E. Correct Bracket Sequence Editor

Recently Polycarp started to develop a text editor that works only with correct bracket sequences (abbreviated as CBS).

Note that a bracket sequence is correct if it is possible to get a correct mathematical expression by adding "+"-s and "1"-s to it. For example, sequences "`(())()`", "`()`" and "`(()(()))`" are correct, while "`)(`", "`(()`" and "`(())) (`" are not. Each bracket in CBS has a pair. For example, in "`(()(()))`":

- 1st bracket is paired with 8th,
- 2d bracket is paired with 3d,
- 3d bracket is paired with 2d,
- 4th bracket is paired with 7th,
- 5th bracket is paired with 6th,
- 6th bracket is paired with 5th,
- 7th bracket is paired with 4th,
- 8th bracket is paired with 1st.

Polycarp's editor currently supports only three operations during the use of CBS. The cursor in the editor takes the whole position of one of the brackets (not the position between the brackets!). There are three operations being supported:

- «`L`» — move the cursor one position to the left,
- «`R`» — move the cursor one position to the right,
- «`D`» — delete the bracket in which the cursor is located, delete the bracket it's paired to and all brackets between them (that is, delete a substring between the bracket in which the cursor is located and the one it's paired to).

After the operation "`D`" the cursor moves to the nearest bracket to the right (of course, among the non-deleted). If there is no such bracket (that is, the suffix of the CBS was deleted), then the cursor moves to the nearest bracket to the left (of course, among the non-deleted).

There are pictures illustrated several usages of operation "`D`" below.

All incorrect operations (shift cursor over the end of CBS, delete the whole CBS, etc.) are not supported by Polycarp's editor.

Polycarp is very proud of his development, can you implement the functionality of his editor?

Input

The first line contains three positive integers n, m and p ($2 \le n \le 500\,000$, $1 \le m \le 500\,000$, $1 \le p \le n$) — the number of brackets in the correct bracket sequence, the number of operations and the initial position of cursor. Positions in the sequence are numbered from left to right, starting from one. It is guaranteed that n is even.

It is followed by the string of n characters "`(`" and "`)`" forming the correct bracket sequence.

Then follow a string of m characters "`L`", "`R`" and "`D`" — a sequence of the operations. Operations are carried out one by one from the first to the last. It is guaranteed that the given operations never move the cursor outside the bracket sequence, as well as the fact that after all operations a bracket sequence will be non-empty.

Output

Print the correct bracket sequence, obtained as a result of applying all operations to the initial sequence.

Examples

input
8 4 5 (())()() RDLD
output
()

input
12 5 3 ((()())(())) RRDLD
output
(()(()))

input
8 8 8 (())()() LLLLLLDD
output
()()

Note

In the first sample the cursor is initially at position 5. Consider actions of the editor:

1. command "R" — the cursor moves to the position 6 on the right;
2. command "D" — the deletion of brackets from the position 5 to the position 6. After that CBS takes the form (())() , the cursor is at the position 5;
3. command "L" — the cursor moves to the position 4 on the left;
4. command "D" — the deletion of brackets from the position 1 to the position 4. After that CBS takes the form () , the cursor is at the position 1.

Thus, the answer is equal to () .