H
Turing
`turing.{c,cpp,java}`

Some interesting documents have recently been found in the ACM archives. These documents contain an account of the original version of the ACM ICPC competition, which took place in Bletchley Park, England, in the year 1943.

In this original contest, programs submitted by contestants were executed in a simplified version of the so-called deterministic Turing Machine, which undoubtedly most of you are familiar with. In this simplified version, the tape is not infinitely long. Instead, tape cells are numbered 0 to $10^3 - 1$ (inclusive), from left to right. Also, the alphabet used is the unary alphabet, meaning that, prior to the execution of the program, the tape will be initialized with the string consisting of n ones followed by all zeroes, where n is the numerical value of the input, and after execution the tape contents should be m ones followed by all zeroes, where m is the numerical value of the corresponding output. In the beginning, the tape head is at position 0, and states are also numbered starting with 0, which is assumed to be the initial state. The machines work as usual: for every machine state q and every bit c (0 or 1), there is at most one rule determining what the next state will be if the symbol at the position of the tape head is c and the current state is q; the rule also specifies the symbol to write at the current position and in which direction the head should move. When no rule applies, the machine stops.

Your task will be to write a program that receives one of the Turing machines submitted by the contestants, as well as the test cases used (input and output pairs), and returns a verdict about the machine's results for each of the cases. Note that in this original contest, unlike the current one, there was a different verdict for each test case, instead of a general one.

Input Description

The input consists of a series of specifications (at most 30) of a Turing Machine and of the corresponding test cases.

The structure of each test case is the following:

The first line contains $1 \le N \le 1000$, the number of rules for the machine, and $1 \le M \le 100$, the number of test cases for the Turing Machine.

N lines follow, each one containing a rule, in the format q_{prev} c_{prev} q_{next} c_{next} mov. q_{prev} is the state of the machine before the rule is applied, c_{prev} is the content (either 0 or 1) of the tape at the current position p before the rule is applied, q_{next} is the state of the machine after the rule is applied, c_{next} is the content of position p after the rule is applied, and mov is the direction in which the tape head moves one step after applying the rule ("L" if it moves to the left and "R" if it moves to the right). The states should be integers between 0 and 1000, inclusive. The Turing machine should be deterministic; that is, there should be at most one transition from any pair (q_{prev}, c_{prev}), and should not be repeated in your output.

After this, M lines follow, each one containing two space-separated numbers, X and Y, $1 \le X, Y \le 1000$. X is the value of the input to the program, and Y is the value of the expected output.

The end of input is signaled by a case with $N = 0$ and $M = 0$.

Output Description

Your output should be *MLE* if the machine attempts to access any cell outside the tape range specified above; *TLE* if it runs for at least 10^4 iterations without stopping or causing an MLE error; *WA* if the machine stops but returns an incorrect result, and *AC* if the machine stops and returns the expected result. The output for each case should go in a different line.

Sample Input

```
6 3
0 1 1 0 R
0 0 0 0 R
1 1 1 1 R
1 0 2 1 L
2 1 2 1 L
2 0 900 1 R
1 3
300 301
0 1
0 0
```

Sample Output

```
WA
AC
MLE
```