# Problem 2 – Flea Racing

One relatively lazy day at the office, your boss decided to bring his pet dog with him to work. Unfortunately it had been infested with fleas that started to jump all over the office. Our creative programmers decided to catch them and race them for entertainment. Each flea would be examined carefully and its **jumping distance** calculated. After that a **name** would be assigned to it. Programmers would **pick the length of the track** and align all fleas on the start line. They would also choose the **maximum allowed number of jumps**. Alternating, each flea would jump the distance it can until the maximum jumps have passed for each flea. If no flea has passed the finish line until this moment the winner will be deemed the one that has gotten the **furthest**. If multiple fleas have gotten to the same position, the winner will be the flea (amongst those that have jumped furthest) that has **jumped last** (There must be only 1 winner, right?). To compensate for that order of jumping problem programmers decided that if at any point during the race a flea jumps **exactly on the finish line or behind** **it**, it would be deemed winner and all other fleas denied the chance to jump any more (even if the next flea would have jumped further behind the finish line). Because you are the smartest and most ambitious programmer in your team, you are assigned the task of creating the system for simulating the races. The system should print the **final state of the track** which holds the **current position of each flea**. If any flea has jumped behind the finished line and is deemed winner it should be displayed at the **last possible position of the track**. Each flea is represented on the track by the capitalized version of the first letter in its name. Don’t forget these are **programmer fleas**, so their starting position is 0 not 1. Progress is marked in red in the first example.

### Input

The input will be passed to the first JavaScript function found in your code as **a single** **array,** containing:

* the **number of jumps allowed**,
* **the length of the track**,
* **strings** in the format **'name, jumpDistance'** containing the **data for each flea**.

### Output

The output consists of several lines. On the first 2 lines print “#” symbols equal to the length of the track. After those lines (called the audience) print the **final** state of the track. Print another set of audience (2 lines, “#” symbol, length of the track), followed by the winner. Check the examples for correct formatting.

### Constraints

* The **length of the track and flea jump distance capability** will be integers in the range **[1…100]**.
* The array of fleas will contain each **flea** as an individual **array of 2 elements** – **name** and **jump distance**.
* The **amount of participating fleas and number of jumps allowed** will be in range **[1…10]**
* For cosmetic purposes the name of each flea will begin with a different letter.
* Time limit: 0.3 sec. Memory limit: 16 MB.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 10  19  angel, 9  Boris, 10  Georgi, 3  Dimitar, 7 | ###################  ###################  ..................A  ..........B........  ...G...............  .......D...........  ###################  ###################  Winner: angel |  | 3  5  cura, 1  Pepi, 1  UlTraFlea, 1  BOIKO, 1 | #####  #####  ...C.  ...P.  ...U.  ...B.  #####  #####  Winner: BOIKO |