B Palindromic Walk

TIME LIMIT: 8.0S
MEMORY LIMIT: 1024MB

Bob loves walks. But not just any walks: Bob only enjoys palindromic walks.

The graph where Bob lives has n nodes. Initially, no two nodes are connected. You need to process q queries of two types:

- "`1` u v c": Add a bidirectional edge between two different nodes u and v labeled with the character c. Before this query, it is guaranteed that there is no edge between u and v.
- "`2` u v": Answer whether there exists a walk between two different nodes u and v such that the characters on the edges in the walk form a palindrome. If such a walk exists, output 1; otherwise, output 0.

A walk from u to v is an alternating sequence of nodes and edges starting at node u and ending at node v. In this walk, the same edge may be used multiple times.

A palindrome is a sequence of characters that reads the same forward and backward (for example, "`radar`", "`abba`", "`zzz`").

INPUT

The first line contains two integers, n and q: the number of nodes and the number of queries, respectively ($2 \leq n \leq 1000$, $1 \leq q \leq 3 \cdot 10^5$).

Each of the following q lines contains a query in the format described above: either "`1` u v c" or "`2` u v" (u and v are integers, $1 \leq u, v \leq n$, $u \neq v$, and c is a lowercase English letter). All bidirectional edges added by queries of type `1` are distinct.

OUTPUT

For each query of the second type, output a line with a single integer: 1 if such a walk exists, or 0 if it does not.

SAMPLES

Sample input 1	Sample output 1
`5 6`	`0`
`1 1 2 a`	`1`
`1 2 3 b`	`1`
`2 1 3`	
`2 2 3`	
`1 3 4 a`	
`2 1 4`	

Sample input 2	Sample output 2
5 7 1 1 2 a 2 1 2 1 2 3 b 1 3 4 a 2 2 4 1 4 5 a 2 2 5	1 0 1