A row of cookies
1 second 256 MB

You want to maintain a row of cookies. In this row, the cookies are ordered from left to right. Each cookie has a *unique* number on it. You start with a row with no cookies. An example of a row of cookies are shown below.

In this example, the number of the 1st cookie is 3. The 6th cookie has number 13.

You can perform two operations:

- **Insert** a new cookie after the first *k*-th cookie, where 0 <= *k* <= 5. If k=0, the new cookie is inserted as the first cookie. It is possible that there are less than *k* cookies in the row; in this case, the new cookie will be inserted at the end of the row.
- **Eat** the *k*-th cookie, where 1 <= *k* <= 5. Again, it is possible that there are less than *k* cooikes; in this case you do not eat any cookie.

The following example shows a few of these operations.

Step	Operation	k	New cookie number	Cookie row after the operation
(Initial)	-	-	-	(empty)
1	2 (eat)	3	-	(empty) (no 3rd cookie)
2	1 (insert)	1	1	1 (insert at the end)
3	1 (insert)	0	3	3, 1
4	1 (insert)	1	7	3, 7, 1
5	1 (insert)	3	29	3, 7, 1, 29
6	2 (eat)	3	-	3, 7, 29
7	1 (insert)	0	90	90, 3, 7, 29
8	2 (eat)	1	-	3, 7, 29
9	1 (insert)	5	100	3, 7, 29, 100 (insert at the end)
10	2 (eat)	5	-	3, 7, 29, 100 (no 5th cookie)

Write a program that perform these operations and output the final row at the end.

Input
The first line of the input contains an integer *T*, the number of operations to perform (1 <= T <= 100,000). The next *T* lines describe the operations.

Line 1 + *i*, for 1<=*i*<=*T*, describes the *i*-th operation in the following format. The first integer *L* is 1 (insert) or 2 (eat).

- If *L* = 1, there are two integers *k* and *x*. (0<=*k*<=5; 1<=*x*<=1,000,000,000) In this case, you want to insert a cookie with number *x* after the *k*-th cookie.
- If *L* = 2, there is an integer *k*. (1<=*k*<=5) In this case you want to eat the *k*-th cookie.

Output
You program should print the list of cookie numbers left after the last operations, one number per line.

Scoring
- There are 30% of test cases that T <= 1,000.
- There are 30% of test cases that all operations are insert (i.e., $L = 1$ in every operation).

Example

Input	Output
10 2 3 1 1 1 1 0 3 1 1 7 1 3 29 2 3 1 0 90 2 1 1 5 100 2 5	3 7 29 100