Circle Game

Sheila is a naughty girl. Yesterday she came up with a boring game and forced other classmates to play with her. The rule is simple. Some classmates will form a circle initially. Each classmate will be identified by a tag (an integer). Unluckily, you are chosen to be the judge of this game and answer Sheila's queries.

Sheila will give the following types of orders:

1. Student with tag *T* is added to the circle.

2. Remove a student from the circle

3. Do the query on current circle.

Please use c++ and write a class to implement linked list for this problem and do not use the libraries <list> <vector> etc.

Input

The input consists of a single test case. The first line contains an integer N (1 <= N <= 1000) indicating the initial number of students in the circle. Then follows N integers indicating the tags of the N students in clockwise direction. The circle begins with the first number in the sequence. Then there is an integer Q (1 <= Q <= 100) denoting the number of operations.

Initially, the reference point points to the first student, then after each operation, the reference point is updated.

There will be 3 kinds of operations in the following format:

(1) 1 i tag : A new student will be inserted after the i-th student counting from ***the reference point*** in clockwise direction. It is guaranteed that i is a positive integer. After this operation, the new reference point points to the newly inserted student.

(2) 2 i : remove the student at i-th position counting from ***the reference point*** in clockwise direction. The new reference point points to the student who follows the removed student.

(3) 3 i : print the tag of the i-th student counting from ***the reference point*** in clockwise direction. The new reference point points to this student.

It may happen that i >= (total number of students in the circle), in which case you may use modulo function to make your program faster.

Output

For each type (3) operation in the input, print the corresponding tag in one line.

Sample Input	Sample Output
6 1 5 3 3 2 10 4 1 3 20 3 3 2 5 3 6	2 5

Hints:

Initially, the circle is {1→5→3→3→2→10→back to 1}

After first operation, tag 20 is inserted after the 3rd element: {1→5→3→20→3→2→10} and 20 becomes the reference point.

The first query of position 3 is '2' at the moment. {20→3→2→10→1→5→3} and then 2 becomes the reference point. {2→10→1→5→3→20→3}

After the next operation "2 5", the 5th element '3' is removed and the follower '20' becomes the reference point: {20→3→2→10→1→5}.

The next query of position 6 is '5'.