Problem E. Shifting a Matrix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given $N \times N$ matrix A initialized with $A_{i,j} = (i-1) \cdot N + j$, where $A_{i,j}$ is the entry of the i-th row and the j-th column of A. Note that i and j are 1-based.

You are also given an operation sequence which consists of the four types of shift operations: left, right, up, and down shifts. More precisely, these operations are defined as follows:

- Left shift with i: circular shift of the i-th row to the left, i.e., setting previous $A_{i,k}$ to new $A_{i,k-1}$ for $2 \le k \le N$, and previous $A_{i,1}$ to new $A_{i,N}$.
- Right shift with i: circular shift of the i-th row to the right, i.e., setting previous $A_{i,k}$ to new $A_{i,k+1}$ for $1 \le k \le N-1$, and previous $A_{i,N}$ to new $A_{i,1}$.
- Up shift with j: circular shift of the j-th column to the above, i.e., setting previous $A_{k,j}$ to new $A_{k-1,j}$ for $2 \le k \le N$, and previous $A_{1,j}$ to new $A_{N,j}$.
- Down shift with j: circular shift of the j-th column to the below, i.e., setting previous $A_{k,j}$ to new $A_{k+1,j}$ for $1 \le k \le N-1$, and previous $A_{N,j}$ to new $A_{1,j}$.

An operation sequence is given as a string. You have to apply operations to a given matrix from left to right in a given string. Left, right, up, and down shifts are referred as 'L', 'R', 'U', and 'D' respectively in a string, and the following number indicates the row/column to be shifted. For example, "R25" means we should perform right shift with 25. In addition, the notion supports repetition of operation sequences. An operation sequence surrounded by a pair of parentheses must be repeated exactly m times, where m is the number following the close parenthesis. For example, "(L1R2)10" means we should repeat exactly 10 times the set of the two operations: left shift with 1 and right shift with 2 in this order.

Given operation sequences are guaranteed to follow the following BNF:

```
<sequence> := <sequence><rep> | <sequence><op> | <rep> | <op>
<rep> := '('<sequence>')'<number>
<op> := <shift><number>
<shift> := 'L' | 'R' | 'U' | 'D'
<number> := <nonzero_digit> |<number><digit>
<digit> := '0' | <nonzero_digit>
<nonzero_digit> := '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
```

Given N and an operation sequence as a string, make a program to compute the $N \times N$ matrix after operations indicated by the operation sequence.

Input

The first line of the input contains two integers N and L, where N $(1 \le N \le 100)$ is the size of the given matrix and L $(2 \le L \le 1,000)$ is the length of the following string. The second line contains a string S representing the given operation sequence. You can assume that S follows the above BNF. You can also assume numbers representing rows and columns are no less than 1 and no more than N, and the number of each repetition is no less than 1 and no more than 10^9 in the given string.

Output

Output the matrix after the operations in N lines, where the i-th line contains single-space separated N integers representing the i-th row of A after the operations.

Examples

standard input	standard output

`3 2`
`R1`
	`3 1 2`
	`4 5 6`
	`7 8 9`

`3 7`
`(U2)300`
	`1 2 3`
	`4 5 6`
	`7 8 9`

`3 7`
`(R1D1)3`
	`3 4 7`
	`1 5 6`
	`2 8 9`