The Tower

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Houraisan Kaguya is a NEET princess who loves to watch videos at home. Recently, she often watches videos on a famous video-sharing website called Mikufans.

There is a useful feature on Mikufans that allows users to leave a message during the video playback, which is called *danmaku*. Sometimes, there are so many *danmaku* messages at the same time that Kaguya cannot take them all in.

Source: https://bilibili.com/video/BV1xx411c79H

For simplicity, we only focus on the top *danmaku* messages: top *danmaku* messages are displayed at the top of the video screen, and each message occupies exactly one line. There is no limit to the number of *danmaku* messages at the same time (although Kaguya's screen will only display the first 10^9 lines, the remaining messages will still be correctly maintained in the overflow area of the screen).

During the video playback, there may be three types of events:

1. A **new** user sends some top *danmaku* messages. Each message will be placed at the topmost empty line in order.
2. The *danmaku* messages from a specific user disappear, and the lines they are in become empty lines. The other messages will **not** be affected and still remain in their positions.
3. Kaguya is interested in a *danmaku* message, so she wants to know the sender of the top *danmaku* message at a specific line.

Kaguya has many videos to watch every day, and she is too busy to re-watch the video from the beginning, so she asks you for help. Please help her find the senders of the *danmaku* messages.

Input

The first line of input contains one integer n ($1 \le n \le 5 \times 10^5$), representing the number of events.

Each of the following n lines contains one event in order. Each event is described in one of the following formats:

- `1 k`: A new user sends k ($1 \leq k \leq 10^9$) top *danmaku* messages. The ID of the user is the smallest positive integer that has not been used before.
- `2 u`: The *danmaku* messages from user u disappeared. It is guaranteed that the ID is valid, and the *danmaku* messages from user u have not disappeared before.
- `3 l`: Kaguya wants to know the ID of the sender of the *danmaku* message at the l-th ($1 \leq l \leq 10^9$) topmost line. If that line is empty, the answer is defined as `0`.

Output

For each query of type `3`, output the answer in a single line.

Examples

standard input	standard output
7 1 2 1 4 3 3 2 1 3 2 1 4 3 7	2 0 3
5 3 6 3 8 1 2 1 5 3 2	0 0 1