Croatian Open Competition in Informatics

Round 3, December 12th 2020

Tasks

Task	Time limit	Memory limit	Points
Knjige	1 second	512 MiB	50
Vlak	1 second	512 MiB	70
Sateliti	3 seconds	512 MiB	110
Selotejp	1 second	512 MiB	110
Specijacija	4 seconds	1 GiB	110
Total			450

Task Knjige

Tin is a very special boy. He doesn't like a lot of things, for example he doesn't like Barcelona, getting defeated in video games or any sort of mess...

Today he is visiting his friend Ante to once and for all decide who is the best FIFA player. The moment he entered Ante's apartment, he was greeted with an unpleasant surprise. Ante has two shelves on his wall side by side. The left one contains n books about the numerous accomplishments of Barcelona **stacked on top of each other**, and the right one is empty.

He didn't mind so much that Ante was reading, in his opinion, trashy books, but what bothered him much more was that the books were a total mess, that is, they weren't sorted from thinnest to thickest. As Ante is a good friend, he immediately hurried to rearrange the books to Tin's liking. In one move he can:

- **take a book from the top** of some shelf in his left or right hand, if he is not holding some other book in that hand; or
- **put the book** he is holding in some hand **on top** of some shelf.

Ante's strong suit is football, not rearranging books, so he asks you to find some sequence of moves, that he will then perform, so that in the end all books will be on the **left shelf** and sorted **from thinnest to thickest**, in the order **from top to bottom**.

Input

The first line contains an integer n ($1 \leq n \leq 100$), the number of books on the left shelf.

The second line contains n integers d_i ($1 \leq d_i \leq 1000$) that represent the thicknesses of the books, from top to bottom.

Output

In the first line output an integer k ($0 \leq k \leq 100\,000$), the number of moves in your solution.

In the following k lines output the moves in the form `INSTRUCTION HAND SHELF`, where:

- `INSTRUCTION` is the word `UZMI` (Croatian for *take*) if Ante should take a book from some shelf, or the word `STAVI` (Croatian for *put*) if he should put a book on some shelf
- `HAND` is the letter `L` if Ante should use his left hand, or the letter `D` (Croatian word for *right* is *desno*) if he should use his right hand
- `SHELF` is the letter `L` if this move acts on the left shelf, or the letter `D` if it acts on the right shelf.

Your solution does **not** have to be of minimum possible length, but the number of moves must not exceed 100 000. It can be proven that for the given constraints a solution always exists.

Examples

input

```
3
2 3 1
```

output

```
8
UZMI L L
STAVI L D
UZMI L L
UZMI D L
STAVI L L
UZMI L D
STAVI L L
STAVI D L
```

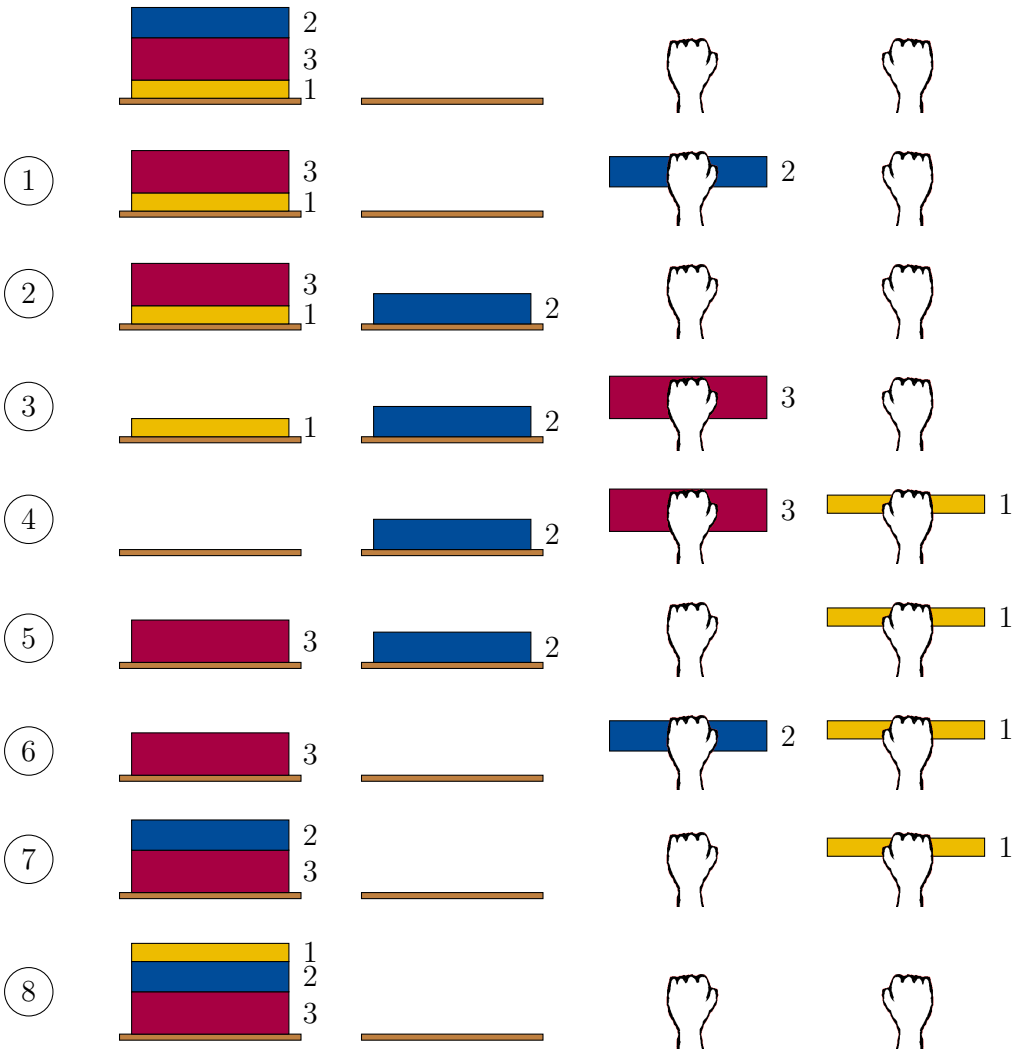
input

```
4
1 1 2 5
```

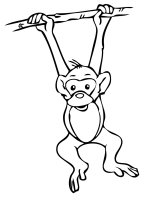
output

```
0
```

Clarification of the first example:



Task Vlak



Nina and Emilija are playing a game on a piece of paper. Initially, the paper is empty. In one move a player appends a letter to the end of the word that is currently written on the paper. They alternate turns, and Nina plays first.

Players must choose the letters in such a way that the following condition is met: the word that is written **after** the player's move must be a prefix of some word in that players favourite song. If the player can't make a move, she loses.

If both players play optimally, determine who wins.

Input

The first line contains a positive integer n, the number of words in Nina's favourite song.
Each of the following n lines contains a word from Nina's favourite song.

The following line contains a positive integer m, the number of words in Emilija's favourite song.
Each of the following m lines contains a word from Emilija's favourite song.

Words in input contain only lowercase letters, and the sum of the lengths of all words is at most 200 000.

Output

Output `Nina` or `Emilija`, the name of the winner.

Scoring

In test cases worth 40 points the sum of the lengths of the words will be at most 2000.

Examples

input

```
2
aaa
bbb
3
aab
aba
bbb
```

output

```
Nina
```

input

```
2
acg
beh
2
adi
bfj
```

output

```
Emilija
```

input

```
3
ja
sam
vlak
5
sto
zgazit
ce
te
mali
```

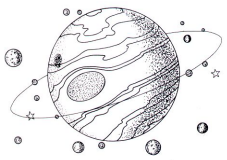
output

```
Nina
```

Clarification of the first example:

If Nina first writes `b`, Emilija must write `b`, and then Nina can write `b`. The current word is `bbb`, and Emilija can't make a move, so Nina wins.
If Nina would first write `a`, Emilija could write `b`. The word would be `ab`, and Nina wouldn't be able to make a move, and she would lose.

Task Sateliti



For crater exploration purposes, the Arecibo telescope records images of Saturn's satellites. The scientific team must distinguish between satellite images and group the images by satellite, but it's not that simple because satellites could be shot from different angles.

Captured images can be displayed as $n \times m$ matrices, filled with `'*'` (crater) and `'.'` (plain surface). We say that two images correspond to the same satellite if one can get the other by **circular shifts of rows and columns**.

To make the verification process easier, scientists want to find the **lexicographically smallest** image corresponding to the satellite from the given image. When comparing images, we compare strings obtained by concatenating all rows of the image, where characters are compared by ASCII value.

Input

The first line contains integers n and m ($1 \leq n, m \leq 1000$), the dimensions of the image.

Each of the following n lines contains m characters `'*'` and `'.'`. This represents the captured image.

Output

Output n lines with m characters each, the wanted lexicographically smallest image.

Scoring

Subtask	Points	Constraints
1	10	$1 \leq n, m \leq 50$
2	40	$1 \leq n, m \leq 300$
3	60	No additional constraints.

Examples

input

```
3 3
.**
*..
.*.
```

output

```
**.
..*
*..
```

input

```
3 4
....
..*.
....
```

output

```
*...
....
....
```

input

```
3 5
.**..
.***.
..**.
```

output

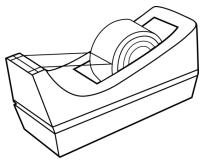
```
***..
.**..
**...
```

Clarification of the first example:

All images that can be obtained by circular shifts are:

```
.**   .*.   *..   **.   *..   ..*   *.*   ..*   .*.
*..   .**   .*.   ..*   **.   *..   .*.   *.*   ..*
.*.   *..   .**   *..   ..*   **.   ..*   .*.   *.*
```

Task Selotejp



For Mirko there is no greater happiness than finding a new roll of sticky tape, and today he is especially happy because he had also found Slavko's Advent calendar.

The Advent calendar can be represented as a table with n rows and m columns. Each square contains a little window, and behind each window is a piece of chocolate. Slavko had already opened some of the windows, and others are still closed.

Mirko decided to use his sticky tape to glue all closed windows shut. The tape is infinitely long, and it is one calendar cell wide. Mirko can rip off a piece of tape and use it to glue some **sequence of horizontally or vertically adjacent closed windows** shut. He doesn't want to put more than one piece of tape over some window, since he wants to remain friends with Slavko.

He is wondering what is the **minimal** number of pieces of tape he needs to glue **all** closed windows shut.

Input

The first line contains integers n and m ($1 \leq n \leq 1000$, $1 \leq m \leq 10$), dimensions of the Advent calendar.

Each of the following n lines contains m characters `'.'` and `'#'` that represent the Advent calendar. The character `'.'` denotes an open window, and the character `'#'` denotes a closed window.

Output

Output the minimal number of pieces of tape needed to glue all closed windows shut.

Scoring

Subtask	Points	Constraints
1	35	Each closed window is adjacent to (shares a side with) at most two other closed windows.
2	35	$1 \leq n \leq 10$
3	40	No additional constraints.

Examples

input

```
2 3
#.#
###
```

output

```
3
```

input

```
4 3
.#.
###
.##
.#.
```

output

```
3
```

input

```
4 4
####
#.#.
#.##
####
```

output

```
5
```

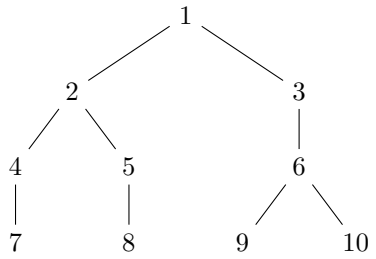
Clarification of the first example:

One possible solution is to use one piece of tape for the first column, one piece for the third column, and one piece for the window in the second row and second column.

Task Specijacija

You are given a positive integer n and a sequence $a_1, a_2, \ldots, a_n$ of positive integers, such that $\frac{i(i-1)}{2} < a_i \leq \frac{i(i+1)}{2}$.

The sequence parameterizes a tree with $\frac{(n+1)(n+2)}{2}$ vertices, consisting of $n+1$ levels with $1, 2, \ldots, n+1$ vertices, in the following way:



The tree parameterized by $a = (1, 2, 6)$.

The i-th level contains vertices $\frac{i(i-1)}{2} + 1, \ldots, \frac{i(i+1)}{2}$. The vertex a_i has two children, and the rest of the vertices on the level have one child each.

We want to answer q queries of the form "what is the largest common ancestor of x and y", i.e. the vertex with the largest label which is an ancestor of both x and y.

Input

The first line contains integers n, q and t ($1 \leq n, q \leq 200\,000$, $t \in \{0, 1\}$), the number of parameters, the number of queries, and a value which will be used to determine the labels of vertices in the queries.

The second line contains a sequence of n integers a_i ($\frac{i(i-1)}{2} < a_i \leq \frac{i(i+1)}{2}$) which parameterize the tree.

The i-th of the following q lines contains two integers $\widetilde{x}_i$ and $\widetilde{y}_i$ ($1 \leq \widetilde{x}_i, \widetilde{y}_i \leq \frac{(n+1)(n+2)}{2}$) which will be used to determine the labels of vertices in the queries.

Let z_i be the answer to the i-th query, and let $z_0 = 0$. The labels in the i-th query x_i and y_i are:

$$x_i = \left((\widetilde{x}_i - 1 + t \cdot z_{i-1}) \bmod \tfrac{(n+1)(n+2)}{2}\right) + 1,$$
$$y_i = \left((\widetilde{y}_i - 1 + t \cdot z_{i-1}) \bmod \tfrac{(n+1)(n+2)}{2}\right) + 1,$$

where mod is the remainder of integer divison.

Remark: Note that if $t = 0$, it holds $x_i = \widetilde{x}_i$ and $y_i = \widetilde{y}_i$, so all queries are known from input. If $t = 1$, the queries are not known in advance, but are determined using answers to previous queries.

Output

Output q lines. In the i-th line, output the largest common ancestor of x_i and y_i.

Scoring

Subtask	Points	Constraints
1	10	$q = 1$, $t = 0$
2	10	$n \leq 1000$, $t = 0$
3	30	$t = 0$
4	60	$t = 1$

Examples

input

```
3 5 0
1 2 6
7 10
8 5
6 2
9 10
2 3
```

output

```
1
5
1
6
1
```

input

```
3 5 1
1 2 6
7 10
8 5
6 2
9 10
2 3
```

output

```
1
6
2
1
1
```

Clarification of the examples:

The tree from both examples is shown on the figure in the statement.

Labels of verticies in queries in the second example are:
$x_1 = 7,\ y_1 = 10,$
$x_2 = 9,\ y_2 = 6,$
$x_3 = 2,\ y_3 = 8,$
$x_4 = 1,\ y_4 = 2,$
$x_5 = 3,\ y_5 = 4.$