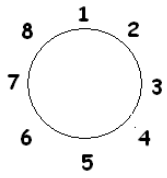
TASK	VOLIM	MISA	SLOM	PUTNIK	PALETA	LINIJE

source code	`volim.pas`
	`volim.c`
	`volim.cpp`
		`misa.pas`
		`misa.c`
		`misa.cpp`
			`slom.pas`
			`slom.c`
			`slom.cpp`
				`putnik.pas`
				`putnik.c`
				`putnik.cpp`
					`paleta.pas`
					`paleta.c`
					`paleta.cpp`
						`linije.pas`
						`linije.c`
						`linije.cpp`

input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	1 second	1 second
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	64 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Problems translated from Croatian by: **Paula Gombar**

Croatia's national television programme is broadcasting an entertainment show titled "I Love Croatia", modeled on the licensed format *I love my country*. In this show two teams of celebrities and public figures play various games which require knowledge about Croatia. One of the games is *Happy Birthday,* which will be used in this task, although somewhat altered.



Eight players numbered one through eight are seated in a circle (see picture). One of them is holding a box which will explode after 3 minutes and 30 seconds **from the beginning of the game** when some colorful confetti will be blown out. The game begins with a question to the player holding the box. If the players **answers incorrectly or skips the question**, he is immediately given the next question. If the player **answers correctly**, he passes the box to the first player seated on his left and then that player gets the next question.

You are given the numbered label of the player who has the box in the beginning and the outcomes of the first **N** questions asked. Determine the numbered label of the player who had the box when it finally exploded. The question outcome is described with the following data - time passed **from the beginning of the question being asked to the moment an answer was given** and whether the answer was true ("**T**"), false ("**N**") or skipped ("**P**"). The time between giving the answer and asking the next question shouldn't be taken into consideration, as well as time necessary for the box to be passed to the next player. The box will surely explode on a player's turn.

INPUT

The first line of input contains a positive integer **K** ($1 \leq \mathbf{K} \leq 8$), the numbered label of the player who has the box initially.

The second line of input contains a positive integer **N** ($1 \leq \mathbf{N} \leq 100$), the number of questions asked during the game.

Each of the following **N** lines contains a positive integer **T** ($1 \leq \mathbf{T} \leq 100$), time passed from the beginning of the i^{th} question being asked to the moment an answer was given, measured in seconds, and a single character **Z** ('**T**', '**N**' or '**P**'), the type of answer given.

OUTPUT

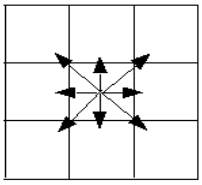
The first and only line of output must contain the numbered label of the player who had the box when it finally exploded.

SAMPLE TESTS

input	**input**	**input**
1 5 20 T 50 T 80 T 50 T 30 T	3 5 100 T 100 N 100 T 100 T 100 N	5 6 70 T 50 P 30 N 50 T 30 P 80 T
output	**output**	**output**
5	4	7

A nice part of the Roman Catholic Mass is the rite of peace when people shake hands with their neighbours and say "peace be with you". Mirko has found a way to turn this ritual into his own favor.

Inside the church, there are **R** rows of benches where each row can hold a capacity of **S** people. We can imagine the seating order as a matrix sized **R** x **S** where each element represents either a person or an empty seating space. Let us assume that each person shakes hands with their neighbours. That means that the neighbours are located in one of the eight neighbouring elements (if such element exists):



A seating order of the people inside the church has been given before Mirko enters. Mirko is, of course, late for the morning Mass and will sit in an empty space so that he shakes hands with **as many** people as he can. If there are no empty seats left, Mirko will simply give up on the idea and go to the evening Mass instead. We can assume that nobody enters the church after Mirko.

Calculate **the total number of handshakes** given during the morning Mass.

INPUT

The first line of input contains positive integers **R** and **S** ($1 \leq$ **R**, **S** ≤ 50) as stated in the text.

Each of the following **R** lines contains **S** characters. These **R** x **S** characters represent the seating order. The character '.' (dot) represents an empty place and the character 'o' (lowercase letter o) represents a person.

OUTPUT

The first and only line of output should contain the required number of handshakes.

SCORING

In test data worth 20% of total points, **R** will be 1.

In test data worth 20% of total points, **R** will be 2.

In test data worth 20% of total points, the church will be full, meaning all seats will be taken.

SAMPLE TESTS

input	**input**
2 3 ..o o..	2 2 oo oo
output	**output**
2	6

Clarification of the first example: Mirko will sit in a way that he can shake hands with both persons, which gives us two possibilities:

.oo ..o

o.. oo

Little Marin spent all his day generating test data for COCI. He simply couldn't make it work, so he had a nervous breakdown and can't even see clearly anymore. Every time he blinks while reading, the letters in a word get mixed up so that the letters from the second half of the word (the shorter half, if the length is an odd number) "jump in" between the letters from the first half in the following way:

- the last letter "jumps in" between the first and the second letter
- the penultimate letter "jumps in" between the second and the third letter
- the $\mathbf{k}^{th}$ letter from the end "jumps in" between the $\mathbf{k}^{th}$ and the $(\mathbf{k+1})^{th}$ letter from the beginning

For example, the word "abcdef" would become "afbecd" after blinking.

If Marin blinks again, the same thing happens. After two blinks, the word "abcdef" becomes "adfcbe".

Marin has decided to write a program to help him determine what's exactly written on the screen. Unfortunately, after a day's work, he's simply too tired and he needs your help. You are given **X**, the number of blinks, and the word Marin sees on the screen. Write a program to solve the mystery for Marin and determine what was actually the word before he blinked **X** times.

INPUT

The first line of input contains a positive integer **X** ($1 \leq \mathbf{X} \leq 1\ 000\ 000\ 000$), the number of times Marin blinked.

The second line of input contains the word from the screen, its length being from the interval [3, 1000]. The word will consist only from small letters of English alphabet.

OUTPUT

The first and only line of output must contain the original word, before Marin blinked **X** times.

SCORING

In test data worth 50 points, **X** will be smaller than or equal to 100.

SAMPLE TESTS

ulaz	**ulaz**	**ulaz**
4 acefdb	1000 aaaaaa	11 srama
izlaz	**izlaz**	**izlaz**
abcdef	aaaaaa	sarma

Clarification of the first example: The word gets altered in the following order: abcdef, afbecd, adfcbe, aedbfc, acefdb.

Chances are that you have probably already heard of the travelling salesman problem. If you have, then you are aware that it is an NP-hard problem because it lacks an efficient solution. Well, this task is an uncommon version of the famous problem! Its uncommonness derives from the fact that this version is, actually, solvable.

The travelling salesman is on a mission to visit **N** cities, each exactly once. The cities are represented by numbers 1, 2, ..., **N**. What we know is the direct flight duration between each pair of cities. The salesman, being the efficient man that he is, wants to modify the city visiting sequence so that the total flight duration is the minimum possible.

Alas, all is not so simple. In addition, the salesman has a peculiar condition regarding the sequence. For **each** city labeled **K** must apply: either all cities with **labels smaller than K** have been visited before the city labeled **K** or they will all be visited after the city labeled **K**. In other words, the situation when one of such cities is visited before, and the other after is not allowed.

Assist the poor fellow in his ambitious mission and calculate the minimum total flight duration needed in order to travel to all the cities, starting from whichever and ending in whichever city, visiting every city exactly once, so that his peculiar request is fulfilled.

INPUT

The first line of input contains the positive integer **N** ($2 \leq \mathbf{N} \leq 1500$), the number of cities.

Each of the following **N** lines contains **N** positive integers from the interval [0, 1000]. The number in $\mathbf{B}^{th}$ place in the $\mathbf{A}^{th}$ row represents the flight duration between cities **A** and **B**; that number is equal to the $\mathbf{A}^{th}$ number in the $\mathbf{B}^{th}$ row. When **A** = **B**, that number is 0. Otherwise, it is a positive value.

OUTPUT

The first and only line of output must contain the required minimum total flight duration.

SCORING

In test data worth 1/3 of total points, **N** will be at most 10.

In test data worth 1/2 of total points, **N** will be at most 20.

SAMPLE TESTS

ulaz	ulaz
3 0 5 2 5 0 4 2 4 0	4 0 15 7 8 15 0 16 9 7 16 0 12 8 9 12 0
izlaz	**izlaz**
7	31

Clarification of the first example: the optimal sequence is 2, 1, 3 or 3, 1, 2. The sequence 1, 3, 2 is even more favourable, but it does not fulfill the salesman's condition.

Clarification of the second example: the sequence is either 3, 1, 2, 4 or 4, 2, 1, 3.

Little Mirko spends his free time painting. For this hobby, he likes to use brushes and a pallet containing **K** colors overall. His friend Slavko decided to use Mirko's talent and gave him his new coloring book for Mirko to color. The coloring book contains **N** images numbered 1, 2, ..., **N**.

Mirko has decided to paint each image in exactly one color of the possible **K** colors from his pallet. However, he really likes colorful things. He chose **N** numbers $\mathbf{f_i}$ and decided to paint the image numbered **i** differently than the images numbered $\mathbf{f_i}$, except when $\mathbf{f_i = i}$. If $\mathbf{f_i = i}$, that means he can paint the image numbered $\mathbf{f_i}$ whichever color he likes, as long as all other conditions have been met.

Mirko wants to know the number of possible ways to color Slavko's coloring book and he desperately needs your help! Calculate the number of possible ways to color the book. Given the fact that the output can be very large, print the answer modulo 1 000 000 007.

INPUT

The first line of input contains positive integers **N**, **K** ($1 \leq \mathbf{N}, \mathbf{K} \leq 1\,000\,000$).

Following line contains **N** numbers $\mathbf{f_i}$ ($1 \leq \mathbf{f_i} \leq \mathbf{N}$), the number stated in the text.

OUTPUT

The first and only line must contain the number of possible ways to color Slavko's book.

SCORING

In test data worth 50% of total points, all numbers $\mathbf{f_i}$ will be different.

SAMPLE TESTS

ulaz	**ulaz**	**ulaz**	**ulaz**
2 3 2 1	3 4 2 3 1	3 4 2 1 1	3 4 1 1 2
izlaz	**izlaz**	**izlaz**	**izlaz**
6	24	36	36

Clarification of the first example: Mirko has three colors and decided that the image numbered 2 mustn't be of the same color as the image numbered 1. The possible colorings are (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), where the first number in the brackets represents the color of the first image and the second number the color of the second image.

Clarification of the fourth example: Mirko has four colors. There are no conditions regarding the first image, it can be painted in whichever color. The second must be different than the first, and the third different than the second. That means that those two images can be colored in the remaining 3 colors. This gives us a total of 36 combinations.

Mirko and his faithful friend Slavko got really bored one day. The result of their boredom is the creation of a new game! In the beginning of the game, they draw **N** points in a coordinate system. The players take turns and Mirko plays first. He draws a straight line which is parallel to one of the axes of the coordinate system and passes through one of the **N** points. In the following moves, the player draws a straight line which is parallel to one of the axes of the coordinate system and passes through one of the **N** points located on the line drawn in the previous move of the opponent. No single line must be drawn twice. The loser is the player who cannot play his move. Determine who has the winning strategy.

INPUT

The first and only line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 10\,000$).

Each of the following N lines contains two integers **X** and **Y**, the coordinates of the points drawn ($1 \leq \mathbf{X}, \mathbf{Y} \leq 500$).

OUTPUT

The first and only line of output must contain the name of the winner, either 'Mirko' or 'Slavko'.

SCORING

In test cases worth 40% of total points, **N** will not exceed 10.

SAMPLE TESTS

input	input
3 1 1 1 2 1 3	4 1 1 1 2 2 1 2 2
output	**output**
Mirko	Slavko

Clarification of the first example: If Mirko draws the line y = 1, Slavko has to draw x = 1. Then Mirko draws the line y = 2, and Slavko's only remaining move is to draw x = 1 again, which isn't allowed.