**Substitute Algorithm**

In data analysis, the substitute algorithm determines the single final value to use for each data parameter. The substitute algorithm works in the following way:

* Data is acquired from multiple sources in order from least to most preferred, i.e., if a parameter Pi is present in both source 1 and source 2, the parameter from the higher priority source, source 2, is used in the final parameter list, and any value from an earlier source is superseded.
* As new parameters arrive, they are added to the list
* If a parameter Pi is present only in one of the sources, it is directly added to the final parameter list.

Hence,

* The result of performing the above operations until all the parameters from source 1 and source 2 are exhausted is the result of SubstituteAlgorithm (source 1, source 2).
* Each time a new value for a parameter is encountered from a higher preferred site, the old data is superseded.
* Assuming three sources S1, S2, S3:

SubstituteAlgorithm (S1, S2, S3) = SubstituteAlgorithm (SubstituteAlgorithm (S1, S2), S3).

Given a list of sources S1, S2....Sn, find the final parameter list given by SubstituteAlgorithm (S1, S2,.....Sn). Produce your results in the ascending order of the keys.

For example, a rating parameter of buy, sell or hold from three sources in increasing order of preference: [buy, sell, hold], where buy is from S1, immediately superseded by sell S2, immediately superseded by hold S3. The final rating is the only one that hasn't been superseded, so you use 'hold’ as the final rating.

**Requirement**

* Create a class SubstituteAlgorithm with a constructor which accepts the number of sources, the number of parameters of each source and the array of source strings presented from lowest to highest preference as arguments.
* The input arguments received thru the constructor should be assigned to *self*.
* Create a function *compute\_parameter\_value* within the class. The function must return an array of strings that denotes the final parameter list values with the keys in their ascending order.
* Write the code to read the input as string, create the object and call the function to get the output.

**Input (Pass as string with line separators):**

* The first line contains a positive integer n, the number of sources.
* The next line contains a positive integer, p, denoting the number of parameters of each source.
* Each of the next n lines contains an array of p space-separated strings of format key:value, denoting the key and value of source[i] parameters.

**Constraints:**

* 1<=n<100
* 1<=p<1000

**Sample Input**

2

3

P1:a P3:b P5:x

P1:b P2:q P5:x

**Sample Output**

P1:b P2:q P3:b P5:x

**Explanation**

The value latest source for each of the Pi is considered for the final parameter list.

P1 (source 2) = b

P2 (source 2) = q

P3 (source 1) = b

P5 (source 2) = x