



That is Your Queue

Link submit:

https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=3359

Solution:

C++	http://ideone.com/wVHtP0
Java	https://ideone.com/RthR6N
Python	https://ideone.com/gKBnvu

Tóm tắt đề:

Chính phủ đã giải quyết vấn đề chăm sóc sức khỏe cho toàn thể người dân. Mỗi người dân được gán một con số phân biệt (từ 1 đến P) với P là tổng dân số của cả nước.

Mọi người sẽ được xếp vào hàng đợi, ai số thứ tự nhỏ sẽ được khám trước, ai lớn hơn sẽ khám sau. Tuy nhiên, vẫn có trường hợp ngoại lệ, khi có một ca khám bệnh khẩn cấp sẽ được ưu tiên lên khám trước.

Cho bạn một chuỗi các truy vấn hãy xác định thứ tự khám bệnh của các công dân.

Input:

Có nhiều bộ test, nhiều nhất 10 bộ. Mỗi bộ test gồm các thông tin sau:

Dòng đầu tiên chứa hai số P C ($1 \leq P \leq 1.000.000.000$, $1 \leq C \leq 1000$) lần lượt là số lượng dân số và số lượng truy vấn.

C dòng tiếp theo mỗi dòng sẽ chứa một trong hai định dạng sau:

- N: công dân trên đầu hàng đợi vào khám bệnh. Và sau đó đưa về cuối danh sách.
- E x: công dân thứ x được ưu tiên vào phía trước hàng đợi.

Kết thúc dữ liệu là hai số 0 0.

Output:

Với mỗi bộ test in theo định dạng sau:

- Dòng đầu in "Case x:" với x là số thứ tự của bộ test bắt đầu từ 1.
- Các dòng tiếp theo ứng với mỗi truy vấn N in ra số của người được khám bệnh.

Ví dụ:

3 6	Case 1:
N	1
N	2
E 1	1
N	3
N	2
N	Case 2:
10 2	1
N	2
N	
0 0	

Giải thích ví dụ:

Với bộ test 1, hàng đợi ban đầu: 1, 2, 3.

N → xuất số 1. Hàng đợi lúc này 2, 3, 1.

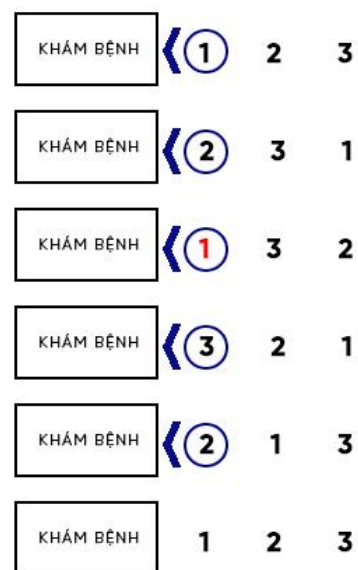
N → xuất số 2. Hàng đợi lúc này 3, 1, 2.

E 1 → Đem số 1 lên đầu lúc này hàng đợi 1, 3, 2.

N → Xuất ra số 1. Hàng đợi 3, 2, 1.

N → Xuất ra số 3. Hàng đợi 2, 1, 3.

N → Xuất ra số 2. Hàng đợi 1, 3, 2.



Hướng dẫn giải:

Ý tưởng đơn giản ban đầu bạn sẽ nghĩ ra là bạn sẽ tạo một queue chứa P phần tử. Sau đó mỗi lần gặp một truy vấn N thì bạn in phần tử đầu queue ra rồi bỏ nó lại vào queue (bỏ lại vào queue nghĩa là lúc này nó đã được đem về cuối queue).

Nếu gặp truy vấn E x thì bạn sẽ tìm tới vị trí của x rồi đem nó lên đầu queue.

Cách giải trên là phù hợp, tuy nhiên bạn cần lưu ý là số lượng P rất lớn 10^9 như vậy nếu bạn cứ thao tác với ý tưởng trên chắc chắn sẽ bị quá thời gian TLE. Vì vậy, bạn cần một ý tưởng khác.

Bạn sẽ khởi tạo queue ban đầu tùy thuộc vào số P hoặc C (số nào nhỏ hơn thì tạo queue theo số đó). Bạn tạo như vậy vì dù queue bạn có lớn tới đâu đi nữa thì nó cũng chỉ tối đa C thao tác vì thế việc tạo P phần tử ban đầu là không cần thiết.

Sau khi tạo số phần tử dựa vào P hoặc C ra thì bạn sẽ làm bình thường, theo ví dụ này bạn sẽ làm như sau:

2000 6	Case 1:
N	1
N	2
E 1500	1500
N	3
N	4
N	
0 0	

Bạn sẽ tạo queue gồm 6 phần tử: 1 2 3 4 5 6.

N → Xuất số 1. Hàng đợi lúc này 2 3 4 5 6 1

N → Xuất số 2. Hàng đợi lúc này 3 4 5 6 1 2

E 1500 → tạo một con số 1500 và đưa nó lên đầu hàng đợi 1500 3 4 5 6 1 2

N → Xuất số 1500 ra. Hàng đợi lúc này 3 4 5 6 1 2 1500

N → Xuất số 3 ra. Hàng đợi lúc này 4 5 6 1 2 1500 3

N → Xuất số 4 ra. Hàng đợi lúc này 5 6 1 2 1500 3 4

Độ phức tạp: $O(T * C * \min(C, P))$ với T là số lượng test case, P và C lần lượt là số lượng người khám bệnh và số câu lệnh trong mỗi test case.