Croatian Olympiad in Informatics

Zagreb, March 25^{th} 2017

Tasks

Task	Time limit	Memory limit	Score
Ili	4 seconds	1024 MiB	100
Raspad	6 seconds	1024 MiB	100
Trapezi	0.5 seconds	1024 MiB	100
Zagrade	3 seconds	1024 MiB	100
Total			400

ministarstvo znanosti i obrazovanja

HRVATSKI SAVEZ INFORMATIČARA

HRVATSKA ZAJEDNICA TEHNIČKE KULTURE

Task: Ili

Mirko is building a simple logic circuit in his workshop. The circuit consists of n *starting wires* denoted with x_1, x_2, ..., x_n and m logic elements OR denoted with c_1, c_2, ..., c_m. Each element has exactly two *inputs* and one *output*. Each of the inputs is connected to either a starting wire x_j or to the output of another element c_j. Of course, there are no cycles in a logic circuit and, moreover, it holds that the input of c_j can be connected to the output of c_i only when it holds $i < j$.

Each starting wire in the circuit can be set to value 0 or 1, and the value of the output of each element is the logic OR operation of its inputs — the value is 0 if the values of both inputs are 0, otherwise it is 1.

Mirko does not know the initial values of the starting wires, but with careful measurements, he has determined the values of the output of some elements. Find the remaining values of the outputs that can be unambiguously determined based on the measurements.

Input

The first line of input contains the positive integers n and m — the number of starting wires and the number of elements in the circuit. The following line contains a string of exactly m characters that describes the measured value of the output of the element c_j, or is equal to "`?`" if Mirko did not perform this measurement. The j^{th} of the following m lines contains labels of two inputs of the circuit c_j, each label being either a label of the starting wire in the form of "`x`i" where it holds $1 \le i \le n$, or a label of the element "`c`i" where it holds $1 \le i < j$. The two inputs of the element c_j may be the same. You can assume that the measured values are mutually consistent.

Output

The first line of output must contain a string of m characters — the j^{th} character in the string must correspond to the value of the output of c_j or be "`?`" if that value cannot be unambiguously determined.

Scoring

Subtask	Score	Constraints
1	7	$n \le 15, m \le 20$
2	42	$n \le 500, m \le 500$
3	51	$n \le 10\,000, m \le 10\,000$

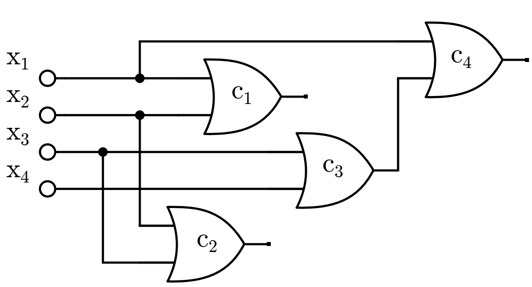
Sample tests

input

```
4 4
10??
x1 x2
x2 x3
x3 x4
x1 c3
```

output

```
10?1
```



input

```
4 5
11???
x1 x2
x3 x4
x1 x3
x2 x4
c3 c4
```

output

```
11??1
```

Task: Raspad

A nearby meadow consists of quadratic *fields* organized in n rows and m columns. The rows are denoted with numbers from 1 to n from top to bottom, and the columns with numbers from 1 to m from left to right. Some fields are grass fields (denoted with "`1`"), whereas some are underwater because of the heavy spring rainfall (denoted with "`0`"). Two grass fields are *connected* if it is possible to get from one field to another using a series of moves where, in each step, we move to the adjacent grass field located up, down, left or right. A *component* is a set of mutually connected grass fields that is *maximal* in the sense that, if A is a field in the component K, then all the adjacent grass fields of A are also in the component K.

For a given meadow P and indices a and b ($1 \le a \le b \le n$), P^a_b is a meadow consisting of rows between the a^{th} and the b^{th} row of the original meadow P (including both a^{th} and b^{th} row). The *complexity* of meadow P^a_b is the number of components of the grass fields located on the meadow. Determine the sum of the complexities of all possible meadows P^a_b.

Input

The first line of input contains the positive integers n and m — dimensions of the meadow. Each of the following n lines contains a string of exactly m characters that denotes one row of the meadow. Each character of the string is either the digit "`0`" or the digit "`1`".

Output

You must output the required sum of all complexities.

Scoring

Subtask	Score	Constraints
1	9	$n \le 100$, $m \le 50$
2	17	$n \le 1\,000$, $m \le 50$
3	35	$n \le 100\,000$, $m \le 15$
4	39	$n \le 100\,000$, $m \le 50$

Sample tests

input

```
4 4
1101
1111
1010
1011
```

output

```
14
```

input

```
5 7
0100010
0111110
0101001
1111011
0100100
```

output

```
33
```

input

```
4 12
011111010111
110000101001
110111101111
111101111111
```

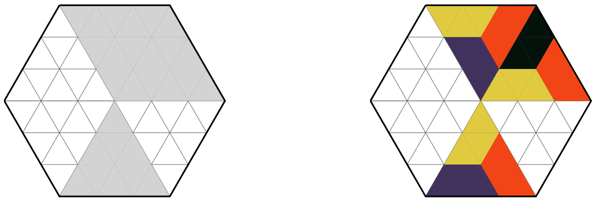
output

```
28
```

Explanation of the first sample: If we denote the complexity of meadow P^a_b with $|P^a_b|$ then it holds that $|P^1_1| = 2$, $|P^1_2| = 1$, $|P^1_3| = 1$, $|P^1_4| = 1$, $|P^2_2| = 1$, $|P^2_3| = 1$, $|P^2_4| = 1$, $|P^3_3| = 2$, $|P^3_4| = 2$, $|P^4_4| = 2$, and the sum of these numbers is 14.

Task: Trapezi

We can create a *hexagonal puzzle* the size of n by dividing a regular hexagon into equilateral triangles by drawing $2n - 1$ equidistant parallel lines between each three pairs of opposite hexagon sides. Some of the triangles in the puzzle are shaded and need to be covered with *puzzle pieces*. Each piece is a trapezoid that consists of three equilateral triangles placed side by side. The pieces come in 6 different colours denoted with numbers from 1 to 6, and we have an unlimited number of pieces of each colour at our disposal.



Slika 1: Puzzle of size 3 from the first sample and one solution.

The goal of the puzzle is to put the pieces on the hexagon so that the following holds:

1. Each piece is placed so it fully covers three shaded triangles.
2. Each shaded triangle is covered by exactly one piece.
3. Two pieces of the same colour do not touch along the side of a triangle (they may touch in a corner).

Determine if it is possible to solve the given puzzle, and, if it is, find one solution.

Input

The first line of input contains the positive integer n — the size of the puzzle. The following $2n$ lines describe the rows of the puzzle from top to bottom. Each of these lines contains a string that describes the triangles in one row of the puzzle from left to right. The digit "`0`" denotes a shaded triangle, whereas "`.`" (dot) denotes a triangle that is not shaded. You can assume that at least one triange will be shaded.

Output

If the puzzle is impossible to solve, output in the first line "`nemoguce`" (Croatian for impossible). Otherwise, output $2n$ lines that describe the solution in the same format as the puzzle is given in the input. Shaded triangles should be denoted with one of the digits from "`1`" to "`6`", instead of the digit "`0`". The digits represent the colour of the pieces the triangle is covered with.

Scoring

Subtask	Score	Constraints
1	6	$n = 1$
2	17	$n = 2$
3	18	$n = 3$
4	22	$n = 4$
5	37	$n = 5$

Sample tests

input

```
3
.000000
...000000
.....000000
.....0.....
...000...
.00000.
```

output

```
.111224
...332442
.....311122
.....1.....
...112...
.33322.
```

input

```
1
.0.
0.0
```

output

```
nemoguce
```

input

```
2
0000.
0000000
..00.0.
.0000
```

output

```
1222.
1133111
..31.2.
.1122
```

Task: Zagrade

An *expression* is a string of consisting only of properly paired brackets. For example, "`()()`" and "`(()())`" are expressions, whereas "`)(`" and "`()(`" are not. We can define expressions inductively as follows:

- "`()`" is an expression.
- If a is an expression, then "(a)" is also an expression.
- If a and b are expressions, then "ab" is also an expression.

A *tree* is a structure consisting of n *nodes* denoted with numbers from 1 to n and $n-1$ *edges* placed so there is a unique path between each two nodes. Additionally, a single character is written in each node. The character is either an open bracket "`(`" or a closed bracket "`)`". For different nodes a and b, $w_{a,b}$ is a string obtained by traversing the unique path from a to b and, one by one, adding the character written in the node we're passing through. The string $w_{a,b}$ also contains the character written in the node a (at the first position) and the character written in the node b (at the last position).

Find the total number of pairs of different nodes a and b such that $w_{a,b}$ is a correct expression.

Input

The first line of contains the an integer n — the number of nodes in the tree. The following line contains an n-character string where each character is either "`)`" or "`(`", the j^{th} character in the string is the character written in the node j. Each of the following $n-1$ lines contains two different positive integers x and y ($1 \leq x, y \leq n$) — the labels of nodes directly connected with an edge.

Output

Output the required number of pairs.

Scoring

Subtask	Score	Constraints
1	10	$n \leq 1\,000$
2	30	$n \leq 300\,000$, the tree is a *chain* — each node $x = 1, \ldots, n-1$ is connected to node $x+1$.
3	60	$n \leq 300\,000$

Sample tests

input

```
4
(())
1 2
2 3
3 4
```

output

```
2
```

input

```
5
())((
1 2
2 3
2 4
3 5
```

output

```
3
```

input

```
7
)()()((
1 2
1 3
1 6
2 4
4 5
5 7
```

output

```
6
```