15

Standard Input	Standard Output
R	9

Problem J : Greedy Termite

There are n wooden rods vertically placed over a horizontal line. The rods are numbered 1 through n from left to right. Each rod i ($1 \leq i \leq n$) is placed at position x_i and has a height h_i .



A termite wants to eat all the rods one by one. It starts eating from an arbitrary rod s ($1 \leq s \leq n$). Then, after eating a rod i , the termite selects the next rod to eat based on the following method. Among the remaining rods j , the one with maximum $h_j - |x_i - x_j|$ is selected. If there are ties, the one with minimum $|x_i - x_j|$ is selected. If there are still ties, the left-most rod is selected.

Your task is to calculate the total (horizontal) distance traveled by the termite to eat all the rods.

Input

The first line of the input contains two space-separated integers n , the number of rods, and s , the starting rod number ($1 \leq s \leq n \leq 100\,000$). The rods are described in the next n lines. On the line $1 + i$ ($1 \leq i \leq n$), the i -th rod is specified with two space-separated integers x_i ($|x_i| \leq 10^9$) and h_i ($1 \leq h_i \leq 10^9$). Additionally, for each i ($1 \leq i \leq n - 1$), $x_i < x_{i+1}$.

Output

You should print a single integer denoting the total distance traveled by the termite.

Example

Standard Input	Standard Output
<pre>5 3 1 3 4 8 8 2 10 4 11 1</pre>	17

Problem K : Plan B

In CrisisLand, every now and then there is a crisis that seriously threatens the security of the country. In the recent crisis, a series of civil protests occurred in multiple cities across the country. The protest started at a city and quickly spread out to other cities. To prevent this from happening again, the government as Plan B –after shutting down the Internet across the country– has decided to quickly send his forces to surround the city where the protest starts from. A city is surrounded if there are forces in each of its neighboring cities. The government has military bases in b different cities, each having many forces to be sent to all cities. The government knows his forces can not pass through the city from which the protest starts as they may be killed. Knowing this, it may be a case that some cities are not possible to be surrounded by forces. These cities are called *critical*. It is assumed that if there is a military base in a city, that city is not critical. Now, the government is eager to know whether there are critical cities in the country or not. As a legionnaire geek, help the government find his answer.

Oh, we forgot to explain the structure of CrisisLand! To resolve this *crisis*, we should mention that CrisisLand consists of n cities numbered from 1 to n . The cities are connected via m roads which can be used in both directions. Two cities are neighbors if there is a road between them. It is guaranteed that the road network of CrisisLand is connected.

Input

The first line of the input contains three positive integers n , m , and b denoting the number of cities, roads, and military bases, respectively ($1 \leq b \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$). Each of the next m lines contains two numbers v_i and u_i denoting a road between cities v_i and u_i . The last line consists of b integers, the cities having a military base.

Output

The output consists of two lines. The first line contains the number of critical cities. The second line contains the critical cities in ascending order.

Example

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

Problem L : The Assembly Code

Jamshid is working with a special computing machine. The machine only supports operations on 32-bit unsigned integers and its memory is an array M of length 2^{32} (for $0 \leq i < 2^{32}$, M_i is a 32-bit unsigned integer).

The assembly code for the machine supports the following addressing modes:

- Immediate addressing: $\# \langle number \rangle$
This type of addressing is used to refer constant values. So, $\#3$ means the constant value 3.
- Direct addressing: $\$ \langle index \rangle$
This mode is used to directly address a memory cell. For example, $\$7$ refers to cell M_7 .
- Indirect addressing: $@ \langle index \rangle$
This addressing mode is used for supporting pointers. It first looks up the value of the memory cell specified by $\langle index \rangle$, and then refers to the cell specified by that value. For example, if $M_5 = 9$, then $@5$ refers to M_9 .

All the memory cells are initialized with 0 when a program starts running.

An assembly code for the machine is a sequence of commands. Each command consists of an operation followed by a number of addresses. The current assembly language for the machine supports a limited number of operations:

- **MOVE** $\langle dest \rangle \langle source \rangle$: It copies the value of $\langle source \rangle$ to the memory cell referred by $\langle dest \rangle$. For example, **MOVE** $\$9 \#13$ sets M_9 to 13, and **MOVE** $@4 \$6$ sets M_{M_4} to the value of M_6 .
- **INPUT** $\langle address \rangle$: It reads a number from the input and stores it in the memory cell referred by $\langle address \rangle$. For example, **INPUT** $\$2$ stores the input value in M_2 , and **INPUT** $@1$ stores the input value in M_{M_1} .
- **OUTPUT** $\langle address \rangle$: It prints the value of $\langle address \rangle$ to the output.
- **ADD** $\langle dest \rangle \langle arg1 \rangle \langle arg2 \rangle$: It puts the sum of the values referred by $\langle arg1 \rangle$ and $\langle arg2 \rangle$ to the memory cell specified by $\langle dest \rangle$. In case of arithmetic overflow, the remainder of the result modulo 2^{32} is stored in the destination. For example, **ADD** $@10 \#4294967290 \#10$ sets $M_{M_{10}}$ to 4, and **ADD** $\$20 @8 \9 sets M_{20} to $M_{M_8} + M_9$.
- **MULT** $\langle dest \rangle \langle arg1 \rangle \langle arg2 \rangle$: It performs the multiplication similar to the **ADD** operation. It has the same behavior in case of arithmetic overflow.
- **AND** $\langle dest \rangle \langle arg1 \rangle \langle arg2 \rangle$: It puts the bit-wise AND of the values referred by $\langle arg1 \rangle$ and $\langle arg2 \rangle$ to the memory cell specified by $\langle dest \rangle$. For example, **AND** $\$15 \$33 \#7$ puts the remainder of M_{33} modulo 8 in M_{15} .
- **OR** $\langle dest \rangle \langle arg1 \rangle \langle arg2 \rangle$: It applies the bit-wise OR similar to the **AND** operation. For example, **OR** $\$121 \$121 \#1$ increments M_{121} if it is even.
- **XOR** $\langle dest \rangle \langle arg1 \rangle \langle arg2 \rangle$: It applies the bit-wise XOR similar to the **AND** operation. For example, **XOR** $@11 \#52 \#37$ sets $M_{M_{11}}$ to 17.

Except the **OUTPUT** operation, the first address given to all the operations must be direct or indirect addresses. Using the above operations, Jamshid wrote an assembly code for the machine. The code reads some numbers from the input and writes a single integer to the output (there is exactly one **OUTPUT** command in the program). Jamshid has executed the program with k different sets of inputs and saved the results. Later, he ran a formatting script on his code, but due to a bug in the script, some parts of the assembly program became corrupt. More

2019 ICPC Asia Tehran Regional Contest

specifically, the 5 arithmetic/bit-wise operations (ADD, MULT, AND, OR, and XOR) were replaced by 5 distinct ASCII characters A, B, C, D, E. The problem is that it's not clear which operation is represented by each ASCII character. Given the corrupted program together with the k input sets and their results, your job is to help Jamshid find the correspondence of the 5 assembly operations to the 5 ASCII characters.

Input

The input starts with the corrupted assembly program. Each line of the program contains a single command as specified before. The program contains at most 100 commands. It is guaranteed that the last command is the only output operation of the program. The next line contains the single integer k ($1 \leq k \leq 100$). Each of the next k lines is a space-separated sequence of integers specifying an execution log of the program. It is the sequence of input numbers given to the program appended by the program output. All numbers in the input are non-negative integers less than 2^{32} .

Output

You should print a single integer in the first line of output denoting the different number of ways to assign the 5 assembly operations to the 5 ASCII characters. The result may be any number between 1 and 120. If the result is unique, you have to print the correspondence in the second line. It should be printed as a space-separated permutation of the operators ADD, MULT, AND, OR, and XOR, in the order which they are respectively replaced by the ASCII characters A, B, C, D, E.

Example

Standard Input	Standard Output
<pre> INPUT \$2 A \$5 \$2 #3 INPUT \$1 MOVE \$3 #20 INPUT @3 B \$4 #1 \$5 E \$7 \$1 \$3 B \$8 #20 #9 D \$8 \$4 \$8 E \$10 \$7 \$8 OUTPUT \$10 3 3 8 9 29 1 0 0 21 4 2 3 30 </pre>	<pre> 1 XOR ADD MULT AND OR </pre>
Standard Input	Standard Output
<pre> INPUT \$0 OUTPUT \$0 2 0 0 1 1 </pre>	<pre> 120 </pre>