Problem C. Cactus Not Enough

Time limit: 3 seconds
Memory limit: 512 megabytes

There was no problem about a cactus at the NERC 2020 online round. That's a bad mistake, so judges decided to fix it. You shall not pass to the World Finals 2021 without solving a problem about a cactus!

A *cactus* is a connected undirected graph in which every edge lies on at most one simple cycle. Intuitively, a cactus is a generalization of a tree where some cycles are allowed. Multiedges (multiple edges between a pair of vertices) and loops (edges that connect a vertex to itself) are not allowed in a cactus.

Cher has got a cactus. She calls cactus *strong* if it is impossible to add an edge to it in such a way that it still remains a cactus. But Cher thinks her cactus is not strong enough. She wants to add the smallest possible number of edges to it to make it strong, i. e. to create a new cactus with the same vertices, so that the original cactus is a subgraph of the new one, and it is impossible to add another edge to it so that the graph remains a cactus. Cher hired you to do this job for her. So... it's on you!

Input

The input consists of one or more independent test cases.

The first line of each test case contains two integers n and m ($1 \le n \le 10^5$, $0 \le m \le 10^5$), where n is the number of vertices in the graph. Vertices are numbered from 1 to n. Edges of the graph are represented by a set of edge-distinct paths, where m is the number of such paths.

Each of the following m lines contains a path in the graph. A path starts with an integer number s_i ($2 \le s_i \le 1000$) followed by s_i integers from 1 to n. These s_i integers represent vertices of a path. Adjacent vertices in a path are distinct. The path can go through the same vertex multiple times, but every edge is traversed exactly once in the whole test case. There are no multiedges in the graph (there is at most one edge between any two vertices).

The last line of the input after all test cases always contains two zeros. It does **not** define a test case. It just marks the end of the input and does not require any output.

All graphs in the input are cacti. The total sum of all values of n and the total sum of all values of m throughout the input both do not exceed 10^5.

Output

For each test case, first output the line with the minimal possible number of additional edges A. Then output A lines, each describing one edge as u_i v_i, where u_i and v_i are the numbers of vertices to connect. After adding these edges, the resulting graph must be a strong cactus.

Example

standard input	standard output
6 1 7 1 2 5 6 2 3 4 3 1 4 1 2 3 1 5 2 3 1 3 5 3 1 2 4 7 2 6 1 2 3 4 5 3 3 6 5 7 0 0	1 1 4 0 1 5 4 2 1 3 6 7