D. Block Tower

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

After too much playing on paper, Iahub has switched to computer games. The game he plays is called "Block Towers". It is played in a rectangular grid with n rows and m columns (it contains $n \times m$ cells). The goal of the game is to build your own city. Some cells in the grid are big holes, where Iahub can't build any building. The rest of cells are empty. In some empty cell Iahub can build exactly one tower of two following types:

1. Blue towers. Each has population limit equal to 100.
2. Red towers. Each has population limit equal to 200. However, it can be built in some cell only if in that moment at least one of the neighbouring cells has a Blue Tower. Two cells are neighbours is they share a side.

Iahub is also allowed to destroy a building from any cell. He can do this operation as much as he wants. After destroying a building, the other buildings are not influenced, and the destroyed cell becomes empty (so Iahub can build a tower in this cell if needed, see the second example for such a case).

Iahub can convince as many population as he wants to come into his city. So he needs to configure his city to allow maximum population possible. Therefore he should find a sequence of operations that builds the city in an optimal way, so that total population limit is as large as possible.

He says he's the best at this game, but he doesn't have the optimal solution. Write a program that calculates the optimal one, to show him that he's not as good as he thinks.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 500$). Each of the next n lines contains m characters, describing the grid. The j-th character in the i-th line is '`.`' if you're allowed to build at the cell with coordinates (i, j) a tower (empty cell) or '`#`' if there is a big hole there.

Output

Print an integer k in the first line $(0 \le k \le 10^6)$ — the number of operations Iahub should perform to obtain optimal result.

Each of the following k lines must contain a single operation in the following format:

1. «`B x y`» $(1 \le x \le n, 1 \le y \le m)$ — building a blue tower at the cell (x, y);
2. «`R x y`» $(1 \le x \le n, 1 \le y \le m)$ — building a red tower at the cell (x, y);
3. «`D x y`» $(1 \le x \le n, 1 \le y \le m)$ — destroying a tower at the cell (x, y).

If there are multiple solutions you can output any of them. Note, that you shouldn't minimize the number of operations.

Examples

input

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

output

```
4
B 1 1
R 1 2
R 2 1
B 2 3
```

input

```
1 3
...
```

output

```
5
B 1 1
B 1 2
R 1 3
D 1 2
R 1 2
```