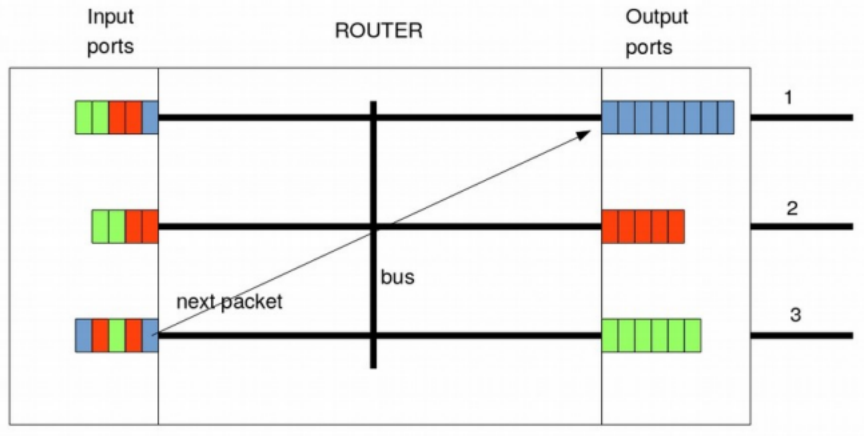
159.201 Algorithms & Data Structures
Assignment 3

A router reads packages from input ports and writes them to output ports based on their address. Input and output ports are numbered $0 \ldots 255$. During each cycle, an arbitrary number of packages arrives at the input queues. The router then tries to read one package from each input port queue, in order (starting with port 0), and if available writes it to the matching output port queue. For clarity, one cycle consists of (a) reading any number of packets into input queues, and (b) sending one packet per input queue to the specified output queue.

Processing proceeds cycle by cycle, until no more inputs arrive and all input queues are empty (so that all packets have been sent to output queues).



Write a C++ program that simulates such a router, reading from standard input and writing to standard output. The input consists of one line per input cycle, in the format

$$n \text{ port}_1 \text{ addr}_1 \text{ value}_1 \text{ ... port}_n \text{ addr}_n \text{ value}_n$$

Here n denoted the number of packages arriving in this cycle, *port* describes the input port where the package arrives, *addr* is the output port the package is destined for, and *value* is the payload. Payloads are single word strings, all other inputs are integers.

Your program should print one line per output port that has packages sent to it. Each line should start with the port number, then list all payloads in the order they arrived in the output queue, separated by a single space character. For example:

Input	Output
2 2 3 easy 2 1 was 2 1 1 that 2 3 ...	1 that was 3 easy ...

You can use CodeRunner to test your work, but must submit your assignment as usual.

Detailed example how the input above is processed:

- Cycle 1 (load):
  - read first line `2 2 3 easy 2 1 was`
  - parse as 2 packets (2,3,easy) and (2,1,was)
  - insert packets into the input queues:

Input Queues (port: packets)	Output Queues (port: packets)
2: (3,easy) (1,was)	

- Cycle 1 (process): transfer the first packet from each input queue

Input Queues (port: packets)	Output Queues (port: packets)
2: (1,was)	3: easy

- Cycle 2 (load):
  - read second line `2 1 1 that 2 3 ...`
  - parse as 2 packets (1,1,that) and (2,3,...)
  - insert packets into the input queues:

Input Queues (port: packets)	Output Queues (port: packets)
1: (1,that) 2: (1,was) (3,...)	3: easy

- Cycle 2 (process): transfer the first packet from each input queue

Input Queues (port: packets)	Output Queues (port: packets)
2: (3,...)	1: that was 3: easy

Note: queues are processed 0..255, so (1,that) is transfered before (1,was).

- Cycle 3 (load): no input pending, but input queues are not empty yet
- Cycle 3 (process): transfer the first packet from each input queue

Input Queues (port: packets)	Output Queues (port: packets)
	1: that was 3: easy ...

- Cycle 4 (load): no input pending, and input queues are empty