ECOO '19 R1 P2 - L-Systems Go

Time limit: 30.0s   **Memory limit:** 512M

A Lindenmayer system (L-system) is a parallel rewriting system and a type of formal grammar. Some uses of L-systems include creating visual representations of vegetation, flowers, trees, and grasses.

An L-system consists of an axiom and a set of rules.

- The axiom (A) is a string that represents the starting state of the L-system.
- The set of rules defines what happens to each letter in an iteration of the system.

Given the axiom, set of rules, and the number of iterations for which to run the system, you are tasked with generating a two-letter code followed by the length of the final state. The two-letter code will be the concatenation of the first letter and last letter of the final state.

Input Specification

The input will contain 10 datasets. Each dataset begins with three terms R, T, A ($1 \leq R \leq 26, 1 \leq T \leq 30, 1 \leq |A| \leq 5$), the number of rules, the number of iterations, and the axiom. The next R lines each contain a character C followed by a string S ($1 \leq |S| \leq 3$) which represents a rule stating that each occurrence of C should be replaced with the string S. It is guaranteed that each letter in the system will have a corresponding rule.

For the first 4 cases, $T \leq 5$. For the first 7 cases, $T \leq 12$.

Output Specification

For each dataset, output a concatenation of first and last letter of the state and the length of the state after T iterations.

Sample Input (Two Datasets Shown)

```
3 5 AC
A CAB
B CB
C ACB
4 5 AD
A AC
B ACA
C BD
D B
```

Sample Output

```
CB 288
AB 60
```

Explanation of Sample Datasets

In the first dataset, the first iteration is AC > CABACB.

In the second dataset, the first two iterations are AD > ACB > ACBDACA.