Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final. **The scoreboard won't be shown during the last 30 minutes of the contest.**

Problem A. Snowman

Time limit: 1 second

Do you wanna build a snowman? Of course, you do! And you finally have enough snow on the walk in front of your house. But you see that snow on different sections of the walkway has different quality: on some sections, snow is good, white and sticky, let's say, such sections are class **a** sections, other sections with a bit worse quality would have class **b**, for even worse quality — class **c** and so on.

To build a base for a snowman you should roll a snowball forward or backward from one track section to another. There is enough snow, so you can go back to the same section as many times as you want. Of course, you want to build a snowman out of the best snow, and the closer to the centre of the snowball, the more important the quality of snow for the future snowman, so at the beginning of the building process you should choose the best sections. For example, if you roll a snowball starting on section with class **c**, then on section **a**, and then on section **b** (**cab**), it would not be as solid as the **bab** ball, and the **aca** snowball would be even better.

Classes of track sections are written in the string s. To make the first ball, you need to roll it through the track sections k times. You can start to roll a ball for the snowman at any section of the track. What sequence of sections should be used to get the most solid ball?

Input

The first line contains the string s of lowercase letters — track sections classes. The number of sections is not less than 2 and not greater than 100.

The second line contains one integer k ($1 \le k \le 10^4$) — the number of sections for the snowball.

Output

Print the sequence of letters without spaces — section classes in the order you will roll the ball.

Scoring

Subtask	Score	Constraints		
		Length s	k	Additional
1	11	$\|s\| = 2$	$k \le 15$	—
2	17	$\|s\| \le 3$	$k \le 15$	—
3	19	$\|s\| \le 100$	$k \le 15$	—
4	24	$\|s\| \le 26$	$k \le 10^4$	all letters in s are different
5	29	$\|s\| \le 100$	$k \le 10^4$	—

Examples

standard input	standard output
dcabe 3	aba
bbb 5	bbbbb

Problem B. Teacher Sorting

Time limit: 1 second

The 9-th grade students are doing Physical Education, they have just run a long distance. The lesson is approaching its end, so the teacher asked students to stand in a line in non-decreasing order of their heights. Students don't always pay their attention, so sometimes they stand in a line not in the order they were asked. The teacher wants to fix the problem.

The teacher looks at the line, and if it's not ordered properly, chooses the i-th and the j-th student in the line, and swaps them. So after the swap, a the i-th student becomes the j-th student, and the other way around. Teacher keeps doing swaps, until the line is ordered properly. Formally speaking, until for all i — the $(i+1)$-th student is not shorter than the i-th student in the line.

Although today it won't be easy for the teacher. Students are very tired after the run so they can hardly stand. The teacher doesn't want to overload them physically, so the teacher won't move any student more than once.

The teacher needs your help. You are given the line: $a_1, a_2, \ldots, a_n$ — the heights of the students. Find the sequence of swaps for teacher to make the line ordered properly, or say that it's not possible.

Input

The first line contains an integer n — the number of students ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$). The number a_i is the height of the i-th student in the line.

Output

If the teacher won't be able to order the students properly, print "`No`".

Otherwise, print "`Yes`" in the first line. In the second line print an integer k — the number of swaps the teacher needs to make. In each of the next k lines print two integers i and j, denoting that the teacher should swap the i-th and the j-th students in the line.

Note that you don't have to minimize the number of swaps. You can print any sequence that will make the line ordered properly in a way that no student was swapped more than once.

Scoring

Subtask	Score	Constraints		
		n	a_i	Additional
1	21	$n \le 10$	$a_i \le 100$	—
2	19	$n \le 2 \cdot 10^5$	$a_i \le 10^9$	all a_i are distinct
3	23	$n \le 2 \cdot 10^5$	$a_i \le 100$	—
4	37	$n \le 2 \cdot 10^5$	$a_i \le 10^9$	—

Examples

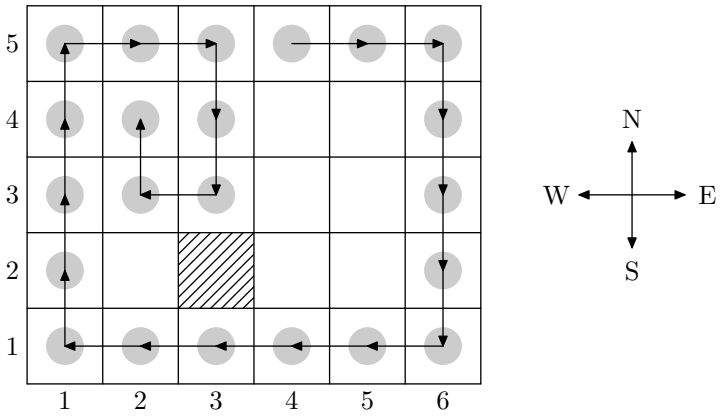
standard input	standard output
3 3 2 1	Yes 1 3 1
6 2 5 5 2 10 9	Yes 2 5 6 2 4
5 2 3 4 5 1	No

Problem C. Garbage Robot

Time limit: 2 seconds

Misha lives in a rectangular room. The room can be represented as a matrix of size $m \times n$. Misha lives a frugal life, so there are only four walls and a sofa in his room, that occupies exactly one cell of the matrix.

Recently, a garbage robot appeared in the room. It starts working in a certain cell of the matrix, and has a certain direction. Then, every minute two actions happen. First, the garbage robot places the garbage in the cell where it is located. Next, it tries to go one cell forward in its current direction. It cannot pass through walls and move into a cell that already contains something (a sofa or garbage that was placed previously). If the robot can't move, it tries to change its direction 90 degrees clockwise. If the robot can't move after that, it shuts down. After the robot shuts down, all the garbage left by it magically disappears until the next time the robot appears.



The robot appeared q times. For each appearance, Misha wants to know how many cells will the robot have time to leave garbage in before it finishes working. Help him!

Input

The first line of the input file contains three integers n, m, and q: the size of the room and the number of robot appearances ($1 \le n, m \le 10^9$; $1 \le q \le 10^5$).

The second line contains two integers: a, b: coordinates of the sofa ($1 \le a \le n$; $1 \le b \le m$).

Next is the q lines. The i-th of them contains two integers x_i, y_i and the character c_i: coordinates of the robot and its direction ($1 \le x \le n$; $1 \le y \le m$). It is guaranteed that $(x, y) \ne (a, b)$. The character specifies one of the four cardinal directions:

- '`E`' — east.
- '`W`' — west.
- '`S`' — south.
- '`N`' — north.

The cells $(u, v + 1)$, $(u, v - 1)$, $(u + 1, v)$, $(u - 1, v)$ are located north, south, east, and west of (u, v), respectively.

Output

Output q lines, with one integer each — the number of cells that the robot will visit, for each robot appearance.

Scoring

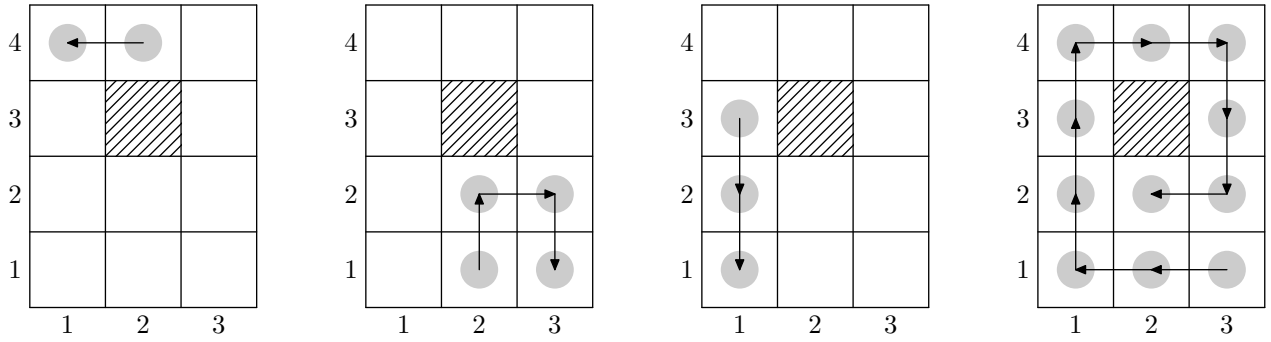
Subtask	Score	Constraints
1	15	$n, m, q \leq 100$
2	35	$n, m, q \leq 2000$
3	50	$n, m \leq 10^9$; $q \leq 10^5$

Example

standard input	standard output
3 4 4 2 3 2 4 S 2 1 N 1 3 E 3 1 W	2 4 3 11

Note

Illustrations for example:



Problem D. Digidivisible Numbers

Time limit: 2 seconds

Let's call a number *digidivisible* in base B, if it is divisible by all digits in its base B representation. For example, 728_{10} is divisible by 7, 2 and 8, so it is digidivisible in base 10, and number $264_8 = 180_{10}$ is divisible by 2, 6 and 4, so it is digidivisible in base 8.

You are given integers B and n, and some set of allowed digits from 1 to $B - 1$. Find the number of digidivisible numbers consisting of n digits in base B, only containing these allowed digits. Solve this problem for some fixed n and B and for multiple sets of allowed digits.

Input

The first line contains two integers B and n ($2 \le B \le 10$; $1 \le n \le 10^9$). The second line contains an integer t ($1 \le t \le 2^{B-1} - 1$) — the number of sets of allowed digits you need to solve this problem for.

Then, t lines follow, i-th line contains a single string s_i, consisting of B zeros and ones. If $s_{i,k} = 1$ (indices begin with 0), then digit k is allowed, otherwise digit k is forbidden. Each set has at least one allowed digit, and digit 0 is always forbidden. All t sets are distinct.

Output

For each one of the t sets print the answer in a separate line. Because the result might be huge, print it modulo 999 999 001.

Scoring

Subtask	Points	Constraints
1	8	$B = 10$, $n \le 5$, $t = 1$ all digits except 0 are allowed
2	9	$B \le 10$, $B^n \le 10^5$, $t = 1$
3	9	$B \le 10$, $B^n \le 10^5$
4	32	$B \le 10$, $n \le 50$
5	13	$B \le 6$, $n \le 10^9$
6	14	$B \le 8$, $n \le 10^9$
7	15	$B \le 10$, $n \le 10^9$

Example

standard input	standard output
10 3 2 0111111111 0010101010	56 17

Note

Total number of 3-digit digidivisible numbers in base 10 is 56. If we only allow even digits, then there are 17 numbers left: 222, 224, 244, 248, 264, 288, 424, 444, 448, 488, 624, 648, 666, 824, 848, 864, 888.

Problem E. Yet Another Minimax Problem

Time limit: 2 seconds

You are given n points on the plane. You have to choose such a straight line, that there will be points on both sides of the line and the minimum distance from one of the points to the line should be the maximum possible. Find this distance.

Input

The first line contains a single integer n — the number of the points ($2 \le n \le 2\,000$).

Each of the following n lines contains two integers x_i and y_i — coordinates of i-th point ($|x_i|, |y_i| \le 10^9$).

It's guaranteed that no two points coincide.

Output

Print single real number — an answer. Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-9}.

Scoring

Subtask	Score	Constraints
1	11	$n \le 10$
2	19	Points form strictly convex non-degenerate polygon, given in the counter clockwise order, $3 \le n$
3	23	$n \le 100$
4	31	$n \le 1\,000$
5	16	No additional constraints

Examples

standard input	standard output
4 0 0 0 1 1 0 1 1	0.50000000000
2 -12 34 56 -78	65.51335741664
8 0 0 2 0 5 1 5 3 4 5 3 4 1 4 0 2	1.10000000000

Note

A picture for the third example: