

Essential

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 64 megabytes

We are given an undirected **connected** graph with n vertexes and m edges.

Find the greatest number of edges we can remove such that if we do all pairs shortest paths on the reduced graph, it will be the same as it used to be before all the reductions were done.

Input

The input contains n and m , the number of vertexes and edges of the graph ($1 \leq n \leq 100$), ($1 \leq m \leq n * (n + 1) / 2$).

The next m lines contain the edges of the graph, which are described by the ends of the edges and the cost of the edge ($1 \leq x, y \leq n$, $x \neq y$, $1 \leq cost \leq 10^5$).

The graph is connected.

Output

The output will contain on the first line x , the biggest number of edges we can remove such that the condition from the statement is fulfilled.

On the next x lines you need to print the edges which will be removed, in lexicographical order, and for each edge we will print the vertex with the smaller number first.

Example

standard input	standard output
6 10 1 4 7 2 3 8 1 5 3 4 6 4 2 5 9 1 6 5 3 5 1 6 3 7 2 1 6 1 3 7	2 1 3 2 5