Games of Chess

Time limit: 2 seconds
Memory limit: 1024 megabytes

There are n friends and n houses in Chess City, both numbered from 1 to n, where friend i lives in house i. The houses are connected by m bidirectional roads, forming a connected network.

Chess City also has n virtual chess clubs, numbered from 1 to n. Each friend must choose exactly one club to join. These choices do not need to be distinct, so some clubs might not have any members.

When friend i hosts a chess party, it is attended by every friend who belongs to the same club as friend i and lives in a house directly connected to house i by a road. The party is considered *successful* if the total number of attendees, including friend i, is even; in this case, they can all play chess simultaneously.

Choose a club for each friend so that every friend's party is successful, or report that it is impossible.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and m, denoting the number of houses and the number of roads ($2 \le n \le 10^5$; $n - 1 \le m \le 2 \cdot 10^5$).

Each of the following m lines contains two integers u and v, describing a bidirectional road between houses u and v ($1 \le u, v \le n$; $u \ne v$). No two houses are connected by more than one road. It is possible to get from any house to any other one using the roads.

It is guaranteed that the sum of n over all test cases does not exceed 10^5, and the sum of m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer -1 if it is impossible to assign a chess club to each friend so that every friend's party is successful.

Otherwise, print n integers $c_1, c_2, \ldots, c_n$, where c_i is the number of the chess club that friend i should join ($1 \le c_i \le n$). If there are multiple answers, print any of them.

Example

standard input	standard output
3 2 1 1 2 3 3 1 2 2 3 3 1 8 10 1 2 1 4 1 7 5 2 5 4 5 7 5 3 2 6 2 8 6 8	1 1 -1 3 3 6 6 6 2 6 2