F2. Gellyfish and Lycoris Radiata (Hard Version)

time limit per test: 5 seconds
memory limit per test: 1024 megabytes

This is the hard version of the problem. The difference between the versions is that in this version, the time limit and the constraints on n and q are higher. You can hack only if you solved all versions of this problem.

Gellyfish has an array consisting of n sets. Initially, all the sets are empty.

Now Gellyfish will do q operations. Each operation contains one modification operation and one query operation, for the i-th ($1 \le i \le q$) operation:

First, there will be a modification operation, which is one of the following:

1. **Insert** operation: You are given an integer r. For the 1-th to r-th sets, insert element i. Note that the element inserted here is i, the index of the operation, not the index of the set.
2. **Reverse** operation: You are given an integer r. Reverse the 1-th to r-th sets.
3. **Delete** operation: You are given an integer x. Delete element x from all sets that contain x.

Followed by a query operation:

- **Query** operation: You are given an integer p. Output the smallest element in the p-th set (If the p-th set is empty, the answer is considered to be 0).

Now, Flower needs to provide the answer for each query operation. Please help her!

Additional constraint on the problem: Gellyfish will only give the next operation after Flower has answered the previous query operation. That is, you need to solve this problem **online**. Please refer to the input format for more details.

Input

The first line contains two integers n and q ($1 \le n, q \le 3 \cdot 10^5$) — the number of the sets and the number of operations.

As you need to respond to the operations online, the operations will be encoded.

The i-th line of the following q lines contains three integers a, b, and c ($1 \le a \le 3$, $1 \le c \le n$) — describing the i-th operation in an encoded form.

Here, a represents the type of modification operation. Among them, $a = 1$ represents **Insert** operation, $a = 2$ represents **Reverse** operation, $a = 3$ represents **Delete** operation.

- If $a = 1$, then the modification operation is the **Insert** operation. It will be guaranteed that $1 \le b \le n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 2$, then the modification operation is the **Reverse** operation. It will be guaranteed that $1 \le b \le n$. r will be calculated as $r = (b + \text{ans}_{i-1} - 1) \bmod n + 1$.
- If $a = 3$, then the modification operation is the **Delete** operation. It will be guaranteed that $1 \le b \le q$. x will be calculated as $x = (b + \text{ans}_{i-1} - 1) \bmod q + 1$.

For the query operation, p will be calculated as $p = (c + \text{ans}_{i-1} - 1) \bmod n + 1$.

Here ans_i ($1 \le i \le q$) represents the answer to the query operation in the i-th operation. Additionally, we define $\text{ans}_0 = 0$.

Output

For each query operation, output the answer to the query.

Example

input Copy

```
5 10
1 2 2
2 3 1
1 5 3
```

Codeforces Round 1028 (Div. 1)
Finished
Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
322841962	Jun/04/2025 13:33	Accepted

→ Problem tags

data structures

No tag edit access

→ Contest materials

- Announcement (en)
- Tutorial (en)

```
2 2 5
1 5 2
2 4 4
3 2 2
3 1 2
3 10 5
3 2 4
```

output

Copy

```
1
0
1
1
3
1
0
5
0
0
```

Note

All the sets are empty in the beginning, so the array is $[\{\},\{\},\{\},\{\},\{\}]$.

With the decoding method given before, we can see what happens in each operation:

1. For the first operation: $a=1,r=2,p=2$. The modification operation is an **Insert** operation; element 1 is inserted into the first two sets; so the array becomes $[\{1\},\{1\},\{\},\{\},\{\}]$, and the smallest element in the second set is 1.
2. For the second operation: $a=2,r=4,p=2$. The modification operation is a **Reverse** operation; the first four sets are reversed; so the array becomes $[\{\},\{\},\{1\},\{1\},\{\}]$, and the second set is empty, which means the answer is 0.
3. For the third operation: $a=1,r=5,p=3$. The modification operation is an **Insert** operation; element 3 is inserted into all the sets; so the array becomes $[\{3\},\{3\},\{1,3\},\{1,3\},\{3\}]$, and the smallest element in the third set is 1.
4. For the fourth operation: $a=2,r=3,p=1$. The modification operation is a **Reverse** operation; the first three sets are reversed; so the array becomes $[\{1,3\},\{3\},\{3\},\{1,3\},\{3\}]$, and the smallest element in the first set is 1.
5. For the fifth operation: $a=1,r=1,p=3$. The modification operation is an **Insert** operation; element 5 is inserted into the first set; so the array becomes $[\{1,3,5\},\{3\},\{3\},\{1,3\},\{3\}]$, and the smallest element in the third set is 3.
6. For the sixth operation: $a=2,r=2,p=2$. The modification operation is a **Reverse** operation; the first two sets are reversed; so the array becomes $[\{3\},\{1,3,5\},\{3\},\{1,3\},\{3\}]$, and the smallest element in the second set is 1.
7. For the seventh operation: $a=3,x=3,p=3$. The modification operation is a **Delete** operation; element 3 is deleted from all the sets; so the array becomes $[\{\},\{1,5\},\{\},\{1\},\{\}]$, and the third set is empty, which means the answer is 0.
8. For the eighth operation: $a=3,x=1,p=2$. The modification operation is a **Delete** operation; element 1 is deleted from all the sets; so the array becomes $[\{\},\{5\},\{\},\{\},\{\}]$, and the smallest element in the second set is 5.
9. For the ninth operation: $a=3,x=5,p=5$. The modification operation is a **Delete** operation; element 5 is deleted from all the sets; so the array becomes $[\{\},\{\},\{\},\{\},\{\}]$, and the fifth set is empty, which means the answer is 0.
10. For the tenth operation: $a=3,x=2,p=4$. The modification operation is a **Delete** operation; element 2 is deleted from all the sets; so the array becomes $[\{\},\{\},\{\},\{\},\{\}]$, and the fourth set is empty, which means the answer is 0.

Please note that although we have not inserted element 2 into the sets, we still delete element 2 from all the sets in the tenth operation, which means that the **Delete** operation doesn't necessarily require the existence of a set to contain the deleted element. It also shows that it is possible to have two **Delete** operations that delete the same element.