Problem A. Podracing

Time limit: 1 second
Memory limit: 64 MB

Padme: These junk dealers must have a weakness of some kind.
Shmi: Gambling. Everything here revolves around betting on those awful races.
Qui-Gon: Podracing. Greed can be a powerful ally.

The ship of queen Amidala was hit in a battle and had to land on a hot planet called Tatooine. Qui-Gon Jinn went to a city to get some spare parts and fix the ship. But in order to buy the details, one needs the local currency and the Jedi didn't have it.

The Jedi entered the shop of spare parts and met a 9-year-old boy Anakin Skywalker. From the day he was born, he'd been living on this planet together with his mother Shmi Skywalker. Anakin long wanted to take part in dangerous podracing, and he even assembled a podracer. Qui-Gon decided that he would help the boy to take part in the competitions. He also agreed with the slave owner Watto that if Anakin wins, the Jedi get spare parts, and the boy gets freedom. All you need to do is to determine the chances that Anakin wins.

The race track goes along the Oy axis and is bounded by two polylines on the left and on the right. The podracers are horizontal segments. They can move in any direction and even touch the road border, but they should always stay horizontal and be between the polylines.

The track is quite difficult as it has spots with cameras, and one is not allowed to hit them whatever happens. The cameras are material points and can be located on the track as well as outside it. If a podracer touches a camera with one of its ends, it goes unnoticed for the podracer and for the camera. Any other contact is not allowed.

The beginning of the track corresponds to the minimum y-coordinate of the vertices of the polylines, the finish line is at the maximum y-coordinate. Initially the podracer can be placed anywhere at the start level. A podracer finishes the race if it reaches the finish line. Help Qui-Gon to determine the maximum possible width of the podracer that is able to successfully finish.

Input

The first line contains integer n — the number of vertices in the left polyline. The i-th of the next n lines contains pairs of integers x_i and y_i — the coordinates of i-th vertex. Then follows number m and m more lines with the coordinates of the right polyline. Then goes integer q and q lines with the coordinates of the cameras ($2 \le n, m, q \le 10^5$). All the coordinates are integers, their absolute value is at most 10^9. The coordinates of vertices in each polyline are given in the increasing order of y-coordinate, the coordinates of the cameras are given in the non-decreasing order of y-coordinate. It is guaranteed that the polylines do not intersect or touch. You may assume that the minimal y-coordinates of both polylines coincide as well as the maximal ones.

Output

Print the answer with the absolute or relative error 10^{-6}.

Example

input	output
2 0 0 5 10 3 5 0 10 5 10 10 1 6 5	4.0

Problem B. The battle near the swamp

Time limit: 1 second
Memory limit: 64 MB

> Gungan: Jar Jar, usen da booma!
> Jar Jar: What? Mesa no have a booma!
> Gungan: Here. Taken dis one.

In the battle with the Trade Federation, Queen Amidala decided to ask gungans for help. Jar Jar Binks escorted the Queen and her people to the holy place where they had an agreement. The gungans agreed to provide their army in order to get the droids of the Federation out from the capital. The gungan ruler Boss Nass was so grateful for uniting the nations that he appointed Jar Jar a general.

And here they are: two armies lined up along the bank of the swamp. The droids of the Federation are well-disciplined soldiers. They stand in neat formation, divided into n blocks of k droids each. The gungans have a foolproof weapon against droids, which is small energy balls called boom booms. One such ball can disable exactly one droid.

Jar Jar Binks also decided to split his army into n parts and give each part a task to destroy the corresponding block of droids. Each part received a truck with boom booms. Now help general Binks calculate the number of boom booms that will be left unused and the number of droids that will survive the attack. You can assume that when a boom boom is fired at a droid by a gungan, it always hits the target.

Input

The first line of the input contains numbers n and k ($1 \le n, k \le 10\,000$). The second line contains n numbers a_i ($0 \le a_i \le 100\,000$) — the number of boom-booms in the i-th truck.

Output

Print two integers — the number of unused boom booms and the number of survived droids.

Example

input	output
4 5 2 7 5 0	2 8

Problem C. CVS

Time limit: 1 second
Memory limit: 64 MB

> Visit I will the cloners on Kamino... And see this army they have created for the Republic.
>
> Yoda

Cloners from the Kamino planet breed some of the finest clones. Such good results are due to the careful management over the clones' evolution. The Kaminuans are now busy working out a new study technology that lets increase the clones' effectiveness. The cloners have come up with a new control system CVS (Clone Version System) that makes managing the progress of experiments easier. The system is quite simple to use.

The Kaminuans have some set of educational programs at their disposal. A clone's effectiveness depends on which programs and in which order he has learned. The Kaminuans can teach any clone a program as long as this clone hasn't learned it already.

To make the experiments even easier to conduct, the Kaminuans enabled canceling the changes made by the last program the clone has learned. In this case, the clone's knowledge returns to the level when the program hasn't yet been studied. Then this clone can study this program in the future. You can cancel programs at any time unless the clone is at the basic knowledge level.

Besides the 'roll back' function, a clone can 're-learn' a program. If one cancels some program by mistake, he can cancel the cancellation. The CVS keeps record of each clone's canceled programs. After a program is canceled, the CVS adds another record. After a clone re-learn a program, the record is deleted. If a clone learn (not relearn) a program, all cancellation record history of this clone is deleted. You can use the re-learn function as long as the record history for this clone contains any records.

Finally, the system has a 'clone' option. If a Kaminuan likes the current variant of a clone, he can clone the clone, that is, create a new clone with the same sequence of taught programs and cancellation history.

Initially the Kaminuans have a single clone with basic knowledge. Help them analyze the progress of the experiments.

Input

The first line of the input contains numbers n — the number of queries — and m — the number of educational programs ($1 \le n, m \le 5 \cdot 10^5$). Each of the following n lines has one of the formats given below.

- *learn* c_i p_i. Teach clone c_i program p_i ($1 \le p_i \le m$).
- *rollback* c_i. Cancel the last learned program for clone c_i.
- *relearn* c_i. Apply 're-learn' function to clone c_i.
- *clone* c_i. Clone the clone c_i.
- *check* c_i. Display the last program clone c_i has learned and knows at the moment.

It is guaranteed that *rollback* won't be applied to the clone that is at the basic knowledge level. *learn* is always applied with the program a clone doesn't already know. *relearn* is only applied if the cancellation history of a clone is not empty. It is also guaranteed that only the clones that already exist can occur in the queries. The numbers are assigned to the clones in the order the clones appear. The Kaminuans started their experiments from clone number one.

Output

For each *check* c_i query display the result on a single line. If some clone has only basic knowledge, print *basic*, otherwise print the number of the last learned program.

Example

input	output
9 10 learn 1 5 learn 1 7 rollback 1 check 1 clone 1 relearn 2 check 2 rollback 1 check 1	5 7 basic

Problem D. This cheeseburger you don't need

Time limit: 1 second
Memory limit: 64 MB

May the Force be with you.

Yoda

Master Yoda is the oldest member of the Jedi Council. He conducts preparatory classes of little Younglings up to the moment they get a mentor. All Younglings adore master Yoda and they hope to grow as strong and wise as he is. Just like all little children, Younglings are absolutely hooked on new games and ideas. Now they decided to learn to speak just like master Yoda. Help the Younglings understand how Yoda would say this or that sentence.

Yoda is speaking the Galactic language using the specific word order — so-called "object-subject-verb".

Your program receives a sentence that interests the Younglings. They have already highlighted all important parts in the sentence. They use the curly {}-brackets for objects, round ()-brackets for subjects and square []-brackets for verbs.

A sentence in the input can be simple or complex. If the sentence is complex, then it consists of two simple sentences separated by a comma. Sometimes a comma is followed by a conjunction that is not in the brackets.

Each simple question has exactly one object, one subject and one verb. Your task is to simply put them in the correct order. Namely, first the object, then the subject, finally the verb. Also, please do not forget that only the first word in the whole sentence should begin with capital letter.

Input

The single line contains a sentence that interests the Younglings. The length of the sentence does not exceed 100 characters. All the words in the sentence consist of Latin letters. The first letter of the first word is capitalized and the rest are small. The sentence may contain a comma. Each simple sentence contains all three types of brackets. Each pair of brackets surrounds one or more words. No pair of brackets can surround the other bracket. Brackets are always located on the borders of words. The words in the sentence are separated by a single space. There is no space character before a comma or a closing bracket and also after an opening bracket. The conjunction (which can be only after a comma) is the only word that is not surrounded by a pair of brackets.

Output

Print the sentence with the word order Yoda would use. All brackets must be omitted. You should separate the words by a single space.

Examples

input
(We) [are] {blind}, if (we) [could not see] {creation of this clone army}
output
Blind we are, if creation of this clone army we could not see

input
{Truly wonderful} (the mind of a child) [is]
output
Truly wonderful the mind of a child is

Problem E. The Emperor's plan

Time limit: 1 second
Memory limit: 64 MB

> Palpatine: In order to ensure the security and continuing stability the Republic will be reorganized into the first Galactic Empire! For a safe and secure society.
> Padme: So this is how liberty dies. With thunderous applause.

The Galactic Republic existed for more than twenty five thousand years and then it was transformed into the first Galactic Empire. The superior Councilor Palpatine ensured his power and position, issued the Declaration of the New Order and proclaimed himself the Galactic Emperor.

The highest body of the Republic was the Galactic Senate. Even after the Empire rose, many senators had a good standing in the society. The Senate prevented Emperor Palpatine from achieving his goals and was an obstacle on the way to power. The Emperor understood that he needed to act from the inside to seize full power. That's why he recruited some of the Senate members. These spies were very professional. During the day, they were just like ordinary senators and they served the Empire at night. In the evening, each spy got a task to remove one of the senators, devoted to the Old Order. The following night all the senators, who had been chosen to be removed, died. Palpatine wanted to get rid of the problem as quickly as possible, that's why he gave the orders in such a way that each night maximum possible quantity of the non-spy senators met their deaths.

Having faced the deaths, senators suspected the Emperor's spies among them. So it was decided to exclude members from the Senate until deaths stop or there remain only spies in the Senate. After excluding from the Senate a senator loses his standing in the society and is no longer a threat for Palpatine. On the other hand, if he was a spy then he won't be able to kill senators any more.

Senators suppose that during the night every spy kills exactly one non-spy senator. The spies don't give themselves away, that's why every time the Senate chooses only the quantity of to-be-excluded members, but personalities are chosen at random. The Senate acts optimally and chooses the quantity of to-be-excluded in order to maximize the expected number of the non-spy senators still acting by the end of the confrontation.

At the moment k out of n Senate members are spies, and the Emperor has just given them new tasks. Help Palpatine count the expected number of senators devoted to the Old Order, which will remain in the Senate by the end of the confrontation.

Input

The single line of the input contains two integers, n and k ($1 \le k \le n \le 200, 1 \le k \le 20$).

Output

Print a single number — the answer to the problem with absolute or relative error at most 10^{-6}.

Example

input	output
4 1	0.6666666667

Problem F. Illegal spices

Time limit: 1 second
Memory limit: 64 MB

> Jabba: Han, my boy, you disappoint me. Why haven't you paid me? And why did you fry poor Greedo?
> Han: Look, Jabba, next time you wanna talk to me, come see me yourself. Don't send one of these twerps.
> Jabba: Han, I can't make exceptions. What if everyone who smuggled for me dropped their cargo at the first sign of an imperial starship?

Han Solo and his flight mechanic Chewbacca are experienced smugglers. They have spent several years working for a crime boss of the Tatooine planet Jabba the Hutt. But even the best of the best blow it sometimes.

During a flight, captain Solo's ship called «Millennium Falcon» met imperial customs officers. Han was transporting cargo consisting of n bags with spices. Each bag weighs an integer number of kilograms. Han noticed the customs officers from above and decided to dump the cargo into space.

Captain Solo dumped the first bag. Then he decided to take a risk and leave part of the cargo in the secret section. Han was checking each bag, and during that he was counting how many bags he has already checked (including the first one) that were lighter than the current one. He left a bag on the ship only if that number was at least p percent from the total number of the bags that have already been checked. Using this strategy Solo hoped to leave the most important part of the cargo on the ship.

As a result, he was left with only k bags that fit in the secret section easily. The customs officers couldn't find anything and they left the «Millennium Falcon».

Now Han understands that he has lost quite a large part of the cargo and that Jabba is going to be quite displeased. Unfortunately, he doesn't remember the total weight of the transported goods. Help Han find the minimum possible total weight just to show him how big a loser he is.

Input

The first line contains integers n and k ($1 \leqslant k < n \leqslant 10^5$). The second line contains integer p ($1 \leqslant p \leqslant 100$).

Output

In the first line, print the answer to the problem. In the second line print n space-separated integers — the bags' weights in kilograms in the order Captain Solo checked them, including the first bag. If there are multiple sequences that meet the problem statement, you can print any of them. It is guaranteed that at least one such sequence exists.

Example

input	output
3 1 50	4 1 2 1

Problem G. Cipher Message 3

Time limit: 2 seconds
Memory limit: 64 MB

> I have placed information vital to the survival of the rebellion into the memory systems of this R2 unit. My father will know how to retrieve it. You must see this droid safely delivered to him on Alderaan.
>
> Leia

Emperor Palpatine has been ruling the Empire for 25 years and Darth Vader has been the head of the Empire Armed Forces. However, the Rebel movement is strong like it never used to be. One of the rebel leaders, Princess Leia from Alderaan, managed to get hold of secret blueprints of the Death Star, the imperial war station.

The Princess was going to deliver the station plan to the secret base for further analysis and searching for vulnerable spots. But her ship was attacked by the space destroyer "Devastator"headed by Darth Vader. At the last moment Princess Leia managed to send her findings to one of the closest planet called Tatooine with her droid R2-D2. Quite conveniently, an old friend of her father Obi-Wan Kenobi lives on that planet.

R2-D2 realizes the importance of his mission. He is going to encrypt the information so that the wrong people won't get it.

The memory of R2-D2 has many files with images. First he wanted to use a well-known encrypting algorithm. The point of the method is to replace the least significant bits of the image with the encrypted message bits. The difference is practically unnoticeable on the picture, so one won't suspect that it contains a hidden message.

But then R2-D2 decided that this method is quite well-known and the information won't be protected enough. He decided to change the least significant bits of the image so that the secret information was a continuous sequence of the bytes of the image file. Help the droid determine if it is possible. And if it is, find the minimum number of bits to alter.

Input

The first line of the input contains integers n and m ($1 \le n, m \le 250\,000$) — the sizes of the image file and of the file with the secret information in bytes. On the second line the content of the file with an image is given and the third line contains the secret information. The files are given as a sequence of space-separated bytes. Each byte is written as a sequence of eight bits in the order from the most to the least significant bit.

Output

Print `"No"`, if it is impossible to encrypt information in this image. Otherwise, print in the first line `"Yes"`, and in the second line — the number of bits to alter and the number of the byte in the file with the image, starting from which the secret information will be recorded. If there are multiple possible variants, print the one where the secret information is written closer to the beginning of the image file.

Examples

input	output
3 2 11110001 11110001 11110000 11110000 11110000	Yes 1 2
3 1 11110000 11110001 11110000 11110000	Yes 0 1

Problem H. Those are not the droids you're looking for

Time limit: 1 second
Memory limit: 64 MB

> Bar owner: Hey. We don't serve their kind here.
> Luke: What?
> Bar owner: Your droids – they'll have to wait outside. We don't want them here.

Planet Tatooine is quite far from the center of the Galaxy. It is at the intersection of many hyperspace paths and it hosts smugglers and hoods of all sorts. The pilots who visited external territories have been to the space port bar called Mos Eisley for a drink at least once.

In this bar you can find a bunch of rascals and scoundrels from all over the Galaxy. The bar owner is ready to make drinks for any client except for, perhaps, a droid. Usually the bar has a lot of smugglers hanging out there. Each smuggler spends at least a minutes inside hoping to meet a good client. Cargo owners show up quite often as well. They usually find a dealer quickly, so they never spend more than b minutes in the bar.

The imperial stormtroopers are searching through Tatooine for the escaped droids. The bar owner said that no droids had ever been on his territory. He also said that nobody except for smugglers and cargo owners had been in the place recently.

Help the stormtroopers find out if the owner is a liar. For that, you are going to need the daily records from the sensor on the entrance door. The sensor keeps record of the time when somebody entered the bar or left it. The stormtroopers got the records after the bar had been closed, so there was nobody in the bar before or after the sensor took the records. You can assume that the sensor is working properly. That is, if somebody went through the bar door, the sensor made a record of that. You can also assume that the bar clients go in and out only through the door with the sensor. But the bar owner and the staff use the 'staff only' door.

Input

The first line of the input contains integers a and b ($1 \le a, b \le 10^9, b + 1 < a$). The second line contains integer n — the number of records from the sensor ($2 \le n \le 1000$). The i-th of the next n lines contains two integers t_i and d_i ($1 \le t_i \le 10^9, d_i \in \{0, 1\}$) — the time of the i-th record and direction (0 — in the bar, 1 — out of the bar). The records in the input are listed in the increasing order of t_i.

Output

If there is no doubt that somebody who was neither a smuggler nor a cargo owner visited the bar, print `"Liar"` on a single line. Otherwise, print a line `"No reason"`. And in the following lines list information on all visitors of the bar. The information about a visitor consists of two space-separated numbers — the time this visitor entered the bar and the time he left the bar. If there are multiple solutions that correspond to the sensor records, print any of them.

Examples

input	output
6 3 4 1 0 2 0 5 1 10 1	No reason 1 10 2 5
6 3 4 1 0 2 0 6 1 10 1	Liar

Problem I. The old Padawan

Time limit: 0.5 second
Memory limit: 64 MB

Use the Force. Yes. Now, the stone. Feel it. Concentrate!

Yoda

Luke Skywalker is having exhausting practice at a God-forsaken planet Dagoba. One of his main difficulties is navigating cumbersome objects using the Power. Luke's task is to hold several stones in the air simultaneously. It takes complete concentration and attentiveness but the fellow keeps getting distracted.

Luke chose a certain order of stones and he lifts them, one by one, strictly following the order. Each second Luke raises a stone in the air. However, if he gets distracted during this second, he cannot lift the stone. Moreover, he drops some stones he had picked before. The stones fall in the order that is reverse to the order they were raised. They fall until the total weight of the fallen stones exceeds k kilograms or there are no more stones to fall down.

The task is considered complete at the moment when Luke gets all of the stones in the air. Luke is good at divination and he can foresee all moments he will get distracted at. Now he wants to understand how much time he is going to need to complete the exercise and move on.

Input

The first line contains three integers: n is the total number of stones, m is the number of moments when Luke gets distracted and k ($1 \leqslant n, m \leqslant 10^5$, $1 \leqslant k \leqslant 10^9$). Next n lines contain the stones' weights w_i (in kilograms) in the order Luke is going to raise them ($1 \leqslant w_i \leqslant 10^4$). Next m lines contain moments t_i, when Luke gets distracted by some events ($1 \leqslant t_i \leqslant 10^9$, $t_i < t_{i+1}$).

Output

Print a single integer — the number of seconds young Skywalker needs to complete the exercise.

Example

input	output
5 1 4 1 2 3 4 5 4	8

In the first three seconds Luke raises stones that weight 1, 2 and 3 kilograms. On the fourth second he gets distracted and drops stones that weight 2 and 3 kilograms. During the next four seconds he raises all the four stones off the ground and finishes the task.

Problem J. The secret module

Time limit: 1 second
Memory limit: 64 MB

> Palpatine: Rise, my friend.
> Darth Vader: The Death Star will be completed on schedule.
> Palpatine: You've done well, Lord Vader.

The Death Star space station was created to keep the whole Galaxy under control. This station was equipped with the most powerful weapons and could carry thousands destroyers. The Death Star also had some precautions introduced in order to prevent the rebels from approaching. Specifically, all destroyers that approached the station were subject to thorough examination. There was a special module responsible for managing the entrance control. The module read and checked the arriving ship's identification number.

The ship's number is a non-negative integer, it is read from the left to the right. The number can have any length. Leading zeroes should not affect the module verdict. The checking module is a finite deterministic automaton with n states. Initially, the automaton is at the first state. It checks a ship's number as follows. It reads a digit and changes its state according to some rule, determined by the current state and the analyzed digit. If the automaton reads the ship's number to the end and finishes in a terminal state, then the station lets the ship in. Otherwise, the ship is destroyed. Any ship with an empty identification number also has to be destroyed, hence the initial state is non-terminal for sure.

This system worked perfectly well until the day the rebels destroyed the station. The designers of the new Death Star decided to use a similar technology for checking ships' numbers. But the rebels already know the numbers of many enemy ships. So, in order to raise security levels, the identification numbers of all imperial destroyers were increased by some number k.

Your task is to make a special module for the new Death Star. The module should let a ship in if and only if its number is the new number of one of the imperial destroyers.

Input

The first line of the input contains integer k ($1 \le k \le 8$). Then the automaton's description follows. The first line of the description contains the number of the states in the automaton n ($1 \le n \le 100$). The second line of the description contains n numbers t_i ($t_i \in \{0, 1\}$). $t_i = 1$, if the i-th state is terminal and $t_i = 0$ otherwise. The i-th of the next n lines contains 10 integers that correspond to the numbers of states the automaton will change to if it is in state i and reads digit $0, 1, \ldots, 9$ correspondingly.

Output

If there is no proper automaton for the Death Star, print `"Impossible"`. If there is one, print `"Success"`, then print the automaton's description in the same format as in the input. Your automaton should have at most 25 600 states. It is guaranteed that if there is a proper automaton, then there also is one that fits into the given limit.

Example

input	output
3 4 0 0 1 0 1 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	Success 5 0 0 0 0 1 1 2 3 4 4 4 4 4 4 4 4 2 3 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4