

---

## Problem A. Potion of Immortality

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

The world famous scientist Innokentiy has just synthesized the potion of immortality. Unfortunately, he put the flask with this potion on the shelf where most dangerous poisons of all time were kept. Now there are  $n$  flasks on this shelf, and the scientist has no idea which of them contains the potion of immortality.

Fortunately, Innokentiy has an infinite amount of rabbits. But the scientist doesn't know how exactly these potions affect rabbits. There is the only thing he knows for sure. If rabbit tastes exactly  $k$  potions from the flasks on the shelf, it will survive if there was the immortality potion among them and die otherwise. If rabbit tastes the number of potions different from  $k$ , the result will be absolutely unpredictable, so the scientist won't make such experiments no matter what.

The scientist intends to minimize the amount of lost rabbits while searching for the potion of immortality. You should determine how many rabbits he has to sacrifice in the worst case.

### Input

The only line contains two integers separated by space —  $n$  and  $k$  ( $1 \leq n \leq 2000$ ,  $1 \leq k \leq n$ ) — the number of flasks on the Innokentiy's shelf and the number of potions Innokentiy will give to a single rabbit to taste.

### Output

If the scientist cannot determine which flask contains the potion of immortality, output «-1». Otherwise, output a single integer — the minimal number of rabbits that Innokentiy will sacrifice in the worst case while searching for the potion of immortality.

### Examples

<code>stdin</code>	<code>stdout</code>
3 2	1
4 2	2

---

## Problem B. A Lot of Joy

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Two boys Gena and Petya wrote on two strips of paper the same string  $s$  that consisted of lowercase Latin letters. Then each boy took one strip, cut it into small pieces so that each piece contained exactly one letter and put all pieces into his pocket. After that Gena and Petya repeated the following procedure until their pockets became empty: they took one piece from their pockets and rejoiced if the letters on these pieces were equal.

Determine the expected number of times the boys rejoiced during this process.

### Input

The input contains the only string  $s$  which was initially written on Gena's and Petya's strips. The string has length between 1 and 200000, inclusively, and consists of lowercase Latin letters.

### Output

Output the only real number — the expected number of times Gena and Petya rejoiced during their business. The absolute or relative error of the answer should not exceed  $10^{-9}$ .

### Examples

<code>stdin</code>	<code>stdout</code>
<code>abc</code>	<code>1.0000000000000000</code>
<code>zzz</code>	<code>3.0000000000000000</code>

---

## Problem C. Very Spacious Office

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Programmers at the company «Perimeter» are working on  $n$  software projects. Their boss Shiftman understands the importance of comfortable working conditions. There is neither dress code nor fixed work schedule in the company, but there always is tea and fresh kiwis in the kitchen. When the team of the «Diplodocus» project complained that their room was too crowded after new employees had joined the company, Shiftman understood that it was time to search for a new spacious office.

A new office building was found quickly. It was located near a subway station and a nice park. In addition, there was a large underground parking. The number of rooms in the office was the same as the number of projects in the company, so Shiftman decided to assign a room to each project, thus creating a unique work atmosphere for the teams. Project managers had their own notions of ideal room for their projects. Of course, the room should not be too small. However, if the room would be too big, the programmers might be afraid that a new team would be added to their room. Help the managers to assign the rooms quickly and without the boss's meddling.

### Input

The first line contains the number  $n$  of projects in the company ( $1 \leq n \leq 100000$ ). In the second line you are given  $n$  numbers, which are the areas of all the rooms in the new office. The  $i$ -th of the following  $n$  lines contains two numbers, which are the minimum and maximum areas of the room in which the team of the  $i$ -th project agrees to work (of course, the minimum area does not exceed the maximum area). All the areas are positive integers and do not exceed  $10^9$ .

### Output

If there is only one way to assign the rooms to the teams, output «Perfect!» in the first line and a permutation of integers from 1 to  $n$  in the second line. In this permutation, the  $i$ -th element must be the number of the room assigned to the team of the  $i$ -th project. The rooms are numbered from 1 to  $n$  in the order in which they are described in the input.

If there are several ways to assign the rooms, output «Ask Shiftman for help.»

If it is impossible to assign the rooms as required, output «Let's search for another office.»

### Examples

stdin	stdout
3 40 50 60 30 70 20 40 60 60	Perfect! 2 1 3
3 40 50 70 30 70 20 50 60 60	Let's search for another office.

---

## Problem D. Holidays

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Everyone knows that the battle of Endor is just a myth fabled by George Lucas for promotion of his movie. Actually, no battle of Endor has happened and the First Galactic Empire prospers to this day.

There are creatures of  $n$  races living in the First Galactic Empire. In order to demonstrate their freedom, equality and brotherhood the Emperor commanded to introduce the holidays. During each of these holidays creatures of one non-empty subset of races should give gifts to creatures of another non-empty subset of races, not intersecting the first one.

The Emperor's stuff is not very strong in maths so you should calculate how many such holidays can be introduced. Two holidays are considered different if they differ in the subset of races which give gifts or in the subset of races which receive gifts.

### Input

The input contains the only integer  $n$  ( $1 \leq n \leq 200000$ ) — the number of races living in the First Galactic Empire.

### Output

Find the number of holidays the Emperor commanded to introduce. This number can be very large, so output the remainder of division of this number by  $10^9 + 9$ .

### Examples

<code>stdin</code>	<code>stdout</code>
2	2
3	12
5	180
200000	82096552

---

## Problem E. Two Labyrinths

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

A labyrinth is the rectangular grid, each of the cells of which is either free or wall, and it's possible to move only between free cells sharing a side.

Constantine and Mike are the world leaders of composing the labyrinths. Each of them has just composed one labyrinth of size  $n \times m$ , and now they are blaming each other for the plagiarism. They consider that the plagiarism takes place if there exists such a path from the upper-left cell to the lower-right cell that is the shortest for both labyrinths. Resolve their conflict and say if the plagiarism took place.

### Input

In the first line two integers  $n$  and  $m$  ( $1 \leq n, m \leq 500$ ) are written — the height and the width of the labyrinths.

In the next  $n$  lines the labyrinth composed by Constantine is written. Each of these  $n$  lines consists of  $m$  characters. Each character is equal either to «#», which denotes a wall, or to «.», which denotes a free cell.

The next line is empty, and in the next  $n$  lines the labyrinth composed by Mike is written in the same format. It is guaranteed that the upper-left and the lower-right cells of both labyrinths are free.

### Output

Output «YES» if there exists such a path from the upper-left to the lower-right cell that is the shortest for both labyrinths. Otherwise output «NO».

### Examples

stdin	stdout
3 5 ..... .#.#. .....  ..... #.#.# .....	NO
3 5 ..... .#.## .....  ..... ##.#. .....	YES

---

## Problem F. Doomsday

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Doomsday comes in  $t$  units of time. In anticipation of such a significant event  $n$  people prepared  $m$  vaults in which, as they think, it will be possible to survive. But each vault can accommodate only  $k$  people and each person can pass only one unit of distance per one unit of time. Fortunately, all people and vaults are now on the straight line, so there is no confusion and calculations should be simple.

You are given the positions of the people and the vaults on the line. You are to find the maximal number of people who can hide in vaults and think they will survive.

### Input

The first line contains four integers  $n$ ,  $m$ ,  $k$  and  $t$  ( $1 \leq n, m, k \leq 200000$ ,  $1 \leq t \leq 10^9$ ) separated by spaces — the number of people, the number of vaults, the capacity of one vault and the time left to the Doomsday.

The second line contains  $n$  integers separated by spaces — the coordinates of the people on the line.

The third line contains  $m$  integers separated by spaces — the coordinates of the vaults on the line.

All the coordinates are between  $-10^9$  and  $10^9$ , inclusively.

### Output

Output one integer — the maximal number of people who can hide in vaults and think they will survive.

### Examples

<code>stdin</code>	<code>stdout</code>
2 2 1 5 45 55 40 60	2
2 2 1 5 45 54 40 60	1
2 2 2 5 45 35 40 60	2
3 3 1 5 40 45 45 45 50 50	3

---

## Problem G. Image Processing

Input file:            **stdin**  
Output file:           **stdout**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

At the time free of games Andrey processes images. Most of all he likes to process grayscale images. Each pixel on a grayscale image has a brightness which is the integer from 0 to  $10^9$ , inclusively. The closer the brightness to zero, the darker is this pixel; zero brightness corresponds to the black color and  $10^9$  — to the white. Andrey calls such pictures *colored*.

When processing colored picture, Andrey chooses an integer  $t$  from 0 to  $10^9$ , inclusively, and creates a new image consisting of only two colors: black and while (he denotes them as 0 and 1, correspondingly). He calls such images *monochrome*.

The principle of creating monochrome images is very simple: Andrey looks at each pixel of the initial colored image and, if the brightness of this pixel is strictly less that  $t$ , this pixel will be black on the monochrome image, otherwise it will be white.

You are given  $k$  monochrome images. You are to determine if they all could be the result of processing the same colored image, and if they could, which exactly colored image and values  $t_1, \dots, t_k$  could be used.

### Input

The first line contains three positive integers  $k$ ,  $n$  and  $m$  ( $1 \leq k \cdot n \cdot m \leq 200000$ ) — the amount of images and their sizes.

Then  $k$  blocks describing monochrome images follow. Each block starts from the empty line, followed by the description of the image:  $n$  lines of  $m$  characters each. These characters can be either «0», which denotes the black color, or «1», which corresponds to the white color.

### Output

If there is no such colored image that all  $k$  given monochrome images can be the result of its processing, output «IMPOSSIBLE».

Otherwise, in the first  $n$  lines output the matrix  $n \times m$ , each element of which is the integer from 0 to  $10^9$  and equals to the brightness of the corresponding pixel of the colored image. Numbers should be separated by spaces.

After the matrix in the next line output  $k$  numbers  $t_1, \dots, t_k$  ( $0 \leq t_i \leq 10^9$ ) used to obtain the corresponding monochrome images.

If there are several possible solutions, output any of them.

---

## Examples

stdin	stdout
2 3 3  011 000 110  111 100 111	2 4 5 2 1 0 4 7 2 4 2
2 3 3  011 000 110  100 101 001	IMPOSSIBLE



---

## Problem H. Mysterious Photos

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Everyone probably heard the rumours about the constellation of Bermuda Triangle: any person who looks to this constellation of three stars is said to disappear completely. Though, it's not clear who then spreads these rumours.

Recently two photos have been sent to the editorial office of the newspaper you work on. Each photo depicts three glowing points on the dark background. The note applied to the photos indicates that they are the photos of the constellation of the Bermuda Triangle.

Of course, the editors cannot check if it's true without the risk of the stuff. But at least it is possible to make sure that each photo depicts the same triple of stars. Remember that photos could be taken with the different zoom and rotation. They could also be taken with the help of a mirror in order to decrease the risk of triggering the curse.

As a regular programmer of the newspaper you have the task to determine if these photos can depict the same triple of stars.

### Input

The input consists of 6 lines. Each of the first three lines contains two integers separated by space — the coordinates of stars on the first photo. Each of the next three lines also contains two integers — the coordinates of stars on the second photo. All coordinates are between  $-10^4$  and  $10^4$ , inclusively. Stars on each photo don't coincide and don't lie on the same line.

### Output

Output «YES» if the photos can depict the same triple of stars and «NO» otherwise.

### Examples

stdin	stdout
0 0 0 2 1 0 0 0 0 4 2 0	YES
0 0 0 2 1 0 0 0 0 4 3 0	NO
5 5 7 6 6 3 1 0 0 1 1 1	YES

---

## Problem I. Derivative of Array

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Let us define the derivatives of array in the following way:

- Zeroth derivative of array  $a_1, \dots, a_n$  is defined as this array  $a_1, \dots, a_n$  itself.
- First derivative of array  $a_1, \dots, a_n$ , where  $n > 1$ , is defined as such an array  $a'_1, \dots, a'_{n-1}$ , that for all  $i = 1, \dots, n - 1$  the equality  $a'_i = a_{i+1} - a_i$  is satisfied.
- $k$ -th derivative of array  $a_1, \dots, a_n$ , where  $1 < k < n$ , is defined as  $(k - 1)$ -th derivative of array  $a'_1, \dots, a'_{n-1}$ .

You are given  $n$  numbers  $b_0, \dots, b_{n-1}$ , each equals either to zero or to one. You are to build an array of length  $n$  so that its  $k$ -th derivative is strictly increasing if  $b_k = 1$  and strictly decreasing if  $b_k = 0$ . Each element of array should be an integer from  $-10^9$  to  $10^9$ , inclusively. If it is impossible, output «IMPOSSIBLE».

### Input

The first line contains the only integer  $n$  ( $1 \leq n \leq 2000$ ) — the length of the array you are to build.

The second line contains  $n$  integers  $b_0, \dots, b_{n-1}$  ( $0 \leq b_i \leq 1$ ) separated by spaces. If  $b_k = 1$ ,  $k$ -th derivative of array should be strictly increasing, and if  $b_k = 0$  — strictly decreasing.

### Output

Output  $n$  integers  $a_1, \dots, a_n$  — the array which derivatives satisfy all the conditions specified by numbers  $b_0, \dots, b_{n-1}$ . The elements of the array should not exceed  $10^9$  by absolute value.

If there are several such arrays, you can output any of them.

If there is no such array or some of its elements are greater than  $10^9$  by absolute value, output «IMPOSSIBLE».

### Examples

stdin	stdout
3 1 1 1	1 2 5
3 0 0 1	-1 -2 -5

---

## Problem J. Deck Shuffling

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

The world famous scientist Innokentiy continues his innovative experiments with decks of cards. Now he has a deck of  $n$  cards and  $k$  shuffle machines to shuffle this deck. As we know,  $i$ -th shuffle machine is characterized by its own numbers  $p_{i,1}, p_{i,2}, \dots, p_{i,n}$  such that if one puts  $n$  cards numbered in the order  $1, 2, \dots, n$  into the machine and presses the button on it, cards will be shuffled forming the deck  $p_{i,1}, p_{i,2}, \dots, p_{i,n}$  where numbers  $p_{i,1}, p_{i,2}, \dots, p_{i,n}$  are the same numbers of cards but rearranged in some order.

At the beginning of the experiment the cards in the deck are ordered as  $a_1, a_2, \dots, a_n$ , i.e. the first position is occupied by the card with number  $a_1$ , the second position — by the card with number  $a_2$ , and so on. The scientist wants to transfer the card with number  $x$  to the first position. He can use all his shuffle machines as many times as he wants. You should determine if he can reach it.

### Input

In the first line the only positive integer  $n$  is written — the number of cards in the Innokentiy's deck.

The second line contains  $n$  distinct integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — the initial order of cards in the deck.

The third line contains the only positive integer  $k$  — the number of shuffle machines Innokentiy has.

Each of the next  $k$  lines contains  $n$  distinct integers  $p_{i,1}, p_{i,2}, \dots, p_{i,n}$  ( $1 \leq p_{i,j} \leq n$ ) characterizing the corresponding shuffle machine.

The last line contains the only integer  $x$  ( $1 \leq x \leq n$ ) — the number of card Innokentiy wants to transfer to the first position in the deck.

Numbers  $n$  and  $k$  satisfy the condition  $1 \leq n \cdot k \leq 200000$ .

### Output

Output «YES» if the scientist can transfer the card with number  $x$  to the first position in the deck, and «NO» otherwise.

### Examples

stdin	stdout
4 4 3 2 1 2 1 2 4 3 2 3 1 4 1	YES
4 4 3 2 1 2 1 2 4 3 2 1 3 4 1	NO

---

## Problem K. Perpetuum Mobile

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

The world famous scientist Innokentiy almost finished the creation of perpetuum mobile. Its main part is the energy generator which allows the other mobile's parts work. The generator consists of two long parallel plates with  $n$  lasers on one of them and  $n$  receivers on another. The generator is considered to be working if each laser emits the beam to some receiver such that exactly one beam is emitted to each receiver.

It is obvious that some beams emitted by distinct lasers can intersect. If two beams intersect each other, one joule of energy is released per second because of the interaction of the beams. So the more beams intersect, the more energy is released. Innokentiy noticed that if the energy generator releases exactly  $k$  joules per second, the perpetuum mobile will work up to 10 times longer. The scientist can direct any laser at any receiver, but he hasn't thought of such a construction that will give exactly the required amount of energy yet. You should help the scientist to tune up the generator.

### Input

The only line contains two integers  $n$  and  $k$  ( $1 \leq n \leq 200000$ ,  $0 \leq k \leq \frac{n(n-1)}{2}$ ) separated by space — the number of lasers in the energy generator and the power of the generator Innokentiy wants to reach.

### Output

Output  $n$  integers separated by spaces.  $i$ -th number should be equal to the number of receiver which the  $i$ -th laser should be directed at. Both lasers and receivers are numbered from 1 to  $n$ . It is guaranteed that the solution exists. If there are several solutions, you can output any of them.

### Examples

<code>stdin</code>	<code>stdout</code>
4 5	4 2 3 1
5 7	4 2 5 3 1
6 0	1 2 3 4 5 6

---

## Problem L. Ministry of Truth

Input file:            **stdin**  
Output file:           **stdout**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Andrey works in the Ministry of Truth. His work is changing articles in newspapers and magazines so that they praise the Party and Big Brother.

Recently Big Brother decided that it would be fine if all words in all articles are read equally both from right to left and from left to right. In his opinion it will greatly simplify the reading of articles — even if the word is read in the reverse order, its sense won't change.

Andrey spends one second to erase one letter from the word and write another one to its place. Only one word left to be changed, after that he will complete the plan and be able to go home. Of course, he wants to do it as soon as possible. But the truth is that he does not clearly understand right now which word he should get after the replacement. Help him to find it out.

### Input

Input contains a single line with the word that Andrey has to change. The word consists of lowercase Latin letters and its length is from 1 to 200000, inclusively.

### Output

Output the word which should be a result of Andrey's work. If there are several possible such words, output any of them.

### Examples

stdin	stdout
abccabd	abacaba
wasitadogoracati <del>a</del> e	wasitacaroracatisaw

---

## Problem M. Heaviside Function

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 256 megabytes

Heaviside function is defined as the piecewise constant function whose value is zero for negative argument and one for non-negative argument:

$$\theta(x) = \begin{cases} 0, & x < 0; \\ 1, & x \geq 0. \end{cases}$$

You are given the function  $f(x) = \theta(s_1x - a_1) + \theta(s_2x - a_2) + \dots + \theta(s_nx - a_n)$ , where  $s_i = \pm 1$ . Calculate its values for argument values  $x_1, x_2, \dots, x_m$ .

### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 200000$ ) — the number of the summands in the function.

Each of the next  $n$  lines contains two integers separated by space —  $s_i$  and  $a_i$  ( $s_i = \pm 1, -10^9 \leq a_i \leq 10^9$ ) — parameters of the  $i$ -th summand.

The next line contains a single integer  $m$  ( $1 \leq m \leq 200000$ ) — the number of the argument values you should calculate the value of the function for.

The last line contains  $m$  integers  $x_1, \dots, x_m$  ( $-10^9 \leq x_i \leq 10^9$ ) separated by spaces — the argument values themselves.

### Output

Output  $m$  lines.  $i$ -th line should contain the value of  $f(x_i)$ .

### Examples

stdin	stdout
6	0
1 3	3
-1 2	0
1 9	2
-1 2	1
1 7	3
-1 2	2
8	3
0 12 2 8 4 -3 7 9	