

Problem

Minesweeper

Time limit: 2 seconds

This is a communication problem.

Little Cyan Fish loves the game *Minesweeper* a lot, and he really wants to play this game during the Universal Cup Finals. Therefore, he gives you the opportunity to experience another Minesweeper-like game, called *CyanSweeper*, which is played on an undirected connected planar graph instead of a grid.

To play this game, he gives you a planar graph with n vertices and m edges. The vertices are labeled from 1 to n . There are k special vertices containing bombs, and the game consists of two phases.



A dangerous mine!

First run. Little Cyan Fish gives you the graph and the labels of the k bomb vertices b_1, b_2, \dots, b_k . You must produce an array a_1, a_2, \dots, a_n as follows:

- For every vertex v without a bomb, a_v must equal the number of bomb-containing neighbors of v (the standard *Minesweeper* count).
- For every vertex v with a bomb, a_v may be any integer in $\{1, 2, \dots, k\}$, subject to the constraint that the values $a_{b_1}, a_{b_2}, \dots, a_{b_k}$ at the bomb vertices form a permutation of $\{1, 2, \dots, k\}$.

In addition to the array a , Little Cyan Fish asks you to transmit a non-empty binary string s of length at most 100 to the second run.

Second run. Little Cyan Fish gives you the same graph (with the same vertex labels), the array a_1, \dots, a_n exactly as produced in the first run, and the transmitted binary string s . However, you will not be given the labels of the bomb vertices. It is your job to output the set of bomb vertices based on the graph, the array a , and the transmitted string s .

Write a program to play *CyanSweeper*.

Input

There are multiple test cases. The first line of the input contains two integers r ($r \in \{1, 2\}$) and T ($1 \leq T \leq 10$), indicating the run number and the number of test cases.

For each test case:

- If $r = 1$, the first line contains three integers n, m, k ($1 \leq k \leq n \leq 5 \times 10^5, n - 1 \leq m \leq 10^6$). The next m lines each contain two integers u and v ($1 \leq u, v \leq n, u \neq v$), describing an undirected edge. The graph is guaranteed to be connected, simple, and planar. The next line contains k distinct integers b_1, b_2, \dots, b_k ($1 \leq b_i \leq n$), the indices of the bomb vertices.
- If $r = 2$, the first line contains two integers n and m . The next m lines describe the edges in the same format as above. The following line contains n integers a_1, a_2, \dots, a_n , exactly as the first-run solution produced them. The last line contains the binary string s that the first-run solution transmitted.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5 and the sum of m over all test cases does not exceed 10^6 .

Output

If $r = 1$, for each test case output the array a_1, a_2, \dots, a_n on one line separated by spaces, followed by the binary string s on the next line. The array must satisfy the rules described above, and s must consist only of the characters 0 and 1 and have length at least 1 and at most 100.

If $r = 2$, for each test case output two lines. The first line contains a single integer k , the number of bombs. The second line should contain k distinct integers corresponding to the bomb vertex indices. You may output the bomb vertex indices in any order. The reported set must coincide exactly with the set of bombs from the corresponding first-run input.

Testing Tool: A testing tool is provided to help contestants develop and test their solutions. You can download this tool from the DOMjudge system. Executing the tool with a “-h” option should describe how to use the tool. The testing tool will only implement some test scenarios and only some functionality of the real judge program.