

Задача A. Connect and Disconnect

Имя входного файла:	<code>connect.in</code>
Имя выходного файла:	<code>connect.out</code>
Ограничение по времени:	3 seconds (<i>4 seconds for Java</i>)
Ограничение по памяти:	256 Mebibytes

Do you know anything about DFS, Depth First Search? For example, using this method, you can determine whether a graph is connected or not in $O(E)$ time. You can even count the number of connected components in the same time.

Do you know anything about DSU, Disjoint Set Union? Using this data structure, you can process queries like “Add an edge to the graph” and “Count the number of connected components in the graph” fast.

And do you know how to solve Dynamic Connectivity Problem? In this problem, you have to process three types of queries fast:

1. Add an edge to the graph
2. Delete an edge from the graph
3. Count the number of connected components in the graph

Формат входных данных

At the first moment, the graph is empty.

The first line of file contains two integers N and K —number of vertices and number of queries ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Next K lines contain queries, one per line. There are three types of queries:

1. $+ u v$: add an edge between vertices u and v . It is guaranteed that there is no such edge in the graph at the time of the query.
2. $- u v$: remove an edge between vertices u and v . It is guaranteed that this edge is present in the graph at the time of the query.
3. $?$: count the number of connectivity components in the graph at the time of the query.

Vertices are numbered 1 through N . No query will have $u = v$. The graph is undirected.

Формат выходных данных

For each ‘?’ query, output the number of connectivity components in the graph at the time of the query on a single line.

Пример

connect.in	connect.out
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	