UCF Local Contest — September 1, 2012

Dungeon Trouble!

filename: `dungeon`

Ash and Hazel used to play a game called "Dungeon Trouble!" when they were young. The mechanics of the game are quite simple. You start with a map of unnumbered dungeons in a cave, some of which are connected to one another via secret passage. The goal is to assign a number between 1 and 5, inclusive, to each dungeon so that no two adjacent dungeons share the same number. The last person to be able to successfully number a dungeon without violating the rule is declared to be the winner of that game.

Ash loves statistics and kept meticulous records of every game played. For each game played, he would write down a graph describing the connection of dungeons in the cave and the number that got assigned to each dungeon. Note that some dungeons may not have gotten numbered due to one player winning before they were all numbered.

Recently, Ash input all of these old games into his super awesome computer, "The Prime Analyzer 9000". Unfortunately, his computer was too specialized and only recognized the prime numbers he input for dungeon numbers; all non-prime dungeon numbers were lost! (For some strange reason, the computer only lost the non-prime dungeon number assignments. All other prime input was preserved as Ash originally input it.)

The Problem:

Ash wants your help to analyze the games to see what information can be gleaned from the data that is left. You still have a perfect mapping of the original dungeon and connections, and the dungeon numbers that were assigned 2, 3, or 5 (the primes). However, you do not know the dungeon numbers of any dungeon which originally had 1 or 4 assigned, nor do you know which dungeons were not originally assigned a number.

Ash wants you to analyze each game and tell him whether the game could possibly have been played to "completion" or not. A game is considered played to "completion" if all dungeons are assigned a number between 1 and 5. Note that a player may "win" a game before all dungeons are assigned a number but we consider a game is played to "completion" if all dungeons are assigned a number. Ash doesn't care who won; he just wants to know whether he and his sister were able to play the game to completion, assuming they both played optimally and each had the goal of completing the game. Note that, for the purpose of this program, the goal is not to win the game; rather to play the game to completion.

The Input:

The first input line contains a positive integer, *t*, indicating the number of dungeons to process. Each dungeon begins with a line containing two positive integers, *n* ($1 \le n \le 26$), which is the number of dungeons in the map, and *m* ($1 \le m \le 325$), which is the number of secret passageways connecting the dungeons. This will be followed by *m* lines, giving the connections between the dungeons. Each line will be of the form *i j* (‘A’ $\leq i < j \leq$ ‘Z’), indicating a secret passage (connection) between dungeon *i* and *j* (secret passages can be traversed in either direction). Following this will be a line containing an integer *k* ($0 \leq k \leq n$), giving how many dungeons already have numbers. This will be followed by *k* lines of the form *i c* (‘A’ $\leq i \leq$ ‘Z’ and c = {2,3 or 5}). It is guaranteed that each of these *k* lines will give a unique dungeon which exist in the given map, and the existing dungeon numbering will not violate any rules of the game. It is also guaranteed that there is a path between any two dungeons in the map, i.e., one can go from any dungeon to any other dungeon using the passages (connections) in the cave.

The Output:

For each dungeon in the input, output a line “`Dungeon #x: s`” where *x* is the dungeon number, starting with 1, and *s* is either the message “`Ash and Hazel played this game to completion`” or “`Ash and Hazel did not complete this game`” for the two possible outcomes. Leave a blank line after the output for each test case. Follow the format illustrated in Sample Output.

(Sample Input/Output on the next page)

Sample Input:

```
3
3 3
A B
A C
B C
1
A 3
3 3
A B
A C
B C
0
4 5
A B
A C
B C
B D
C D
2
A 5
D 2
```

Sample Output:

```
Dungeon #1: Ash and Hazel played this game to completion

Dungeon #2: Ash and Hazel did not complete this game

Dungeon #3: Ash and Hazel played this game to completion
```