C++ Fundamentals – Regular Exam – 10 February 2024

Please submit your source code to all below-described problem in Judge.

1. Government Facility

You're visiting one of the most weird government institutions. Thankfully, you're just a visitor, but while you wait for your friend to finish their business there, you watch some weird queues that form up on this place.

You see two normal queues (let's call them queue 1 and 2) in front of two counters. You also see a row of chairs (let's call it 3). The row of chairs 3 is very strange: people, who sit there last, are first called! This seems very unfair to you, but that's how this administration works.

You notice that the most lucky people just go on one queue, wait for their turn, and then get their job done on the counter and leave the facility.

You also notice that some unlucky people, after they're called on the counter, they are sent from the counter to either the other queue, or to the row of chairs 3 to wait there to be called.

You decide to create a program to track the status of these three lines.

The commands can be:

- A **name** (without any spaces), followed by a **number** (1 or 2). This means that a new person enters the facility and goes either to queue 1 or 2.
- A **number**, as follows:
  - **10** means a person from queue 1 is served and happily leaves the facility, so they're not present anymore in any of the three structures.
  - **20** means a person from queue 2 is served and happily leaves the facility, so they're not present anymore in any of the three structures.
  - **12** means a person from queue 1 is served, but then they’re sent to queue 2 to wait there.
  - **23** means a person from queue 2 is sent to the row of chairs 3, to wait to be called from there.
  - **31** means a person from the row of chairs 3 is called to wait to queue 1.
  - **99** means you must print the three structures in the order how people will get called from them:
    - You print queue 1 as "`1: `" and then all names, separated with a single space, and then a new line.
    - You print queue 2 as "`2: `" and then all names, separated with a single space, and then a new line.
    - You print the row of chairs as "`3: `" and then all names, separated with a single space, and then a new line.
    - **Note:** If any of the structures above is empty, print “<empty>” (see example below)
  - **5** means "end", you're done, your program finishes.
- There are no other commands to consider!

Example 1

Input	Output	Explanation	Lines' contents (after command)

`Ani 1`		**Ani** enters the facility and goes to 1.	`1: Ani`
			`2:`

			3:
Spas 2		**Spas** enters the facility and goes to 2.	1: Ani 2: Spas 3:
23		**Spas** is sent from 2 to 3.	1: Ani 2: 3: Spas
Teo 1 Mia 2		**Teo** enters the facility and goes to 1. **Mia** enters the facility and goes to 2.	1: Ani Teo 2: Mia 3: Spas
23		A person is sent from 2 to 3. That's Mia, as she last went to 3.	1: Ani Teo 2: 3: Mia Spas
Ivan 2		**Ivan** enters the facility and goes to 2.	1: Ani Teo 2: Ivan 3: Mia Spas
99	1: Ani Teo 2: Ivan 3: Mia Spas	At 99 we print the contents of the three structures.	1: Ani Teo 2: Ivan 3: Mia Spas
31 99	1: Ani Teo Mia 2: Ivan 3: Spas	A person is called 3 to 1. Then we print again the contents.	1: Ani Teo Mia 2: Ivan 3: Spas
10 20 99	1: Teo Mia 2: <empty> 3: Spas	A person from 1 gets served and leaves. A person from 2 gets served and leaves. Then we print again the contents.	1: Teo Mia 2: 3: Spas
5		The program ends.	

Example 2

Input	Output	Lines' contents (after command)
Spas 2		1: 2: Spas 3:
Georgi 1		1: Georgi 2: Spas 3:
23 99	1: Georgi 2: <empty> 3: Spas	1: Georgi 2: <empty> 3: Spas
Kunka 2 Stanka 1		1: Georgi Stanka 2: Kunka 3: Spas
99	1: Georgi Stanka	1: Georgi Stanka

	2: Kunka 3: Spas	2: Kunka 3: Spas
Chuck 2		1: Georgi Stanka 2: Kunka Chuck 3: Spas
Norris 2		1: Georgi Stanka 2: Kunka Chuck Norris 3: Spas
23 23		1: Georgi Stanka 2: Norris 3: Chuck Kunka Spas
99	1: Georgi Stanka 2: Norris 3: Chuck Kunka Spas	1: Georgi Stanka 2: Norris 3: Chuck Kunka Spas
20 10 10		1: <empty> 2: <empty> 3: Chuck Kunka Spas
99	1: <empty> 2: <empty> 3: Chuck Kunka Spas	1: <empty> 2: <empty> 3: Chuck Kunka Spas
31 31		1: Chuck Kunka 2: <empty> 3: Spas
99 5	1: Chuck Kunka 2: <empty> 3: Spas	