The Order of Function Inlining
Time limit: 1s

Lim's project manager, Functa, has just attended Software Engineering conference where one of the topic is about source code optimization. At that time, the speaker said that it is important not to encapsulate everything in a function for reliable speed. As a naive manager, Functa is really fascinated with that concept and wants to apply it directly to his current project. He wants to remove all function invocations from his project. Lim, as his underling, has no other option but to accept such request for the sake of his salary. In order to provide a smooth conversion from current project to no-function-involved project, Lim should define the order of function inlining. According to Functa's request, the order should be done in alphabetical order where all functions which are invoked on other functions should be inlined first. Please help Lim to provide such order!

Input:
At first, it accepts N which represents the number of functions. It is then followed by the data of each function where each of them consists of function signature, the number of invoked function on that function, and list of invoked functions, represented as function signature. It is important to note that N will always be lower than 1000 and there will be no recursive function involved.
Output:
A sequence representing the order of function inlining

Sample input:
```
5
a()
2 b() c()
b()
1 c()
d()
0
e()
0
c()
0
```

Sample output:
```
c() -> b() -> a() -> d() -> e()
```

Notes for sample output:
c() is inlined first before b() and a() since it is invoked on these functions.
b() is inlined before a() and after c() since b() is invoked in a() and invokes c().
d() is inlined first before e() since the sequence is displayed based on alphabetical order.