

Problem A. Area Illumination

Time limit: 1 second
Memory limit: 512 megabytes

A new rectangular square of size m by n meters has been built in the city. To illuminate the square, the mayor wants to order innovative lamps, each of which illuminates a square of size $k \times k$ meters, with sides parallel to the boundaries of the square.

The mayor does not want to spend the entire city budget, so he wants to buy as few lamps as possible to illuminate the entire square. Help him determine how many lamps he needs to buy.

Input

The input consists of three integers n , m , and k ($1 \leq n, m, k \leq 10^9$) — the length of the square, the width of the square, and the length of the side of the square illuminated by each lamp.

Output

Output the minimum number of lamps required to illuminate the entire square.

Examples

standard input	standard output
10 9 3	12
4 6 2	6

Problem B. Battleship

Time limit: 1 second
Memory limit: 512 megabytes

Let's consider the classic game of Battleship.

According to the rules, each player has a 10×10 field on which they must place 10 ships: one «Battleship» (occupies 1×4 cells), two «Cruisers» (size 1×3 cells each), three «Destroyers» (size 1×2 cells each), and four «Submarines» (size 1×1 cells each). The following conditions must be met:

- All ships must be placed on the field;
- Each ship must fit entirely within the grid;
- The set of cells occupied by each ship must form a rectangle of the corresponding size;
- Each ship must be oriented either vertically or horizontally;
- Any two cells occupied by different ships cannot coincide or touch each other by side or corner.

We will describe the placement of ships using a 10×10 table, where each element contains the symbol '#' if the corresponding cell is occupied by a ship, and '.' otherwise.

Your task is to determine, given a 10×10 field, whether it corresponds to a valid ship placement that follows the rules of Battleship.

Input

The input consists of 10 lines, separated by line breaks, with 10 characters each — the description of the field. It is guaranteed that each character in the grid is either '#' or '.'.

Output

Print «YES» if the grid described in the input corresponds to a valid ship placement in the game of Battleship, and «NO» otherwise.

Examples

standard input	standard output
<pre>###.....##. .#..#..... .#..... .#..##.... ####...#..#..#.. ##.....</pre>	YES
<pre>##..#.....##. .#..#...# .#..... .#..... ####...#..#..#.. ##.....#</pre>	NO

Problem C. Carpet Showcase

Time limit: 1 second
Memory limit: 256 megabytes

Rene is a designer of abstract carpets. The Rene's carpets are rectangles $n \times m$, where n and m — are natural numbers. Each carpet is divided into $n \times m$ identical unit squares.

Rene designs a carpet showcase at a furniture exhibition. The showcase is a horizontal rectangular area with sides $h \times w$, where h and w are natural numbers; the floor on the showcase is also divided into unit squares.

Renee wants the carpets to be displayed on the showcase neatly.

Rene believes that carpets are laid out neatly if the following rules are followed. Each carpet can lie on the floor or on another carpet, and each unit square of the carpet must be strictly above the unit square of the floor or carpet underneath it. Moreover, if one carpet lies on another carpet, then it must be entirely inside it and have a strictly smaller area.

Rene wants to display carpets with the maximum total area at the exhibition. Help him calculate the maximum total area of carpets that can be present at the exhibition and can be neatly laid out on the showcase.

Input

The only input line contains two natural numbers h and w — the sides of the area for the showcase ($1 \leq h, w \leq 10^6$).

Output

Print a single integer — what is the maximum total area of carpets that can be neatly laid out on the showcase.

Examples

standard input	standard output
1 2	4
2 2	12
3 2	22

Problem D. Redrawn graph

Time limit: 1 second
Memory limit: 512 megabytes

During a very boring math lesson, Masha drew an undirected graph in her notebook with n vertices and m edges. The vertices of the graph were numbered from 1 to n .

After the lesson, she went on a break, and when she returned, she found that the graph seemed to have changed!

Masha's desk mate, Pasha, admitted that he had made some changes to the graph while Masha was away. Specifically, he performed the following operation: he took three distinct vertices of the graph, a , b , and c , and for each pair of vertices (a, b) , (b, c) , and (a, c) , he modified the corresponding edge: if it existed in the current graph, he removed it, and if it didn't exist, he added it.

He performed this operation not just once, but several times (possibly zero or one), each time choosing a , b , and c again.

Masha wants to verify her neighbor's words, so she asks you to find any sequence of Pasha's actions that could have transformed the original graph into the one Masha has in her notebook originally.

Input

The first line contains three numbers, n , m , and k ($3 \leq n \leq 10^5$; $0 \leq m, k \leq 10^5$), which represent the number of vertices, the number of edges in the original graph, and the number of edges in the resulting graph, respectively.

Each of the next m lines contains two numbers, u_i and v_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$), indicating the vertices connected by the i -th edge in the original graph.

The following k lines describe the resulting graph in the same format. It is guaranteed that both graphs do not contain loops or multiple edges.

Output

If it is impossible to obtain the second graph from the first one using Pasha's actions, output «NO» in a single line.

Otherwise, output «YES» in the first line. In the next line, output the number of operations t ($0 \leq t \leq 2 \cdot 10^5$) that Pasha had to perform. In each of the next t lines, output three numbers a_i , b_i , c_i ($1 \leq a_i, b_i, c_i \leq n$, the numbers a_i , b_i , c_i should be pairwise distinct), indicating the vertices that Pasha had to choose on the i -th step.

Note that you do not need to minimize the number of Pasha's actions, but it should not exceed $2 \cdot 10^5$. It can be shown that if a solution exists, then there exists a solution consisting of no more than $2 \cdot 10^5$ actions.

Examples

standard input	standard output
3 0 3 1 2 2 3 3 1	YES 1 1 2 3
4 3 3 1 2 2 3 3 4 1 3 4 2 2 3	YES 2 1 3 4 1 2 4
3 1 1 1 2 2 3	NO

Problem E. Accounting Chaos

Time limit: 1 second
Memory limit: 512 megabytes

There is chaos in the accounting department of a grand hotel: one of the guests, Sergey, has disappeared and cannot be reached, and his bill for the stay has not been paid!

The accountants of the grand hotel take their responsibilities very seriously, so a separate journal is created for each visitor, which has 2 columns: “service” and “cost”. In the “service” column, all the amenities provided by the hotel are recorded every day: breakfast in bed, taxi, stationery, etc., including the cost of the room for each day. In the “cost” column, the cost of each service is written in the local currency. The cost of the same service, including the room, does not change during the guest’s stay.

Unfortunately, due to the confusion caused by Sergey’s disappearance, one of the hotel staff accidentally spilled ink on the first column of the journal. Based on the remaining information, it is known that Sergey stayed in the grand hotel for n days, and the costs of m services provided to him during these days are known.

Since Sergey is a very important guest, he had his own non-standard rate for the room, but no one remembers what it was. The accounting department now wants to know what could be the cost of Sergey’s stay in the hotel per day.

Input

The first line contains two integers n, m ($1 \leq n \leq m \leq 3 \cdot 10^5$) — the number of days Sergey stayed in the hotel and the number of service records.

The second line contains m integers c_1, c_2, \dots, c_m ($1 \leq c_i \leq 10^9$) — the costs of services for all the days.

Output

In the first line, output a single integer k — the number of possible options for the cost of the room per day. It is guaranteed that $k \neq 0$.

In the second line, output k numbers - the possible options for the cost in any order.

Example

standard input	standard output
2 10	3
1 3 6 5 3 2 4 5 3 2	3 5 2

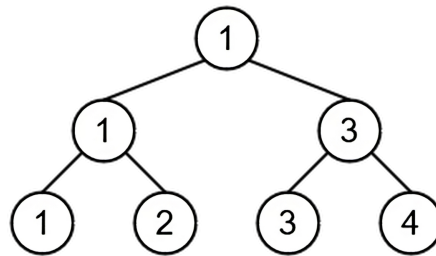
Problem F. Segment Tree

Time limit: 1 second
Memory limit: 512 megabytes

Lisa loves segment trees very much. Today, she found an array of distinct natural numbers of length 2^k at home and immediately built a segment tree with the minimum operation.

A segment tree with the minimum operation is a binary tree where each node has a left and right child. The leaves of the tree, when considered from left to right, contain the elements of the original array. The values of other nodes are calculated as the minimum value of their children's values.

For example, if the original array is $[1, 2, 3, 4]$, then the segment tree with the minimum operation will look like:



Lisa asked her friend Masha to draw the constructed tree on a piece of paper, and Masha did it. The next day, Lisa tried to remember what the original array looked like, but when she found the piece of paper, she was surprised to discover that Masha simply wrote the values in the nodes of the tree in a random order.

Help Lisa restore the original array or determine that Masha incorrectly wrote the values.

Input

The first line contains a single number n — the number of elements of the segment tree written on the piece of paper ($1 \leq n \leq 2 \cdot 10^5$). It is guaranteed that there exists an integer k such that $n = 2^{k+1} - 1$.

The second line contains n numbers a_1, a_2, \dots, a_n — the elements of the segment tree written on the piece of paper ($1 \leq a_i \leq 10^9$).

Output

Print -1 if it is impossible to restore the segment tree with the minimum operation from the numbers written on the piece of paper, which was built from an array of distinct natural numbers. Otherwise, print the original array. If there are multiple solutions, any of them can be printed.

Examples

standard input	standard output
7 2 3 1 1 4 1 3	1 2 3 4
3 1 2 3	-1
3 1 1 1	-1

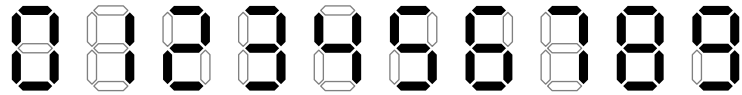
Note

In the third test example from the set of numbers, it is possible to build a segment tree with the minimum operation. However, in the original array, the numbers will not be distinct in this case.

Problem G. Elevator Ride

Time limit: 1 second
Memory limit: 512 megabytes

Every time Katya enters the elevator, she sees the reflection of the floor number in the mirror. The floor number is displayed on a panel, which shows the digits as shown in the picture.



Because the floor number is visible in the mirror, its reflection is reversed with respect to the vertical axis: the order of the digits is reversed and the digits appear as their reflections. However, if a digit x after reflection completely coincides with another digit y , the brain perceives it as y . But if it does not match any other digit, despite being reflected, the brain perceives x as x .

What number will Katya see when she enters the elevator, if it is known that she lives on the k -th floor.

Input

Given an integer k ($1 \leq k \leq 10^{18}$) — the floor where Katya lives.

Output

Output a single number that Katya will see in the mirror, without leading zeros.

Examples

standard input	standard output
13	31
250	25
1234567890	987624351

Problem H. Yurik and Important Tasks

Time limit: 1 second
Memory limit: 512 megabytes

Yurik is a very busy person, so he doesn't have time to complete all the necessary tasks on time. By the end of the week, he has accumulated n unfinished tasks. For convenience, let's number them using integers from 1 to n . At first, Yurik decided to do the tasks in the order of their numbering. However, he quickly realized that some tasks are more important than others, so he decided to change the order of their execution.

Yurik has a favorite *permutation* p_1, p_2, \dots, p_n , and he decided to use it to change the order of task execution. Besides being very busy, Yurik is also very fickle, so he decided to change the order of tasks q times.

Initially, Yurik wrote down the tasks in the order he would perform them on a piece of paper: $1, 2, 3, \dots, n$. After that he will choose two numbers l and r ($1 \leq l \leq r \leq n$) q times, and then change the order of the tasks that are currently in the list at positions from l to r .

The change in the order of task execution happens as follows. First, Yurik assigns *priorities* to the tasks that are in the list at positions from l to r . The task at position l will be assigned priority p_1 , the task at position $l + 1$ will be assigned priority p_2 , and so on. Thus, the task at position r will be assigned priority p_{r-l+1} . After that Yurik rearranges the tasks in **ascending** order of their priorities. For a better understanding of the process, pay attention to the explanation in the examples.

After Yurik changes the order of task execution q times, he gets confused and now wants to know which task he will perform as the k -th in order. Help him figure this out.

Recall that a permutation p_1, p_2, \dots, p_n is an array of pairwise distinct integers from 1 to n .

Input

The first line contains three integers n , q , and k ($1 \leq n \leq 100\,000$; $1 \leq q \leq 100\,000$; $1 \leq k \leq n$).

The second line contains n integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — Yurik's favorite permutation.

Each of the following q lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq n$) — the start and end of the range of tasks that Yurik will change the order of in the i -th time.

Output

Print a single integer t ($1 \leq t \leq n$) — the number of the task that Yurik will perform as the k -th in order after all the changes in the order of task execution.

Example

standard input	standard output
8 6 3 1 5 2 8 7 4 3 6 1 8 2 4 7 8 1 1 1 3 2 7	7

Note

Let's consider the first example. Initially, Yurik planned to perform the tasks in the following order: 1, 2, 3, 4, 5, 6, 7, 8. However, he decided to change the order of task execution 6 times.

In the first time, Yurik chose the tasks that are in the list at positions from 1 to 8 (i.e., all the tasks he had). The first task was assigned priority 1, the second task was assigned priority 5, the third task was assigned priority 2, and so on. After that, Yurik rearranged the tasks in ascending order of their priorities, resulting in the sequence: 1, 3, 7, 6, 2, 8, 5, 4.

In the second time, Yurik chose the tasks that are in the list at positions from 2 to 4: 3, 7, and 6. The task with number 3 was assigned priority 1, the task with number 7 was assigned priority 5, and the task with number 6 was assigned priority 2. After Yurik rearranged the tasks in ascending order of their priorities, the sequence became 1, 3, 6, 7, 2, 8, 5, 4.

In the third time, Yurik chose the last two tasks in the current order: 5 and 4. The first task was assigned priority 1, and the second task was assigned priority 5, so their order of execution did not change.

In the fourth time, only the first task was chosen.

In the fifth time, the first three tasks in the current order were chosen, and after assigning priorities, tasks 3 and 6 swapped places.

Finally, in the sixth time, all tasks except the first and the last were chosen. The final order of task execution looks like this: 1, 6, 7, 5, 3, 8, 2, 4.

Problem I. Roofs

Time limit: 2 seconds
Memory limit: 512 megabytes

During archaeological excavations, the ruins of an ancient temple were discovered. The colonnade, consisting of n columns arranged in a row, has been preserved the best. All the columns have been partially destroyed, so it turned out that the heights of all the columns are different. The height of the i -th column is h_i .

To prevent further damage to the columns due to bad weather, it was decided to cover them with roofs. Each roof is placed horizontally and one end is attached to the top of a certain column. It is possible to install a roof that covers the segment of columns from the i -th to the j -th ($1 \leq i \leq j \leq n$) if one of the following conditions is met:

- If the i -th column is higher than all the others in the segment $[i, j]$, then the roof can be secured with its left end to the i -th column.
- If the j -th column is higher than all the others in the segment $[i, j]$, then the roof can be secured with its right end to the j -th column.

At the top of each column, no more than one roof can be attached. The cost of attaching a roof to the i -th column is c_i , regardless of whether it is directed to the left or to the right and how many columns it covers.

It is required to attach roofs to some columns in such a way that each column is covered by at least one roof, and the total cost is minimized.

Input

The first line contains the number n ($1 \leq n \leq 200\,000$) — the number of columns.

The second line contains n integers h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$) — the heights of the columns. It is guaranteed that all h_i are distinct.

The third line contains n integers c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$) — the costs of attaching roofs to the columns.

Output

Output a single number — the minimum cost to cover all the columns with roofs.

Examples

standard input	standard output
3 3 10 7 2 5 2	7
1 5 2	2

Problem J. Slime Escape

Time limit: 2 seconds
Memory limit: 512 megabytes

Nikita received a slime-making kit as a birthday gift. Excited about the present, Nikita created a square-shaped slime and placed it on the table. However, the slime got scared of Nikita's upcoming experiments and decided to escape.

The table is represented as a rectangular field with n rows and m columns. Some cells on the table contain holes, and if any of the slime's cells end up on a hole, it will fall off the table. The slime occupies 4 cells and initially starts in the top-left with size 2×2 . Its goal is to reach the bottom-right corner of size 2×2 .

The slime can move across the table using *transformations*. A transformation involves moving one of the slime's cells to another cell adjacent to it, sharing either a side or a corner. After each transformation, the slime must form a 4-connected shape (in other words, there must be a path between any two slime cells, with each step being adjacent to the previous cell by a side).

The slime wants to escape from Nikita as quickly as possible, ensuring that its cells never end up on any holes. Find the minimum number of transformations it needs to achieve this.

Input

The first line contains two integers n and m ($2 \leq n, m \leq 3 \cdot 10^5$; $n \cdot m \leq 3 \cdot 10^5$) — the dimensions of the table.

The next n lines contain m characters $a_{i,j}$, describing the table. The character “#” represents a hole in the table, and the character “.” represents the table's surface.

It is guaranteed that the top-left and bottom-right corners of size 2×2 are part of the table's surface.

Output

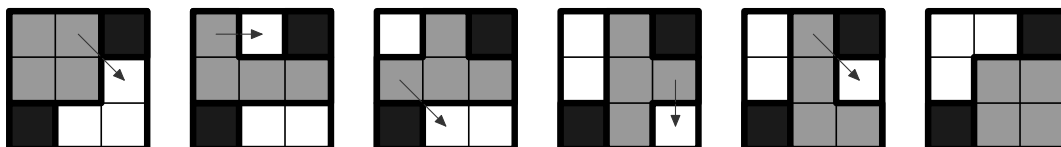
Output the minimum number of transformations required for the slime to occupy the bottom-right corner of size 2×2 , starting from the top-left corner. If it is not possible, output -1 .

Examples

standard input	standard output
<pre>3 3 ..# ... #..</pre>	5
<pre>3 5 ..### ##...</pre>	-1

Note

The figure below shows the transformations for the first example. The black cells represent holes in the table, and the gray cells represent the slime. The arrows indicate the movement of cells.



Problem K. Production Waste

Time limit: 2 seconds
Memory limit: 512 megabytes

A large industrial company manages a plant that produces important goods and ecologically disposes of waste. Even on days when the plants are idle (which can happen frequently) a portion of the waste is still disposed of safely.

It is known that the plants operated for n days, and the i -th day was the d_i -th production day since the company started operating. On this day, w_i units of raw material were used for production. On the days when the plant was idle no raw material was used.

Production and disposal are parameterized by the *material processing coefficient* r . The higher the coefficient, the faster the disposal process, but the more waste is generated during raw material processing. Formally,

- If there are exactly x units of waste at the end of a certain day then at the beginning of the next day there will be $x \cdot (1 - r)$ units of waste remaining.
- If there are exactly x units of waste at the beginning of a day then by the end of the day there will be $x + w \cdot r$ units of waste, where w is the amount of raw material used on that day.

In other words, if at the end of day $j - 1$ there are x units of waste at the plant and w_j units of raw material are used on day j then at the end of day j the amount of waste will be $(1 - r)x + rw_j$. At the beginning of the very first day there was 0 waste.

An inspection will soon arrive at the plant. The inspection is interested in information about the amount of waste that has not yet been disposed of at the end of certain days. Unfortunately these records have been lost, so the company's management asks for your help in answering the inspection's questions based on the production data.

Input

The first line of input contains two space-separated integers n and m , which represent the number of days on which new waste was produced and the number of days of interest to the inspection, respectively ($1 \leq n, m \leq 10^5$).

The second line contains a single real number r , which represents the raw material processing coefficient at the plant ($0 \leq r \leq 1$).

Each of the next n lines contains two space-separated integers d_i and w_i , representing the number of the production day and the amount of raw material used on that day, respectively ($1 \leq d_i, w_i \leq 10^9$). It is guaranteed that all d_i values are distinct.

The last line contains m space-separated integers q_i , representing the days of interest to the inspection ($1 \leq q_i \leq 10^9$).

Output

Output m numbers each on a separate line, representing the amount of waste stored at the factory on each of the inspection days.

Your answer will be accepted if each of the output numbers has an absolute or relative error of no more than 10^{-6} .

Examples

standard input	standard output
3 3 0.5 2 6 3 8 4 10 1 3 5	0 5.5 3.875
5 6 0.3333333 1 27 6 27 3 27 4 27 5 27 1 2 3 4 5 6	9 6 13 17.666666 20.777777 22.851851
1 1 0 1 1000000000 1000000000	0

Note

In the first example, the amount of waste will change as follows:

1. On the first day, there is no waste.
2. On the second day, the waste will be $0.5 \cdot 6 = 3$.
3. On the third day, the waste will be $0.5 \cdot 3 + 0.5 \cdot 8 = 5.5$.
4. On the fourth day, the waste will be $0.5 \cdot 5.5 + 0.5 \cdot 10 = 7.75$.
5. On the fifth day, the plant does not produce anything new, and the waste becomes $0.5 \cdot 7.75 = 3.875$.

Problem L. Seats in the subway

Time limit: 1 second
Memory limit: 512 megabytes

Vasily likes to use his phone in the subway. But he doesn't want anyone to see his phone screen. Vasily asked you to help all those who like to sit alone.

Consider a subway car that has one row of n seats, numbered from 1 to n , initially all seats are empty. k consecutive requests are received. There are two types of requests:

1. “ $-x$ ”. Passenger who entered the x -th left. It is guaranteed that such a passenger is in the subway.
2. “ $+$ ”. A new passenger has arrived. It is guaranteed that there is at least one free seat.

For each request of the second type, you should display a number of free seat. The distance to the nearest occupied seat should be maximum. If there are several such seats, you should display the seat with the minimum number. Starting from the next request, this seat is considered occupied until it is freed.

Input

The first line contains two integers n and k — the number of seats and the number of requests ($1 \leq n \leq 10^{18}$; $1 \leq k \leq 10^5$). The next k lines contain queries, one on each line.

Output

For each request “ $+$ ” print the number of the seat where the corresponding passenger will sit.

Examples

standard input	standard output
5 9 + -1 + + + + -4 + +	1 1 5 3 2 3 4
5 8 + + + + + -4 -3 +	1 5 3 2 4 2