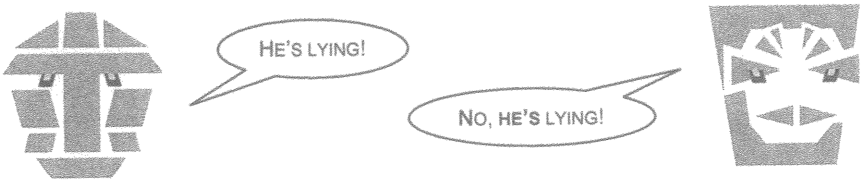
UCF Local Contest — September 5, 2009

Tautobots and Contradicticons

filename: `logotron`

The planet Logotron is inhabited by two types of robots – the Tautobots and the Contradicticons. The Tautobots are programmed to always tell the truth, while the Contradicticons must always lie. Unfortunately, there is no simple way for outsiders to tell them apart, which often causes problems.



The Problem:

You are given a set of statements made by a group of robots. Every robot knows the type of every other robot, as well as itself. Each statement consists of an author (the robot that made the statement), a subject (the robot the statement is about), and the type of the subject (Tautobot or Contradicticon). For example, "Robot 5 says that Robot 2 is a Tautobot" is a valid statement. Note that if Robot 5 is a Contradicticon, then Robot 2 must also be a Contradicticon, since Robot 5 lied.

Given M statements made by N robots, you must find the number of distinct ways to assign a type to each robot, consistent with the statements. Two assignments are considered to be different if at least one robot is a Tautobot in one and a Contradicticon in the other.

The Input:

The first input line contains a positive integer T, indicating the number of test cases to be processed. This will be followed by T test cases.

Each test case is formatted as follows. The first line consists of the numbers N and M ($1 \le N \le 15$, $0 \le M \le 100$). This is followed by M lines, each of which represents a statement by one of the robots. A statement is formatted as "A S X". Here A and S are integers between 1 and N (inclusive) representing the author of the statement and its subject respectively (assume that A and S will be different robots). X will be one of the characters 'T' (for Tautobot) or 'C' (for Contradicticon).

Assume that a robot will not contradict itself (making a statement and then making the opposite of that statement) but different robots may contradict each other (in that case, there is no possible answer, i.e., zero assignments). Also assume that we will not have the same statement repeated by a robot several times, i.e., no two input statements will be completely identical.

The Output:

For each test case, output a single line, formatted as: "Case `#t:`", where `t` is the test case number (starting from 1), a single space, and then the number of distinct assignments that can be made for that case. Follow the format illustrated in Sample Output.

Sample Input:

```
3
3 2
1 2 T
2 3 C
4 2
1 2 T
2 3 C
2 0
```

Sample Output:

```
Case #1: 2
Case #2: 4
Case #3: 4
```