Problem D. Long puzzle

Time limit: 2 seconds

You have a one-dimensional puzzle. Every piece of the puzzle can be described by three values: length, type of the left border, and type of the right border. Borders can be one of three types: straight, convex, and concave. Pieces couldn't be reversed, i.e. you can't swap left and right borders of a piece. Any convex border can be connected with any concave border and vice versa. You can't connect pieces by two straight borders.

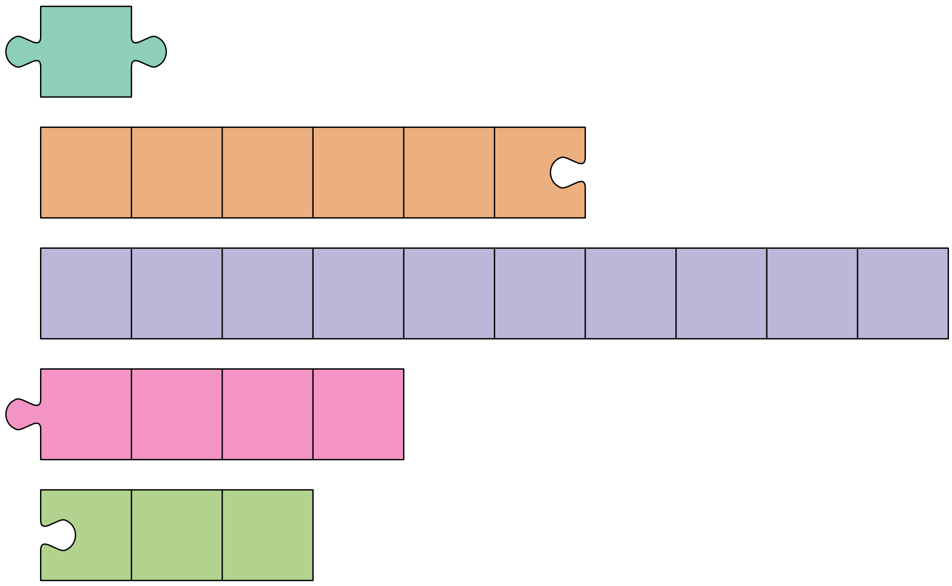

Figure 1: Example of pieces

You want to connect several (possibly one) pieces one after another in order to get a part of length l. The left and the right borders of the part should be straight. Find a number of sets of pieces, such that you can build desired part using all pieces from the set. The number could be large, so calculate it modulo 1 000 000 007. Note that you should find the number of sets of pieces, not the number of different ways of connecting them.

Input

The first line contains two integer numbers n and l — the number of pieces and desired length of a part ($1 \le n \le 300$, $1 \le l \le 300$).

The following n lines contain a description of the pieces. Every line contains a_i, b_i and c_i — the length of the piece, type of its left border, and type of its right border, accordingly ($1 \le a_i \le l$; $b_i, c_i \in \{$“`in`”,“`out`”,“`none`”$\}$). String “`in`” denotes concave border, “`out`” — convex, “`none`” — straight.

Output

Output single integer — the number of sets of pieces, such that you can build desired part using these pieces, modulo 1 000 000 007.

Scoring

Subtask	Score	Constraints
1	20	$n \le 20$
2	20	$b_i \in \{$“in”, “none”$\}$, $c_i \in \{$“out”, “none”$\}$
3	20	$n, l \le 50$
4	20	$n, l \le 100$
5	20	No additional constraints

Examples

standard input	standard output

`5 10`
`1 out out`
`6 none in`
`10 none none`
`4 out none`
`3 in none`
	`3`

`4 5`
`1 none out`
`1 in out`
`2 in out`
`1 in none`
	`1`

Note

Pieces of the puzzle from the first example correspond to the previous picture.

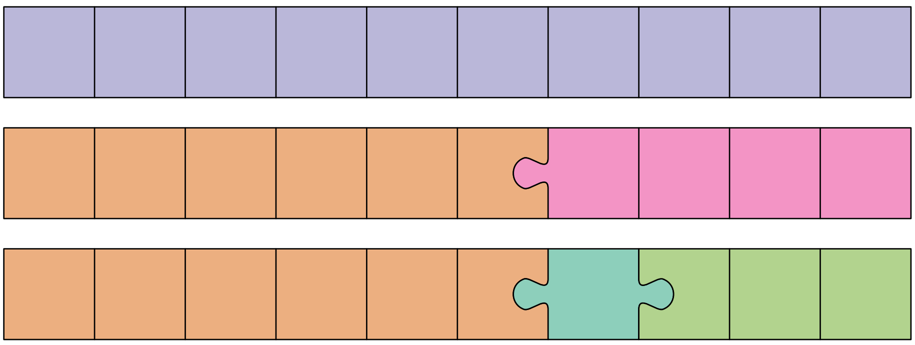

Figure 2: Sets of pieces, such that you can build desired part using them