Task Description: Queue Stack

Creating a queue of integers is not a big challenge. You have tried this in your take home lab. Now, you have to implement a big queue of stacks.
You are given N operations. Each operation is one of the followings:

1. Create a stack with number x. It is guaranteed that a stack with this number has never been created before. This stack with number x should be enqueued to the big queue.
2. Insert an integer y to a stack with number x. It is guaranteed that stack with number x has been created before. You need to find the stack in the queue and insert the integer y into the stack. Note that you can only access the first stack in the queue. Hence you have to use a temporary big queue to store the stacks and put them back when the insertion to the correct stack is done.
3. Merge the first two stacks in the big queue. The integers in the first stack will be "pushed" onto the second stack. Naturally, the integers in the first stack will be in reversed order in the second stack when this operation is done. The first stack will be removed from the big queue. This removed stack will not be used again in the subsequent queries. Ignore this query if there are less than two stacks in the big queue.
4. Print the integer at the top of the stack which is at the front of the big queue.
    a. If the big queue is empty, print "BIG QUEUE IS EMPTY", otherwise
    b. If the stack at the front of the big queue is empty, print "STACK IS EMPTY",
    c. Otherwise, print the integer at the top of the stack.

For query of type 3, suppose we are moving stack A to stack B. Then the following algorithm will be executed ***`while (!A.empty()) { B.push(A.top()); A.pop(); }`***

Input

The first integer will consist of an integer N indicating the number of queries.
N lines follow. Each line will represent a query, be one of the following format.

- "CREATE x" indicates the query of the first type
- "INSERT y x" indicates the query of the second type. That is, insert the integer y into the stack with number x.
- "MERGE" indicates the query of the third type
- "PRINT" indicates the query of the fourth type

Output

For each PRINT query, you have to output an integer as explained above.

Sample Input

```
9
CREATE 1
PRINT
CREATE 2
INSERT 111 1
INSERT 1111 1
PRINT
INSERT 222 2
MERGE
PRINT
```

Sample Output

```
STACK IS EMPTY
1111
```

```
111
```

Constraint

N will be between 1 and 3000 inclusive.

The queue index will be between 1 and 1000000 inclusive.

Each integer to be inserted will be between 1 and 1000000 inclusive.