

# Follow the list

You start with a circular linked list with a single element "1". After that, you get a set of queries.

Query 1 x : Insert element x after current element.

Query 2 : Move to the next element.

Query 3 : Print the current element.

You have to keep processing these queries and printing results.

## Input

First line contains an integer q representing number of queries

Next few lines contain up to two integers. The first integer can be 1,2 or 3, denoting query type.

In case the query type is 1, then a second integer x will be present, denoting the value of the element to be inserted. In query type 2 and 3, there is no second element.

## Output

In case of query 1 and 2, there is no output, in case of query 3, print the current element (in a new line). The current element starts as the single element "1" of the linked list, and only changes with query type 2. Since it's a circular list, it will always be at some valid element.

## Constraints

q (number of queries)  $< 10^6$  , x (inserted element)  $< 10^7$

## Example

### Input

```
12
3
1 2
3
1 3
3
2
3
2
3
1 5
2
3
```

## Output

1  
1  
1  
3  
2  
5

## Explanation

In the beginning, the list is 1 pointing to itself (**bold** is current pointer)

3 - we print 1

1 2 - linked list becomes **1** -> 2

3 - we print 1 (the current element has not changed)

1 3 - linked list becomes **1** -> 3 -> 2 (notice how 3 is added after 1, not after 2)

3 - we print 1 ((the current element has still not changed)

2 - current element moves to 3 (1 -> **3** -> 2)

3 - we print 3

2 - current element moves to 2 (1 -> 3 -> **2**)

3 - we print 2

1 5 - linked list becomes 1 -> 3 -> **2** -> 5

2 - current element moves to 2 (1 -> 3 -> 2 -> **5**)

3 - we print 5

## Example

### Input

30  
2  
3  
3  
2  
3  
2  
1 94  
3  
1 21  
2  
1 40  
2  
1 55  
3

2  
3  
3  
2  
2  
1 68  
1 20  
1 83  
2  
3  
3  
2  
1 62  
2  
2  
3

### Output

1  
1  
1  
1  
40  
55  
55  
83  
83  
68