Challenge

Load Balancing is quite important in Cloud environments. We are always trying to minimize the costs so we keep the number of servers as low as possible. On the other hand we know that capacity and performance improves when we add more servers. The challenge is to keep the servers as busy as possible under a certain load capacity.

On our simulation environment, at each clock tick (time unit), users connect to available servers and request the same task to be executed. Each task consumes `Ttask` ticks of time to get accomplished and after that the user disconnects immediately.

The servers are virtual machines that can be spin up immediately to accommodate new users. Each servers costs $1.00 per clock tick and can handle with `Umax` number of simultaneous users. You can terminate the empty servers.

Your challenge is to build a program in Python that handles with the incoming users and assign them to servers keeping the cost as low as possible.

Input: A file where each line contains the number of new users for that tick.
Output: A file where each line contains a list of the available servers at the end of that tick, represented by the number of users on each server.

Example with `Ttask = 4` and `Umax = 2`

clock ticks	Input	Output	Tip
1	1	1	1 server for 1 user. (1 server created)
2	3	2,2	2 servers for 4 users. (1 server created)
3	0	2,2	idem
4	1	2,2,1	3 servers for 5 users. (1 server created)
5	0	1,2,1	3 servers for 4 users. (First user left after 4 ticks)
6	1	2	1 server for 2 users (3 users left, 1 joined. 2 servers terminated)
7		2	idem
8		1	1 server for 1 user (1 user left)
9		1	idem
10			last user left. (last server terminated)
		$15	Total Cost: $1 x 5 ticks (first VM) + $1 x 4 ticks (second VM) + $1 x 6 ticks (third VM) = $15

Test with the attached input.txt file and submit your code and output file.
Use `Ttask = 5` and `Umax = 10.`

Thanks,
Caio