Arcade Management

Objective
Give practice with linked lists and stacks.

Story
You opened a retro arcade using the capital obtained from your handle generator venture. It's pretty popular among the denizens of Gameland. Most people in Gameland have never seen an arade before woah. You have decided to list the most valuable player (in this case the person in your store that entered with the most amount of money). This is totally not going to backfire or be a huge violation of trust.

The problem you are having is that your system only records two events: 1) the name and the amount of money held by the player entering or 2) if someone is leaving (the system does not say who). Luckily, people like your arcade so much that they refuse to leave before everyone that arrived AFTER them has already left. It makes it very easy to determine who exits when someone leaves.

Problem
Given a list of people entering and leaving determine at certain (specified) moments the most valuable player.

Input
Each line of input will start with a non-negative integer, t ($0 \leq t \leq 3$), representing the type of line to process. If t is 0, then the day is over and the program should terminate. If t is 1, then the program should read in the following player information that is entering the arcade. The player's information is composed of two values separated by a space, an integer, M, and string, S ($1 \leq M \leq 10^9$, $1 \leq |S| \leq 20$), representing the amount of money in tokens and the name of the player entering. If t is 2, then the program should process a player leaving. No further information is provided when t is 2. If t is 3, then your program should output the most valuable player in the arcade.

Output
Output should contain 1 line for each 3 entered. Each line will contain the name of the most valuable player at the time of the request. If two players entered with the same amount of money, print the most recent one (they have probably spent less money than the other one).

Sample Input	Sample Output
1 5 Eric 1 10 John 1 2 Kate 3 1 30 Ash 3 2 2 3 2 3 0	John Ash John Eric
1 3 Josh 1 4 Grace 3 1 5 isASoreLoser 3 1 6 RealFunnyJosh 3 3 2 2 2 3 2 0	Grace isASoreLoser RealFunnyJosh RealFunnyJosh Josh

Explanation

Case 1

Time	Event	People in the arcade
1	Eric enters with 5 tokens	Eric
2	John enters with 10 tokens	Eric, John
3	Kate enters with 2 tokens	Eric, John, Kate
4	Print the MVP (John)	Eric, John, Kate
5	Ash enters with 30 tokens	Eric, John, Kate, Ash
6	Print the MVP (Ash)	Eric, John, Kate, Ash
7	Someone leaves (Ash)	Eric, John, Kate
8	Someone leaves (Kate)	Eric, John
9	Print the MVP (John)	Eric, John
10	Someone leaves (John)	Eric
11	Print the MVP (Eric; by default)	Eric
12	The store closes Someone should check on Eric	Eric

Case 2

Time	Event	People in the arcade
1	Josh enters with 3 tokens	Josh,
2	Grace enters with 4 tokens	Josh, Grace
3	Print the MVP (Grace)	Josh, Grace
4	isASoreLoser enters with 5 tokens (o_o)	Josh, Grace, isASoreLoser
5	Print the MVP (isASoreLoser)	Josh, Grace, isASoreLoser

6	RealFunnyJosh enters with 6 token (>_<)	Josh, Grace, isASoreLoser, RealFunnyJosh
7	Print the MVP (RealFunnyJosh)	Josh, Grace, isASoreLoser, RealFunnyJosh
8	Print the MVP (RealFunnyJosh)	Josh, Grace, isASoreLoser, RealFunnyJosh
9	Someone leaves (RealFunnyJosh)	Josh, Grace, isASoreLoser
10	Someone leaves (isASoreLoser)	Josh, Grace
11	Someone leaves (Grace)	Josh
12	Print the MVP (Josh)	Josh
13	Someone leaves (Josh)	---
14	The day is over (thankfully)	---

Hints

Observation 0: The entering and leaving of customers is that of a stack. A person can only leave if everyone that arrived after that is gone. It's like the other customers are blocking the exit. It's probably a fire hazard, but who knows the rules of Gameland (probably your annoying friend that *has* to read the rules prior to playing other games).

Observation 1: If someone enters and has less money than the current MVP, then their name will never be printed. If someone enters and has greater or equal money compared to that of the current MVP, then they will be the MVP until they leave or until someone with more money enters.

Max Stack: We can keep track of the MVP using a second stack. When we push on a person to the arcade stack, we push on to the MVP stack whoever the MVP for the current people in the arcade is. When we pull someone out of the arcade we can pop from both stacks.

Only Max Stack: We really only need one stack.

Grading Criteria

- No extra input output (e.g. input prompts, “Please enter the number of friends:”)
    - 10 points
- Good comments, whitespace, and variable names
    - 15 points
- Read until the ending type is given using a loop
    - 5 points
- Store the MVPs in a stack for each person enter
    - 10 points
- Pop the MVP stack when someone leaves
    - 10 points
- Programs will be tested on 10 cases
    - 5 points each

No points will be awarded to programs that do not compile using “gcc -std=gnu11 -lm”.

Sometime a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use a Linked list and stack. ***Without this programs will earn at most 50 points!***

Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.

No partial credit will be awarded for an incorrect case.