

Problem I. Hard Times for Your Data

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

There is a storage cluster of n nodes. Each document is stored in exactly two replicas in some two nodes.

Hard times are coming. Budget is shrinking. Bits of storage become very expensive, so some optimization is required. Maximum capacity was determined for each node; that is, how many replicas can be stored in it.

Since existing replicas cannot be moved, we've made a willful decision to drop some documents from the cluster to satisfy new requirements. We want to utilize the cluster completely, so after the cleanup each node should store exactly as many documents as its capacity allows.

We somehow know that it is possible to remove some documents such that each node is utilized perfectly. However, it turned out to be very hard to find exact documents to remove. Can you help us?

Input

In the first line of input, there are two integers n and m ($1 \leq n \leq 500$, $0 \leq m \leq n \cdot (n - 1) / 2$), the number of nodes and the number of sets of documents.

The next line contains n integers f_1, \dots, f_n ($0 \leq f_i \leq 10^9$), where f_i denotes the capacity of node i .

Each of the next m lines contains three integers u_i, v_i, c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq c_i \leq 10^4$). The description means that there are c_i documents with replicas in nodes u_i and v_i .

All pairs of nodes in the description of different document sets are distinct.

Output

First, print a number k : the number of pairs of nodes with common replicas after the removal.

Then print k lines, each containing three integers, in the same format as in the input: the pair of nodes and the number of documents stored in these nodes.

After the removal, each pair of nodes can not have more documents than it had before. Furthermore, the number of documents stored in the i -th node must equal f_i .

It is guaranteed that the answer exists. If there are multiple answers, print any one of them.

Example

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