

Problem A. Nuclear Reactor

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Dream Land has n nuclear reactor plants. Each of the plants is positioned on a straight line p_i ($1 \leq p_i \leq 10^{18}$). At the beginning, all plants are inactive. The president of Dream Land wants to activate exactly k ($1 \leq k \leq n$) plants in such away the distance between any two active plants is as large as possible. The president assigned this task to you.

Input

The first line of input consist of two integers n ($1 \leq n \leq 10^5$) and k ($1 \leq k \leq n$), the number of nuclear plants and the number of activated plants respectively. The second line consist of n integers which are the positions of the nuclear reactor plants p_i ($1 \leq p_i \leq 10^{18}$).

Output

The output consist of exactly k integers the positions of activated plants such as the distance between any two activated plants is as large as possible. If there are multiple answers, print any of them.

Example

standard input	standard output
5 3 4 5 11 8 10	5 8 11

Problem B. Does it Percolate?

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

Percolation is the process of a liquid slowly passing through a filter or surface. We will consider a 2d mesh or lattice grid for our surface.

A lattice grid here is a normal 2-dimensional grid (where each cell is connected to at most 4 cell at its 4-side) with an additional circular connection at its border. Meaning if we have a lattice of dimension $(4, 5)$ then cell at position $(2, 1)$ will have a connection with cell $(2, 5)$, and cell at position $(1, 4)$ will have a connection with cell $(4, 4)$. In other words the left-most cells will have a connection with the right-most cells in the same row and the top-most cells will have a connection with the bottom-most cells in the same column.

Each lattice grid is made of different surface or material, which determines the atomic gap in each cell and whether it will allow any liquid to pass through. For each cell (i, j) in the lattice you will be given the probability p_{ij} of accepting any given liquid. Each liquid also has its own probability threshold p , if any cell has less probability than p then that liquid does not pass through and that cell remains unoccupied.

Given the details of lattice cells and the threshold p it will create clusters of occupied cells (by the liquid) using the connection of each cell. Now if at least one of the cluster has all the following characteristics then the lattice percolates—

- * Has at least 1 cell from each row.
- * Has at least 1 cell from each column.

Input

The first line of input contains two integers and a float. The two integers n and m ($1 \leq n, m \leq 1000$) indicates width and length of the lattice. The float p ($0.00 \leq p \leq 1.00$) indicates the threshold for the given liquid.

The following n lines each contains m floats $p_{11}, p_{1j}, \dots, p_{nm}$ which indicates the probability of accepting any given liquid for cell (i, j) .

The input data set is big.

Output

Print 'YES' if the lattice percolates or else 'NO', without the quotes.

Examples

standard input	standard output
4 4 0.50 0.19 0.52 0.23 0.33 0.78 0.65 0.45 0.49 0.48 0.66 0.77 0.89 0.30 0.36 0.88 0.15	YES
3 3 0.10 0.09 0.09 0.09 0.09 0.10 0.09 0.09 0.09 0.09	NO

Problem C. Perfect Matching

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

There is a class of $2 * n$ students, and the teacher wants all of them to get matched into n pairs. She divides the students into two lines of length n , and numbers the students in both lines from 1 to n .

The i_{th} numbered student in the first line can be partnered with the j_{th} numbered student in the second line if $|i - j| \leq e$.

However, there are k pairs of students that cannot be paired together to avoid trouble in the classroom.

You need to print the number of ways you can match the students into n pairs such that the constraints above are met. One way is different than the other if at least one student has a different partner.

Input

The first line contains 3 integers n, e, k ($1 \leq n \leq 2000, 0 \leq e \leq 4, 0 \leq k \leq 2000$), the number of students in each line, the value that determines the range, and the number of invalid pairs, respectively.

Each of the next k lines contains two integers u_i, v_i ($1 \leq u_i, v_i \leq n$), the number of the student from the first line and the number of the student from the second line that cannot be matched together respectively. No pair of students will appear twice in the input.

Output

Output the number of ways modulo $10^9 + 7$, on a single line.

Examples

standard input	standard output
2 1 0	2
2 1 1 1 2	1

Problem D. North East South West

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Ahmad is playing a game with his friend. The game works like this. Ahmad will start by facing north, then he close his eye. When his friend say “right” he will rotate clockwise by 90 degree. So if he was facing north, he will now face east. If he was facing east, he will now face south. If his friend says “left”, he will rotate counter-clockwise by 90 degree. So if he is facing north, he will now face west. If he was facing west, he will now face south. When his friend say “back” he can either rotate clockwise or counter-clockwise by 180 degree. So if he was facing north, he will now face south. If he was facing west, he will now face east.

After several command, Ahmad will now guess which direction he is facing. Help Ahmad figure-out which direction he is facing now.

Input

Input starts with a number n , ($1 \leq n \leq 10^5$). The next n lines each consist of a single word which is either, “left”, “right” or “back”.

Output

Output a single word, which can be either “north”, “east”, “south”, “west”.

Example

standard input	standard output
4 left left left right	south

Problem E. Tree Division

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Given a tree of size n and an integer k . Your task is to determine if the tree can be divided into k non-intersecting subtrees **of the same size**. Every node of the tree should belong to exactly one subtree.

Input

The first line of input contains two integers n and k ($1 \leq n, k \leq 10^5$), the number of nodes in the tree and the number of subtrees after dividing the tree, respectively.

Each of the following $n - 1$ lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$), representing an edge that connects the two nodes. It is guaranteed that the given graph is a tree.

Output

Print "Yes" if it is possible to divide the tree into K subtrees of the same size. Otherwise print "No".

Examples

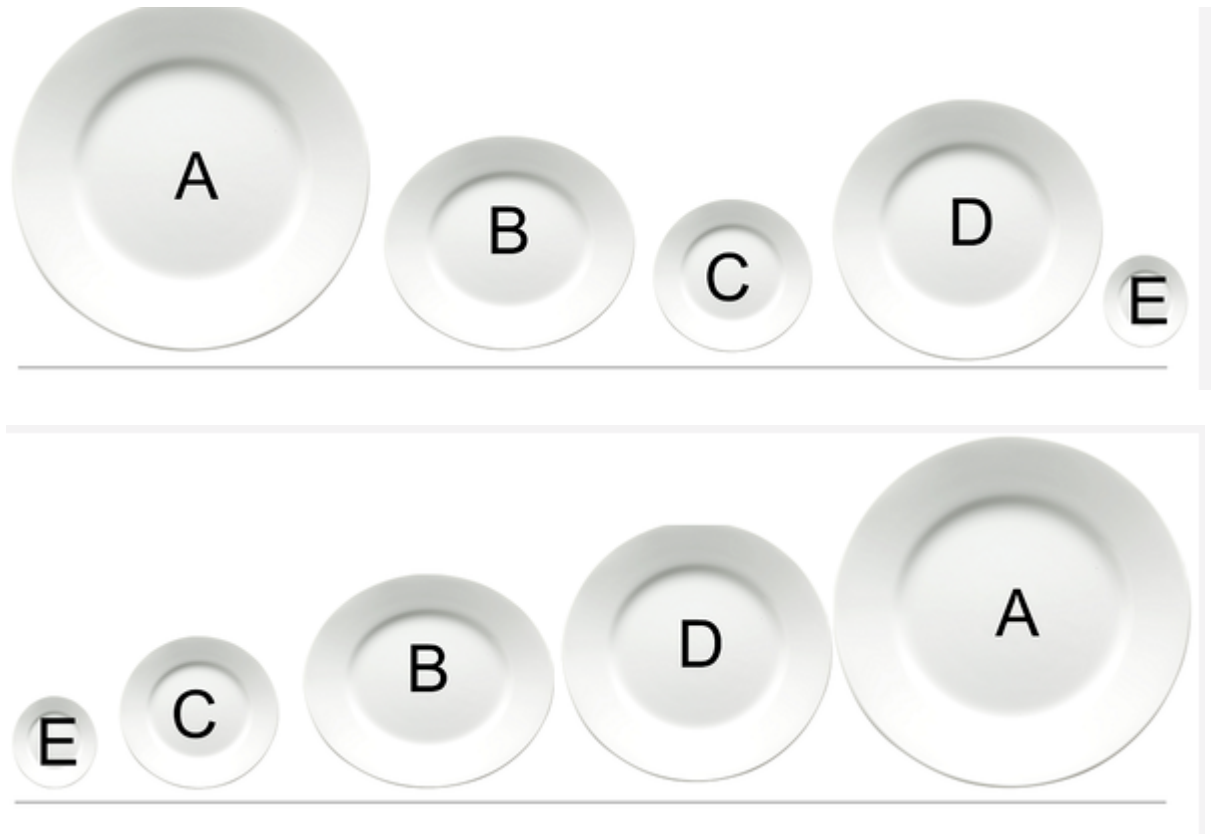
standard input	standard output
4 2 1 2 2 3 3 4	Yes
5 2 5 1 5 2 5 3 3 4	No

Problem F. Kitchen Plates

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

You are given 5 different sizes of kitchen plates. Each plate is marked with a letter **A**, **B**, **C**, **D**, or **E**. You are given 5 statements comparing two different plates, you need to rearrange the plates from smallest size to biggest size.

For example: the sizes of these plates



Input

The input consist of 5 lines. In each line there will be 3 characters, the first and last character will be either **A**, **B**, **C**, **D**, or **E** and the middle character will be either **>** or **<** describing the comparison between two plates sizes. No two plates will be equal.

Output

The output consist of 5 characters, the sorted order of balls from smallest to biggest plate. Otherwise, if the statements are contradicting print **impossible**. If there are multiple answers, print any of them

Examples

standard input	standard output
D>B A>D E<C A>B B>C	ECBDA
B>E A>B E>A C<B D<B	impossible