In an election, the i-th vote was cast for persons[i] at time times[i].

Now, we would like to implement the following query function: TopVotedCandidate.q(int t) will return the number of the person that was leading the election at time t.

Votes cast at time t will count towards our query.  In the case of a tie, the most recent vote (among tied candidates) wins.

**Example 1:**

**Input:** ["TopVotedCandidate","q","q","q","q","q","q"], [[[0,1,1,0,0,1,0],[0,5,10,15,20,25,30]],[3],[12],[25],[15],[24],[8]]

**Output:** [null,0,1,1,0,0,1]

**Explanation:**

At time 3, the votes are [0], and 0 is leading.

At time 12, the votes are [0,1,1], and 1 is leading.

At time 25, the votes are [0,1,1,0,0,1], and 1 is leading (as ties go to the most recent vote.)

This continues for 3 more queries at time 15, 24, and 8.

**Note:**

1. 1 <= persons.length = times.length <= 5000
2. 0 <= persons[i] <= persons.length
3. times is a strictly increasing array with all elements in [0, 10^9].
4. TopVotedCandidate.q is called at most 10000 times per test case.
5. TopVotedCandidate.q(int t) is always called with t >= times[0].