A. Valera and X

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

Valera is a little boy. Yesterday he got a huge Math hometask at school, so Valera didn't have enough time to properly learn the English alphabet for his English lesson. Unfortunately, the English teacher decided to have a test on alphabet today. At the test Valera got a square piece of squared paper. The length of the side equals n squares (n is an odd number) and each unit square contains some small letter of the English alphabet.

Valera needs to know if the letters written on the square piece of paper form letter "X". Valera's teacher thinks that the letters on the piece of paper form an "X", if:

- on both diagonals of the square paper all letters are the same;
- all other squares of the paper (they are not on the diagonals) contain the same letter that is different from the letters on the diagonals.

Help Valera, write the program that completes the described task for him.

Input

The first line contains integer n ($3 \le n < 300$; n is odd). Each of the next n lines contains n small English letters — the description of Valera's paper.

Output

Print string "YES", if the letters on the paper form letter "X". Otherwise, print string "NO". Print the strings without quotes.

Examples

input

```
5
xooox
oxoxo
soxoo
oxoxo
xooox
```

output

```
NO
```

input

```
3
wsw
sws
wsw
```

output

```
YES
```

input

```
3
xpx
pxp
xpe
```

output

```
NO
```

B. Marathon

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

Valera takes part in the Berland Marathon. The marathon race starts at the stadium that can be represented on the plane as a square whose lower left corner is located at point with coordinates $(0, 0)$ and the length of the side equals a meters. The sides of the square are parallel to coordinate axes.

As the length of the marathon race is very long, Valera needs to have extra drink during the race. The coach gives Valera a bottle of drink each d meters of the path. We know that Valera starts at the point with coordinates $(0, 0)$ and runs counter-clockwise. That is, when Valera covers a meters, he reaches the point with coordinates $(a, 0)$. We also know that the length of the marathon race equals $nd + 0.5$ meters.

Help Valera's coach determine where he should be located to help Valera. Specifically, determine the coordinates of Valera's positions when he covers $d, 2 \cdot d, \ldots, n \cdot d$ meters.

Input

The first line contains two space-separated real numbers a and d $(1 \le a, d \le 10^5)$, given with precision till 4 decimal digits after the decimal point. Number a denotes the length of the square's side that describes the stadium. Number d shows that after each d meters Valera gets an extra drink.

The second line contains integer n $(1 \le n \le 10^5)$ showing that Valera needs an extra drink n times.

Output

Print n lines, each line should contain two real numbers x_i and y_i, separated by a space. Numbers x_i and y_i in the i-th line mean that Valera is at point with coordinates (x_i, y_i) after he covers $i \cdot d$ meters. Your solution will be considered correct if the absolute or relative error doesn't exceed 10^{-4}.

Note, that this problem have huge amount of output data. Please, do not use cout stream for output in this problem.

Examples

input

```
2 5
2
```

output

```
1.0000000000 2.0000000000
2.0000000000 0.0000000000
```

input

```
4.147 2.8819
6
```

output

```
2.8819000000 0.0000000000
4.1470000000 1.6168000000
3.7953000000 4.1470000000
0.9134000000 4.1470000000
0.0000000000 2.1785000000
0.7034000000 0.0000000000
```

C. Restore Graph

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

Valera had an undirected connected graph without self-loops and multiple edges consisting of n vertices. The graph had an interesting property: there were at most k edges adjacent to each of its vertices. For convenience, we will assume that the graph vertices were indexed by integers from 1 to n.

One day Valera counted the shortest distances from one of the graph vertices to all other ones and wrote them out in array d. Thus, element $d[i]$ of the array shows the shortest distance from the vertex Valera chose to vertex number i.

Then something irreparable terrible happened. Valera lost the initial graph. However, he still has the array d. Help him restore the lost graph.

Input

The first line contains two space-separated integers n and k $(1 \le k < n \le 10^5)$. Number n shows the number of vertices in the original graph. Number k shows that at most k edges were adjacent to each vertex in the original graph.

The second line contains space-separated integers $d[1], d[2], \dots, d[n]$ $(0 \le d[i] < n)$. Number $d[i]$ shows the shortest distance from the vertex Valera chose to the vertex number i.

Output

If Valera made a mistake in his notes and the required graph doesn't exist, print in the first line number -1. Otherwise, in the first line print integer m $(0 \le m \le 10^6)$ — the number of edges in the found graph.

In each of the next m lines print two space-separated integers a_i and b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$, denoting the edge that connects vertices with numbers a_i and b_i. The graph shouldn't contain self-loops and multiple edges. If there are multiple possible answers, print any of them.

Examples

input

```
3 2
0 1 1
```

output

```
3
1 2
1 3
3 2
```

input

```
4 2
2 0 1 3
```

output

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

input

```
3 1
0 0 0
```

output

```
-1
```

D. Minesweeper 1D

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Game "Minesweeper 1D" is played on a line of squares, the line's height is 1 square, the line's width is n squares. Some of the squares contain bombs. If a square doesn't contain a bomb, then it contains a number from 0 to 2 — the total number of bombs in adjacent squares.

For example, the correct field to play looks like that: `001*2***101*`. The cells that are marked with "*" contain bombs. Note that on the correct field the numbers represent the number of bombs in adjacent cells. For example, field `2*` is not correct, because cell with value 2 must have two adjacent cells with bombs.

Valera wants to make a correct field to play "Minesweeper 1D". He has already painted a squared field with width of n cells, put several bombs on the field and wrote numbers into some cells. Now he wonders how many ways to fill the remaining cells with bombs and numbers are there if we should get a correct field in the end.

Input

The first line contains sequence of characters without spaces $s_1s_2... s_n$ $(1 \leq n \leq 10^6)$, containing only characters "*", "?" and digits "0", "1" or "2". If character s_i equals "*", then the i-th cell of the field contains a bomb. If character s_i equals "?", then Valera hasn't yet decided what to put in the i-th cell. Character s_i, that is equal to a digit, represents the digit written in the i-th square.

Output

Print a single integer — the number of ways Valera can fill the empty cells and get a correct field.

As the answer can be rather large, print it modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
?01???
```

output

```
4
```

input

```
?
```

output

```
2
```

input

```
**12
```

output

```
0
```

input

```
1
```

output

```
0
```

Note

In the first test sample you can get the following correct fields: `001**1`, `001***`, `001*2*`, `001*10`.

E. Maze 1D

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

Valera has a strip infinite in both directions and consisting of cells. The cell numbers are numbered by integers. The cell number 0 has a robot.

The robot has instructions — the sequence of moves that he must perform. In one move, the robot moves one cell to the left or one cell to the right, according to instructions. Before the robot starts moving, Valera puts obstacles in some cells of the strip, excluding cell number 0. If the robot should go into the cell with an obstacle according the instructions, it will skip this move.

Also Valera indicates the finish cell in which the robot has to be after completing the entire instructions. The finishing cell should be different from the starting one. It is believed that the robot *completed the instructions successfully*, if during the process of moving he visited the finish cell exactly once — at its last move. Moreover, the latter move cannot be skipped.

Let's assume that k is the minimum number of obstacles that Valera must put to make the robot able to complete the entire sequence of instructions successfully and end up in some finishing cell. You need to calculate in how many ways Valera can choose k obstacles and the finishing cell so that the robot is able to complete the instructions successfully.

Input

The first line contains a sequence of characters without spaces $s_1 s_2 ... s_n$ $(1 \le n \le 10^6)$, consisting only of letters "L" and "R". If character s_i equals "L", then the robot on the i-th move must try to move one cell to the left. If the s_i-th character equals "R", then the robot on the i-th move must try to move one cell to the right.

Output

Print a single integer — the required number of ways. It's guaranteed that this number fits into 64-bit signed integer type.

Examples

input

```
RR
```

output

```
1
```

input

```
RRL
```

output

```
1
```

Note

In the first sample Valera mustn't add any obstacles and his finishing cell must be cell 2.

In the second sample, Valera must add an obstacle in cell number 1, and his finishing cell must be cell number -1. In this case robot skips the first two moves and on the third move he goes straight from the starting cell to the finishing one. But if Valera doesn't add any obstacles, or adds an obstacle to another cell, then the robot visits the finishing cell more than once.