Java Priority Queue

In computer science, a priority queue is an abstract data type which is like a regular queue, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. - Wikipedia

---

In this problem we will test your knowledge on Java Priority Queue.

You have to deal with 2 types of events: *ENTER* (a student enters the queue) or *SERVED*.

A unique token is assigned to any student entering the queue. The queue serves the students based on the following criteria:

1. The student having the highest *Cumulative Grade Point Average* (CGPA) is served first.
2. Any students having the *same CGPA* will be served by name in ascending case-sensitive alphabetical order.
3. Any students having the *same CGPA and name* will be served in ascending token order.

Given a sequence of n events, print the names of students who are yet to be served(based on above criteria). If the queue is empty, print `EMPTY`.

Input Format

The first line contains an integer, n, denoting the total number of events. Each of the n subsequent lines will be of the following two forms:

1. **ENTER name CGPA token** - The student to be inserted into the priority queue.
2. **SERVED** - The highest priority student in the queue was served.

Constraints

Constraints

- $2 \leq n \leq 1000$
- $0 \leq CGPA \leq 4.00$ where $CGPA \in \mathbb{R}$
- $1 \leq token_i \leq 10^5$ where each token i is a unique integer.
- $2 \leq |name| \leq 30$

Output Format

Print the names (based on the criteria) of the students who are not served at all after executing all n events; if every student in the queue was served, then print `EMPTY`.

Sample Input

```
12
ENTER John 3.75 50
ENTER Mark 3.8 24
ENTER Shafaet 3.7 35
SERVED
SERVED
ENTER Samiha 3.85 36
SERVED
```

```
ENTER Ashley 3.9 42
ENTER Maria 3.6 46
ENTER Anik 3.95 49
ENTER Dan 3.95 50
SERVED
```

Sample Output

```
Dan
Ashley
Shafaet
Maria
```

Explanation

Let's call our queue Q.

n_0: We add John to the empty queue.
$Q_0 = \{(\text{John}, 3.75, 50)\}$

n_1: We add Mark to the queue; $Q_1 = \{(\text{John}, 3.75, 50), (\text{Mark}, 3.8, 24)\}$

n_2: We add Shafaet to the queue; $Q_2 = \{(\text{John}, 3.75, 50), (\text{Mark}, 3.8, 24), (\text{Shafaet}, 3.7, 35)\}$

n_3: Mark is served as he has the highest CGPA; $P_3 = \{(\text{John}, 3.75, 50), (\text{Shafaet}, 3.7, 35)\}$

n_4: John is served next as he has the highest CGPA; $P_4 = \{(\text{Shafaet}, 3.7, 35)\}$

n_5: We add Samiha to the queue;
$Q_2 = \{(\text{Shafaet}, 3.7, 35), (\text{Samiha}, 3.85, 36)\}$

n_6: Samiha is served as she has the highest CGPA; $P_6 = \{(\text{Shafaet}, 3.7, 35)\}$

n_7 through n_{10}, the next four students are added giving us:
$Q_{10} = \{(\text{Shafaet}, 3.7, 35), (\text{Ashley}, 3.9, 42), (\text{Maria}, 3.6, 46), (\text{Anik}, 3.95, 49), (\text{Dan}, 3.95, 50)\}$

n_{11}: Anik is served because though both Anil and Dan have the highest CGPA but Anik comes first when sorted in alphabetic order;
$P_{11} = \{(\text{Dan}, 3.95, 50), (\text{Ashley}, 3.9, 42), (\text{Shafaet}, 3.7, 35), (\text{Maria}, 3.6, 46)\}$

As all events are completed, we print names of each remaining students on a new line.