Codeforces Round #108 (Div. 2)

A. Marks

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya, or Mr. Vasily Petrov is a dean of a department in a local university. After the winter exams he got his hands on a group's gradebook.

Overall the group has n students. They received marks for m subjects. Each student got a mark from 1 to 9 (inclusive) for each subject.

Let's consider a student the *best at some subject*, if there is no student who got a higher mark for this subject. Let's consider a student *successful*, if there exists a subject he is the *best at*.

Your task is to find the number of *successful* students in the group.

Input

The first input line contains two integers n and m ($1 \le n, m \le 100$) — the number of students and the number of subjects, correspondingly. Next n lines each containing m characters describe the gradebook. Each character in the gradebook is a number from 1 to 9. Note that the marks in a rows are not sepatated by spaces.

Output

Print the single number — the number of *successful* students in the given group.

Examples

input
3 3 223 232 112
output
2

input
3 5 91728 11828 11111
output
3

Note

In the first sample test the student number 1 is the best at subjects 1 and 3, student 2 is the best at subjects 1 and 2, but student 3 isn't the best at any subject.

In the second sample test each student is the best at at least one subject.

B. Steps

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Vasya went out for a walk in the yard but there weren't any of his friends outside and he had no one to play touch and run. But the boy didn't lose the high spirits and decided to play touch and run with himself. You may ask: "How did he do that?" The answer is simple.

Vasya noticed that the yard is a rectangular $n \times m$ field. The squares have coordinates (x, y) ($1 \le x \le n, 1 \le y \le m$), where x is the index of the row and y is the index of the column.

Initially Vasya stands in the square with coordinates (x_c, y_c). To play, he has got a list of k vectors (dx_i, dy_i) of non-zero length. The game goes like this. The boy considers all vectors in the order from 1 to k, and consecutively chooses each vector as the current one. After the boy has chosen a current vector, he makes the maximally possible number of valid steps in the vector's direction (it is possible that he makes zero steps).

A *step* is defined as one movement from the square where the boy is standing now, in the direction of the current vector. That is, if Vasya is positioned in square (x, y), and the current vector is (dx, dy), one step moves Vasya to square $(x + dx, y + dy)$. A step is considered *valid*, if the boy does not go out of the yard if he performs the step.

Vasya stepped on and on, on and on until he ran out of vectors in his list. Ha had been stepping for so long that he completely forgot how many steps he had made. Help the boy and count how many steps he had made.

Input

The first input line contains two integers n and m ($1 \le n, m \le 10^9$) — the yard's sizes. The second line contains integers x_c and y_c — the initial square's coordinates ($1 \le x_c \le n, 1 \le y_c \le m$).

The third line contains an integer k ($1 \le k \le 10^4$) — the number of vectors. Then follow k lines, each of them contains two integers dx_i and dy_i ($|dx_i|, |dy_i| \le 10^9, |dx| + |dy| \ge 1$).

Output

Print the single number — the number of steps Vasya had made.

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specificator.

Examples

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

input
10 10 1 2 1 -1 0
output
0

Note

In the first sample Vasya is initially positioned at square $(1, 1)$ and makes 3 steps by the first vector $(1, 1)$. So, he consecutively visits the squares $(2, 2), (3, 3), (4, 4)$. Then he makes 0 steps by the second vector $(1, 1)$. He makes 1 more step by the third vector $(0, -2)$ and he ends up in square $(4, 2)$. Overall, Vasya makes 4 steps.

In the second sample Vasya is initially positioned in square $(1, 2)$ and makes 0 steps by vector $(-1, 0)$, as the square with coordinates $(0, 2)$ is located outside the yard.

C. Pocket Book

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day little Vasya found mom's pocket book. The book had n names of her friends and unusually enough, each name was exactly m letters long. Let's number the names from 1 to n in the order in which they are written.

As mom wasn't home, Vasya decided to play with names: he chose three integers i, j, k ($1 \le i < j \le n$, $1 \le k \le m$), then he took names number i and j and swapped their prefixes of length k. For example, if we take names "CBDAD" and "AABRD" and swap their prefixes with the length of 3, the result will be names "AABAD" and "CBDRD".

You wonder how many different names Vasya can write instead of name number 1, if Vasya is allowed to perform any number of the described actions. As Vasya performs each action, he chooses numbers i, j, k independently from the previous moves and his choice is based entirely on his will. The sought number can be very large, so you should only find it modulo 1000000007 ($10^9 + 7$).

Input

The first input line contains two integers n and m ($1 \le n, m \le 100$) — the number of names and the length of each name, correspondingly. Then n lines contain names, each name consists of exactly m uppercase Latin letters.

Output

Print the single number — the number of different names that could end up in position number 1 in the pocket book after the applying the procedures described above. Print the number modulo 1000000007 ($10^9 + 7$).

Examples

input
2 3 AAB BAA
output
4

input
4 5 ABABA BCGDG AAAAA YABSA
output
216

Note

In the first sample Vasya can get the following names in the position number 1: "AAB", "AAA", "BAA" and "BAB".

D. Frames

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Vasya got hold of a sheet of checkered paper $n \times m$ squares in size. Our Vasya adores geometrical figures, so he painted two rectangles on the paper. The rectangles' sides are parallel to the coordinates' axes, also the length of each side of each rectangle is no less than 3 squares and the sides are painted by the grid lines. The sides can also be part of the sheet of paper's edge. Then Vasya hatched all squares on the rectangles' *frames*.

Let's define a rectangle's frame as the set of squares **inside** the rectangle that share at least one side with its border.

A little later Vasya found a sheet of paper of exactly the same size and couldn't guess whether it is the same sheet of paper or a different one. So, he asked you to check whether the sheet of paper he had found contains two painted frames and nothing besides them.

Please note that the frames painted by Vasya can arbitrarily intersect, overlap or even completely coincide.

The coordinates on the sheet of paper are introduced in such a way that the X axis goes from top to bottom, the x coordinates of the squares' numbers take values from 1 to n and the Y axis goes from the left to the right and the y coordinates of the squares' numbers take values from 1 to m.

Input

The first input line contains two integers n and m ($3 \le n, m \le 1000$) — the sizes of the sheet of paper Vasya found. Next n lines, each consisting of m symbols "." (dot) and "#" (number sign), describe the found sheet of paper. The symbol "#" represents a hatched square and the symbol "." represents a non-hatched square.

Output

In the first line print the single word "YES" or "NO", meaning whether it is true that the found sheet of paper has two frames painted on it. If the answer is positive, then print in the second line 4 integers: the coordinates of the upper left and lower right corners of the first frame. In the third line print 4 integers: the coordinates of the upper left and the lower right corners of the second frame. If there are multiple answers, print any of them.

Examples

input
4 5 ##### #.#.# ###.# #####
output
YES 1 1 3 3 1 1 4 5

input
5 6 ...### ...### #####. #...#. #####.
output
NO

Note

In the first sample there are two frames on the picture. The first one is:

```
###..
#.#..
###..
.....
```

The second one is:

```
#####
#...#
#...#
#####
```

In the second sample the painted figures are not frames. Note that the height and width of valid frames is no less than 3.

E. Garden

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya has a very beautiful country garden that can be represented as an $n \times m$ rectangular field divided into $n \cdot m$ squares. One beautiful day Vasya remembered that he needs to pave roads between k important squares that contain buildings. To pave a road, he can cover some squares of his garden with concrete.

For each garden square we know number a_{ij} that represents the number of flowers that grow in the square with coordinates (i, j). When a square is covered with concrete, all flowers that grow in the square die.

Vasya wants to cover some squares with concrete so that the following conditions were fulfilled:

- all k important squares should necessarily be covered with concrete
- from each important square there should be a way to any other important square. The way should go be paved with concrete-covered squares considering that neighboring squares are squares that have a common side
- the total number of dead plants should be minimum

As Vasya has a rather large garden, he asks you to help him.

Input

The first input line contains three integers n, m and k ($1 \le n, m \le 100$, $n \cdot m \le 200$, $1 \le k \le min(n \cdot m, 7)$) — the garden's sizes and the number of the important squares. Each of the next n lines contains m numbers a_{ij} ($1 \le a_{ij} \le 1000$) — the numbers of flowers in the squares. Next k lines contain coordinates of important squares written as "x y" (without quotes) ($1 \le x \le n$, $1 \le y \le m$). The numbers written on one line are separated by spaces. It is guaranteed that all k important squares have different coordinates.

Output

In the first line print the single integer — the minimum number of plants that die during the road construction. Then print n lines each containing m characters — the garden's plan. In this plan use character "X" (uppercase Latin letter X) to represent a concrete-covered square and use character "." (dot) for a square that isn't covered with concrete. If there are multiple solutions, print any of them.

Examples

input
3 3 2 1 2 3 1 2 3 1 2 3 1 2 3 3
output
9 .X. .X. .XX

input
4 5 4 1 4 5 1 2 2 2 2 2 7 2 4 1 4 5 3 2 1 7 1 1 1 1 5 4 1 4 4
output
26 X..XX XXXX. X.X.. X.XX.