Stand In a Queue

Saarang team is out for distributing the pre-registered t-shirts and has set up a counter near Himalaya lawns. However due to lack of proper management from volunteers, the team is having a tough time managing the queue and needs your help !

Instead of having a single queue, students have now arranged themselves in multiple sub- queues with each sub-queue merging into the main queue at some point. Queue structure can be visualised as multiple lanes merging into the main lane. Your task is to arrange these students in a single main queue subject to following constraints. Smallest sub-queue will first move to the end of main queue then the next smallest, and so on until we have a single queue. In the end print this single queue. Students will be represented by numbers(their ids) and list of students in main and sub-queues will be input to the program.

Note : Use stack data structure to represent main and sub-queues; and finally place all the students in queue data structure. Implement and use size() and print() functions for stack and queue respectively.

First line of input N indicates number of queues. Next N lines will have ids of students in respective queues and first list indicates the main queue. As you can see queue splits at student with id 2 and with id 4. Output is based on the fact that third queue has only one student (with id 10) and will be appended to main queue (1 2 3 4 5 6 10) then next sub-queue has size of 3 (with student ids 7 8 and 9) and will now be appended to main queue (1 2 3 4 5 6 10 7 8 9).

Input Format

3
1 2 3 4 5 6
1 2 **7 8 9**
1 2 3 4 **10**

Constraints

1 <= N <= 50 total students in queue <= 100

Output Format

1 2 3 4 5 6 10 7 8 9