Problem B. Cactus

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

A *cactus* is a simple undirected connected graph in which every edge belongs to at most one simple cycle.

Now, there is a cactus accepting the following two operations:

1. Select a vertex with an odd degree in the graph, and remove all edges connected to it.
2. Make a copy of the current graph, and then draw additional edges between the corresponding vertices in the current graph and in the copy, forming a new graph. Formally speaking, suppose the current graph has n vertices in total, labeled from 1 to n. First, add n new vertices labeled from $n+1$ to $2n$. Then, for every edge (u, v) in the current graph, add an edge $(u+n, v+n)$. Lastly, add the edges $(1, n+1)$, $(2, n+2)$, $\ldots$, $(n, 2n)$. If the current graph has n vertices and m edges, the new graph has $2n$ vertices and $2m+n$ edges.

Because the second operation is costly, it can only be used at most once. The first operation can be used any number of times in any order.

Find a sequence of operations such that, after all operations in the sequence, the final graph has the least possible number of edges.

Input

The first line of input contains two integers n and m, the number of vertices and the number of edges in the initial graph ($1 \le n \le 3 \cdot 10^5$, $n - 1 \le m \le \frac{3(n-1)}{2}$).

Each of the next m lines contains two integers u and v denoting the endpoints of an edge ($1 \le u, v \le n$). The graph is connected and contains no parallel edges and no self-loops.

Output

On the first line, print two integers m' and K, the number of edges left in the final graph and the total number of operations.

Then print K more lines. Each line represents an operation:

1. When using the first operation on vertex x, print "`1` x".
2. When using the second operation, just print "`2`".

If there are several optimal answers, print any one of them.

Examples

standard input	standard output
3 3 1 2 1 3 2 3	0 6 2 1 1 1 5 1 2 1 4 1 3
7 7 1 2 1 3 2 3 2 4 2 5 3 6 3 7	0 14 1 4 1 5 1 6 1 7 2 1 1 1 4 1 5 1 6 1 7 1 9 1 2 1 8 1 3