

## Card tricks

During summer vacation, when Red1 received his new deck of binary edition bicycle playing cards, he thought it would be a waste of their quality not to learn a few card tricks. After practicing for a few weeks, he learned the following three techniques:

- The **first** technique is fast insert that lets him insert any card of value  $v$  and shuffle the deck at lightning speed.
- The **second** technique is a quick retrieval that lets him find the biggest card of value  $x$  such that its value is less than or equal to the mean value of the cards currently in the deck. After this, he always opts to remove the card from the deck to make the trick seem more genuine. The card will always be shown when it is removed.
- The **third** and last trick is a spreading trick that lets him flip all of the cards in his deck at a snap of his fingers. This reveals the labels on the back of these cards which, for each card of value  $v$ , has a value  $v'$  such that it has all of the bits of  $v$  reversed when  $v$  is interpreted as a 32-bit unsigned integer.

To reverse the bits of a number, you take its binary representation and swap the bits in positions  $i$  and  $31 - i$ . For example, the number  $(3)_{10} = (\dots 011)_2$  becomes  $(110\dots)_2 = (3221225472)_{10}$  and  $(5)_{10} = (\dots 0101)_2$  becomes  $(1010\dots)_2 = (2684354560)_{10}$ .

Red1's friends are quite impressed by these tricks, but they are doubtful of his sleight of hand. They would like to test him by having him do these tricks  $Q$  times. Simulate each of the queries in order and for each query of the second trick, output the card that Red1 shows to his friends.

### Constraints

- $Q \leq 10^5$
- $v_i \leq 2^{32} - 1$

### I/O

The type of each query is either 1, 2, or 3 corresponding to the first, second and third tricks respectively. When the query is 1, there is an additional value to be read that corresponds to the label of the card being inserted.

Let  $T[i]$  be the type of the  $i$ th query and  $V[i]$  be the value of the card inserted when  $T[i] = 1$ ,  $O[i]$  be the answer to the  $i$ th query of type 2, and  $Q_2$  be the number of queries of type 2.

## Input

```
Q
T[0] // query of type 2 or 3
T[1] V[1] // query of type 1
...
T[Q-1]
```

## Output

```
O[0]
...
O[Q-1]
```

## Subtasks

The final grade for this task will be the given by the sum of points of all subtasks that have passed in at least one of your submissions.

Test group	Points	Constraints
1	8	$Q \leq 10^3$ , there are no queries of type 3.
2	13	There are no queries of type 3, all $v_i$ are distinct
3	17	All $v_i$ are powers of two.
4	25	$Q \leq 5 * 10^2$
5	37	No additional constraints

## Examples

### Example 1

#### Input :

```
6
1 1
1 2
1 3
2
3
2
```

**Output :**

2

2147483648