I

Addition-Subtraction Game

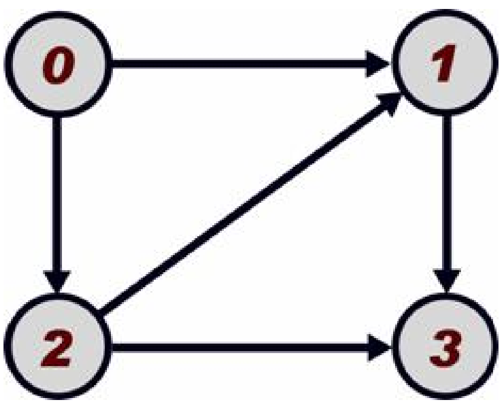
Input: Standard Input
Output: Standard Output

You and your friend are playing a **2** player game. The game is played in a graph of **V** vertices. The vertices are numbered from **0** to **V-1**. The graph has some directed edges. But the graph does not contain any cycles or loops. The rule of the game is as follows.

1. Initially vertex **i** has a positive value $\mathbf{value_i}$

2. Both players make their moves by turns. In his turn the player chooses a vertex with the following properties.
   - The value of the vertex is strictly positive.
   - The vertex has one or more outgoing edges.

   If there is no such vertex the player loses and the game terminates.



3. If the player can select a vertex the player will decrease the value of the selected vertex **i** by **1**. Then from the set of vertices which have an incoming edge from vertex **i,** the player will select $\mathbf{K_i}$ (this value will be given as input) vertices and increase the value of those vertices by **1**. Among these selected $\mathbf{K_i}$ vertices there can be duplicated vertices. And if a vertex is selected **n** times its value will be increased by **1** every time. Or in another word its value will be increased by **n**. For example if the $\mathbf{K_i=6}$ and the selected vertex set is **{2,2,2,3,3,5}** then $\mathbf{value_2}$ will be increased by **3**, $\mathbf{value_3}$ will be increased by **2** and $\mathbf{value_5}$ will be increased by **1**.

Now consider the graph on the right.

Let the values of **K** be **{2,1,3,2}**.

Now the value set **{0,0,0,5}** is a losing terminating position because the player cannot select any vertex which have outgoing edges and positive values.
For the value set **{3,4,5,6}** the current player can go to the following value states by **1** move.

- **{2,5,6,6}** – select the vertex **0**, decrease its value by **1**. And increase both of **1** and **2** by **1**. Here $\mathbf{K_0=2}$.

- **{2,6,5,6}** – select the vertex **0**, decrease its value by **1** and increase its adjacent **1** by **2**. Here $\mathbf{K_0=2}$.
- **{2,4,7,6}** – select the vertex **0**, decrease its value by **1** and increase its adjacent **2** by **2**. Here $\mathbf{K_0=2}$.
- **{3,3,5,7}** – select the vertex **1**, decrease its value by **1** and increase its adjacent **3** by **1**. Here $\mathbf{K_1=1}$.
- **{3,7,4,6}** – select the vertex **2**, decrease its value by **1** and increase its adjacent **1** by **3**. Here $\mathbf{K_2=3}$.
- **{3,5,4,8}** – select the vertex **2**, decrease its value by **1** and increase its adjacent **1** by **1** and **3** by **2**. Here $\mathbf{K_2=3}$.
- **{3,6,4,7}** – select the vertex **2**, decrease its value by **1** and increase its adjacent **1** by **2** and **3** by **1**. Here $\mathbf{K_2=3}$.
- **{3,4,4,9}** – select the vertex 2, decrease its value by 1 and increase its adjacent 3 by 3. Here $\mathbf{K_2=3}$.

Now given the graph and initial values of each of the vertices your task is to determine if the first player wins or loses given that both players play perfectly.

Input

Input contains multiple number of test cases. First line contains **T(1 ≤ T ≤ 20)** the number of test cases. Each test case starts with a line **V(2 ≤ V ≤ 100)** and **E(2 ≤ E ≤ 1500)**. **V** is the number of vertices and **E** is the number of edges. Each of the next **E** lines contains **2** integers **FROM(0 ≤ FROM < V)** and **TO(0 ≤ TO < V)** denoting that there is a directed edge from **FROM** to **TO**. **FROM** and **TO** will not be equal. **Also each vertex will have at most 15 outgoing edges**. Next line contains **V** integers K_0, K_1,... K_{V-1}. Each of the value of **K** is between **1** and **100** inclusive. Next line contains **R(1 ≤ R ≤ 100)** the number of rounds. There will be **R** round of game with this graph. Each of the next **R** lines contains the description of each round. Each round consists of **V** integers $Value_0$ $Value_1$ ...$Value_{V-1}$ denoting the initial value of each vertex. Each of these $Value_i$ will be between **1** and **100** inclusive.

Output

For each test case output consist of **R+1** lines. First line is **"Game#i:"** where **i** is the game number. Game number starts from **1**. Each of the next **R** lines contains **"Round#j: RESULT"** where **j** is the number of round. **RESULT** is either **WINNING** when the initial values of this round is a winning position for the first player or **LOSING** when the initial values of this round is a losing position for the first player. We will assume that both players play perfectly. Print a blank line after the output of each test case. See the output for sample input for more clarification.

Sample Input

```
2
3 3
1 0
2 0
1 2
0 2 2
5
3 0 0
4 1 0
5 0 1
1 1 1
2 2 2
4 3
0 1
1 2
2 3
3 2 1 0
5
0 0 0 0
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
```

Output for Sample Input

```
Game#1:
Round#1: LOSING
Round#2: WINNING
Round#3: WINNING
Round#4: WINNING
Round#5: LOSING

Game#2:
Round#1: LOSING
Round#2: LOSING
Round#3: WINNING
Round#4: WINNING
Round#5: LOSING
```

Problem setter: Abdullah al Mahmud, Special Thanks: Rujia Liu