Problem B. Xoring Machine

Input file:	`xoring.in`
Output file:	`xoring.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider a sequence consisting of N positive integers: $x_1, x_2, \ldots, x_N$. You can perform the following operations with this sequence:

1. For each i from 2 to N in increasing order, set $x_i = x_i$ xor x_{i-1}. Denote this operation as "`L`".
2. For each i from N to 2 in decreasing order, set $x_i = x_i$ xor x_{i-1}. Denote this operation as "`R`".

You are given an initial sequence x_1, x_2, ..., x_N and a string of operations which consists of "`L`", "`R`" and repeat commands. A repeat command looks like "T`(...)`", where T is an integer ($1 \le T \le 1\,000\,000$) and the brackets contain an arbitrary non-empty string of operations. It means that you must apply the string in brackets T times.

Apply all the operations to the initial sequence and print the result.

Input

The first line of input contains an integer N ($1 \le N \le 30\,000$), the length of the initial sequence. The second line contains N integers x_i ($0 \le x_i \le 10^9$). The third line contains a string of operations formatted as described above. It is guaranteed that this string contains no more than 100 000 characters. Also it is guaranteed that the number of operations after expanding all "repeat" commands will be no more than 10^{18}.

Output

On the first line, print the resulting sequence x_1, x_2, ..., x_N after performing the given sequence of operations.

Examples

xoring.in	xoring.out
4 1 2 3 4 LLRL	1 2 2 6
5 8 2 1 4 16 3(L)2(R)LR4(L2(R))	8 10 11 15 23