COCI '17 Contest 3 #3 Retro

Time limit: 0.5s **Memory limit:** 512M

Little Mirko got a video game console for Christmas. It wasn't a *Playstation 4* or an *Xbox one*, but *Atari 2600*, and it came with one free game. The protagonist of the game is standing on the bottom of the screen, and there are various objects dispersed on the rest of the screen, falling towards the bottom.

More precisely, the screen can be represented as a grid of $R \times S$ pixels arranged in R rows and S columns. The protagonist takes up one pixel of the lowest line and is marked with `M`. The rest of the pixels are marked with some of the characters: `.` (empty space), `*` (bomb), `(` (open bracket) or `)` (closed bracket).

The protagonist can move one pixel to the left or to the right in a single move, but doesn't need to, whereas the rest of the objects simultaneously move one pixel down (possibly out of the screen). When the protagonist finds himself at the same position as one of the brackets, we say that he picked up that bracket and added it at the end of his array of acquired brackets. The protagonist's goal is to acquire the longest possible **valid** bracket expression.

A valid bracket expression is defined inductively in the following way:

- `()` is a valid expression
- If `a` is a valid expression, then `(a)` is a valid expression as well
- If `a` and `b` are valid expressions, then `ab` is a valid expression as well

The game ends when the protagonist finds himself at the same position as the bomb, or when all the objects fall out of the screen.

Input Specification

The first line of input contains the positive integers R and S $(1 \leq R, S \leq 300)$ that represent the dimensions of the screen. Each of the following R lines contains S characters `M`, `.`, `*`, `(` or `)` that represent the initial state of the screen.

Test data will be such that there will always exist at least one valid bracket expression that is possible to acquire.

Output Specification

In the first line, you must output the length of the longest valid bracket expression that Mirko can acquire. In the second line, output that expression. If there are multiple longest valid expressions, output the **lexicographically smallest** one.

Scoring

In test cases worth 25% of total points, it will hold $1 \leq R \leq 15$.

In test cases worth 50% of total points, it will hold $1 \leq R \leq 100$.

If you output the correct length, but the wrong expression, you will be awarded 40% of points for that test case. In any case, in order to score points, your output **must** consist of two non-empty lines.

Sample Input 1

```
5 4
..).
.)(.
(.)*
*(.*
..M.
```

Sample Output 1

```
4
(())
```

Explanation for Sample Output 1

The protagonist's moves are: `left`, `left`, `right` `right`.

Sample Input 2

```
6 3
)(.
*..
(**
)()
().
M..
```

Sample Output 2

```
4
()()
```

Explanation for Sample Output 2

The protagonist's moves are: `stay still`, `stay still`, `stay still`, `right`, `left`.

Sample Input 3

```
6 3
((.
*..
(**
)()
().
M..
```

Sample Output 3

```
2
()
```

Explanation for Sample Output 3

The protagonist's moves are: `stay still`, `stay still`, `right`.