

# Bugcat's Unique Queue

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         256 megabytes

Bugcat is currently managing a queue. Bugcat has a specific preference: it does not want any duplicate numbers to appear in the queue at the same time. Additionally, all numbers entering the queue must be within the range  $[1, m]$ .

You need to process  $Q$  operations. There are three types of operations:

- Type 1 (Push): Attempt to add the integer  $x$  to the back of the queue. If  $x$  is already present in the queue, do nothing. Otherwise, add  $x$  to the tail.
- Type 2 (Pop): Remove the element at the head of the queue. It is guaranteed that the queue is not empty when this operation is called.
- Type 3 (Query): Find the  $x$ -th number currently in the queue (indexed from 1, where 1 is the head). If the queue contains fewer than  $x$  elements, output  $-1$ .

Your task is to output the result for every Type 3 operation.

## Input

The first line contains two integers:  $Q$  (the number of operations) and  $m$  (the maximum value of elements), where  $1 \leq Q, m \leq 10^6$ .

The next  $Q$  lines describe the operations:

- 1  $x$ : Attempt to add  $x$  to the queue ( $1 \leq x \leq m$ ).
- 2: Pop the element at the head of the queue.
- 3  $x$ : Query the  $x$ -th element in the queue ( $1 \leq x \leq Q$ ).

## Output

For each Type 3 operation, print the result on a new line.

## Example

standard input	standard output
10 10	5
1 3	-1
1 5	3
2	8
3 1	
3 2	
1 3	
1 5	
1 8	
3 2	
3 3	