A. Xenia and Divisors

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

Xenia the mathematician has a sequence consisting of n (n is divisible by 3) positive integers, each of them is at most 7. She wants to split the sequence into groups of three so that for each group of three a, b, c the following conditions held:

- $a < b < c$;
- a divides b, b divides c.

Naturally, Xenia wants each element of the sequence to belong to exactly one group of three. Thus, if the required partition exists, then it has $\frac{n}{3}$ groups of three.

Help Xenia, find the required partition or else say that it doesn't exist.

Input

The first line contains integer n $(3 \le n \le 99999)$ — the number of elements in the sequence. The next line contains n positive integers, each of them is at most 7.

It is guaranteed that n is divisible by 3.

Output

If the required partition exists, print $\frac{n}{3}$ groups of three. Print each group as values of the elements it contains. You should print values in increasing order. Separate the groups and integers in groups by whitespaces. If there are multiple solutions, you can print any of them.

If there is no solution, print -1.

Sample test(s)

input

```
6
1 1 1 2 2 2
```

output

```
-1
```

input

```
6
2 2 1 1 4 6
```

output

```
1 2 4
1 2 6
```

B. Xenia and Spies

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

Xenia the vigorous detective faced n $(n \geq 2)$ foreign spies lined up in a row. We'll consider the spies numbered from 1 to n from left to right.

Spy s has an important note. He has to pass the note to spy f. Xenia interrogates the spies in several steps. During one step the spy keeping the important note can pass the note to one of his neighbours in the row. In other words, if this spy's number is x, he can pass the note to another spy, either x - 1 or $x + 1$ (if $x = 1$ or $x = n$, then the spy has only one neighbour). Also during a step the spy can keep a note and not pass it to anyone.

But nothing is that easy. During m steps Xenia watches some spies attentively. Specifically, during step t_i (steps are numbered from 1) Xenia watches spies numbers $l_i, l_i + 1, l_i + 2, ..., r_i$ $(1 \leq l_i \leq r_i \leq n)$. Of course, if during some step a spy is watched, he can't do anything: neither give the note nor take it from some other spy. Otherwise, Xenia reveals the spies' cunning plot. Nevertheless, if the spy at the current step keeps the note, Xenia sees nothing suspicious even if she watches him.

You've got s and f. Also, you have the steps during which Xenia watches spies and which spies she is going to watch during each step. Find the best way the spies should act in order to pass the note from spy s to spy f as quickly as possible (in the minimum number of steps).

Input

The first line contains four integers n, m, s and f $(1 \leq n, m \leq 10^5; 1 \leq s, f \leq n; s \neq f; n \geq 2)$. Each of the following m lines contains three integers t_i, l_i, r_i $(1 \leq t_i \leq 10^9, 1 \leq l_i \leq r_i \leq n)$. It is guaranteed that $t_1 < t_2 < t_3 < ... < t_m$.

Output

Print k characters in a line: the i-th character in the line must represent the spies' actions on step i. If on step i the spy with the note must pass the note to the spy with a lesser number, the i-th character should equal "`L`". If on step i the spy with the note must pass it to the spy with a larger number, the i-th character must equal "`R`". If the spy must keep the note at the i-th step, the i-th character must equal "`X`".

As a result of applying the printed sequence of actions spy s must pass the note to spy f. The number of printed characters k must be as small as possible. Xenia must not catch the spies passing the note.

If there are miltiple optimal solutions, you can print any of them. It is guaranteed that the answer exists.

Sample test(s)

input

```
3 5 1 3
1 1 2
2 2 3
3 3 3
4 1 1
10 1 3
```

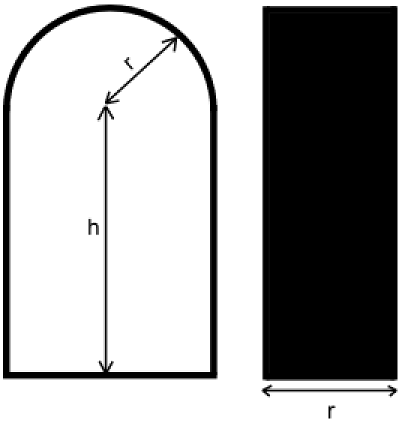
output

```
XXRR
```

C. Cupboard and Balloons

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

A girl named Xenia has a cupboard that looks like an arc from ahead. The arc is made of a semicircle with radius r (the cupboard's top) and two walls of height h (the cupboard's sides). The cupboard's depth is r, that is, it looks like a rectangle with base r and height $h + r$ from the sides. The figure below shows what the cupboard looks like (the front view is on the left, the side view is on the right).



Xenia got lots of balloons for her birthday. The girl hates the mess, so she wants to store the balloons in the cupboard. Luckily, each balloon is a sphere with radius $\frac{r}{2}$. Help Xenia calculate the maximum number of balloons she can put in her cupboard.

You can say that a balloon is in the cupboard if you can't see any part of the balloon on the left or right view. The balloons in the cupboard can touch each other. It is not allowed to squeeze the balloons or deform them in any way. You can assume that the cupboard's walls are negligibly thin.

Input

The single line contains two integers r, h $(1 \le r, h \le 10^7)$.

Output

Print a single integer — the maximum number of balloons Xenia can put in the cupboard.

Sample test(s)

```
input
1 1
output
3
```

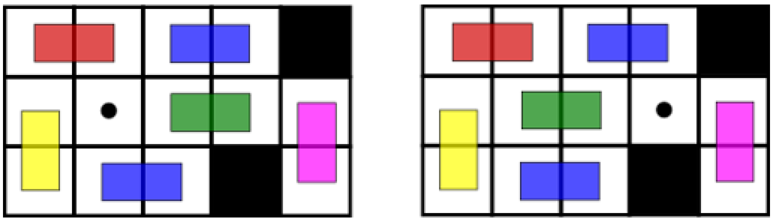
```
input
1 2
output
5
```

```
input
2 1
output
2
```

D. Xenia and Dominoes

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

Xenia likes puzzles very much. She is especially fond of the puzzles that consist of domino pieces. Look at the picture that shows one of such puzzles.



A puzzle is a $3 \times n$ table with forbidden cells (black squares) containing dominoes (colored rectangles on the picture). A puzzle is called *correct* if it meets the following conditions:

- each domino occupies exactly two non-forbidden cells of the table;
- no two dominoes occupy the same table cell;
- exactly one non-forbidden cell of the table is unoccupied by any domino (it is marked by a circle in the picture).

To solve the puzzle, you need multiple steps to transport an empty cell from the starting position to some specified position. A move is transporting a domino to the empty cell, provided that the puzzle stays correct. **The horizontal dominoes can be moved only horizontally, and vertical dominoes can be moved only vertically. You can't rotate dominoes.** The picture shows a probable move.

Xenia has a $3 \times n$ table with forbidden cells and a cell marked with a circle. Also, Xenia has very many identical dominoes. Now Xenia is wondering, how many distinct correct puzzles she can make if she puts dominoes on the existing table. Also, Xenia wants the circle-marked cell to be empty in the resulting puzzle. The puzzle must contain at least one move.

Help Xenia, count the described number of puzzles. As the described number can be rather large, print the remainder after dividing it by 1000000007 $(10^9 + 7)$.

Input

The first line contains integer n $(3 \le n \le 10^4)$ — the puzzle's size. Each of the following three lines contains n characters — the description of the table. The j-th character of the i-th line equals "X" if the corresponding cell is forbidden; it equals ".", if the corresponding cell is non-forbidden and "O", if the corresponding cell is marked with a circle.

It is guaranteed that exactly one cell in the table is marked with a circle. It is guaranteed that all cells of a given table having at least one common point with the marked cell is non-forbidden.

Output

Print a single number — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
5
....X
.O...
...X.
```

output

```
1
```

input

```
5
.....
.O...
.....
```

output

```
2
```

input

```
3
...
...
..O
```

output

```
4
```

Note

Two puzzles are considered distinct if there is a pair of cells that contain one domino in one puzzle and do not contain it in the other one.

E. Xenia and Tree

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Xenia the programmer has a tree consisting of n nodes. We will consider the tree nodes indexed from 1 to n. We will also consider the first node to be initially painted red, and the other nodes — to be painted blue.

The *distance* between two tree nodes v and u is the number of edges in the shortest path between v and u.

Xenia needs to learn how to quickly execute queries of two types:

1. paint a specified blue node in red;
2. calculate which red node is the closest to the given one and print the shortest distance to the closest red node.

Your task is to write a program which will execute the described queries.

Input

The first line contains two integers n and m $(2 \le n \le 10^5, 1 \le m \le 10^5)$ — the number of nodes in the tree and the number of queries. Next n - 1 lines contain the tree edges, the i-th line contains a pair of integers a_i, b_i $(1 \le a_i, b_i \le n, a_i \ne b_i)$ — an edge of the tree.

Next m lines contain queries. Each query is specified as a pair of integers t_i, v_i $(1 \le t_i \le 2, 1 \le v_i \le n)$. If $t_i = 1$, then as a reply to the query we need to paint a blue node v_i in red. If $t_i = 2$, then we should reply to the query by printing the shortest distance from some red node to node v_i.

It is guaranteed that the given graph is a tree and that all queries are correct.

Output

For each second type query print the reply in a single line.

Sample test(s)

input

```
5 4
1 2
2 3
2 4
4 5
2 1
2 5
1 2
2 5
```

output

```
0
3
2
```