Problem D. Segments

Input file:	`standard input`
Output file:	`standard output`
Time limit:	~~3 seconds~~ 5 seconds (because of slow server)
Memory limit:	40 megabytes

There is a multiset of segments S. Difference between multiset and set is that multiset allows multiple instances of one segment, unlike a set.

Given two integer numbers n and t. You have n operations of following types that are made over the multiset:

1. Insert segment $[l, r]$ into the multiset S. The segment is assigned with id — minimum positive integer number that was not assigned to any other segment before.

2. Erase the segment with assigned number id from the multiset S. It is guaranteed that at the moment of erasing there is a segment in the multiset S with assigned number id.

3. Count the number of segments from the multiset S that has at least k *integer points* in common with given segment $[l, r]$.

Integer point x is common for both segments $[l_i, r_i]$ and $[l_j, r_j]$, if $l_i \le x \le r_i$ and $l_j \le x \le r_j$.

Input

The first line of input contains two integer numbers n and t $(1 \le n \le 2 \cdot 10^5, 0 \le t \le 1)$ — number of operations and constant number. Each of next n lines describes one query.

1. Queries of first type are given in following format: 1 a_i b_i $(0 \le a_i, b_i \le 2 \cdot 10^9)$.

2. Queries of second type are given in following format: 2 id_i $(1 \le id_i \le n)$.

3. Queries of third type are given in following format: 3 a_i b_i k_i $(0 \le a_i, b_i, k_i \le 2 \cdot 10^9)$.

Please note that end points of segments $[l_i, r_i]$ for queries of type 1 and 3 are **encoded**, in order to decode them you need to perform the following transformations:

$$l_i = (a_i \oplus (t * lastans)) \quad r_i = (b_i \oplus (t * lastans))$$

where $lastans$ — last answer to the query of type 3 (initially $lastans$ equals to 0). If it turned out that l_i is greater than r_i, you should swap the values of l_i and r_i.

It is guaranteed that there will be at least one query of type 3 in input.

Here $\oplus$ denotes the bitwise XOR operation.

Consider that problem has **unusual memory limit**.

Output

For each query of type 3 print answer in separate line.

Scoring

This task contains six subtasks:

1. $n \le 5 \cdot 10^3$. Scored 7 points.

2. $n \le 10^5$. First comes queries of type 1, then of type 3 and there is no query of type 2. Scored 15 points.

3. $n \le 2 \cdot 10^5$, $k_i = 1$ for all third type queries. Scored 16 points.

4. $n \le 10^5$, $t = 0$. Scored 17 points.

5. $n \le 10^5$. Scored 20 points.

6. $n \leq 2 \cdot 10^5$. Scored 25 points.

Examples

standard input	standard output

6 1
1 1 2
3 2 4 2
1 3 5
3 2 3 1
2 1
3 0 3 1
	0
	2
	0

6 0
1 3 10
1 3 5
3 6 10 6
2 1
1 3 10
3 6 4 2
	0
	2