

Problem F. Planar Max Cut

Input file: *standard input*
Output file: *standard output*
Time limit: 6 seconds
Memory limit: 512 mebibytes

Some of you may know how to find a Minimum Cut in a graph. Some of you may also have heard about the fact that Maximum Cut problem is NP-complete (more formally, the decision version of finding the cut of value at least k is NP-complete). I bet that some of you even thought about something like: “Hmm, what if I simply negate all costs and transform a Max Cut problem into a Min Cut problem? Where are my Turing Award and billion dollars... Oh, snap, I got it.”

It turns out that, for some restricted classes of graphs, problems may become easier than in the general case. Consider the Planar Maximum Cut problem which is formulated as follows. Consider an undirected graph G consisting of n vertices $V = \{v_1, v_2, \dots, v_n\}$ and m edges $E = \{(v_{a_1}, v_{b_1}), (v_{a_2}, v_{b_2}), \dots, (v_{a_m}, v_{b_m})\}$. The graph is planar, and you are given its embedding into the plane: besides the description of the graph, you know the coordinates of the vertices such that, if we draw segments corresponding to the edges of the graph, no two edges would share an internal point. There is also an integer cost c_j associated with j -th edge of the graph.

Your task is to partition vertices of the graph into two disjoint sets A and B ($A \cup B = V$) such that the total cost of cut edges is maximum possible. An edge is called a cut edge if exactly one of its endpoints belongs to A and exactly one belongs to B .

Input

The first line of input contains two integers n and m ($1 \leq n \leq 200$, $1 \leq m \leq 1000$), the number of vertices of the graph and the number of edges of the graph respectively.

The i -th of the following n lines contains two integers x_i and y_i ($-10^4 \leq x_i, y_i \leq 10^4$), the coordinates of i -th vertex in the planar embedding. No two points coincide.

The j -th of the following m lines contains three integers a_j , b_j and c_j ($1 \leq a_j, b_j \leq n$, $a_j \neq b_j$, $0 \leq c_j \leq 10^5$), the endpoints of the j -th edge and the cost associated with it.

Output

On the first line, print the maximum possible total cost of the cut edges.

On the second line, print n integers s_1, s_2, \dots, s_n ($s_i \in \{0, 1\}$) where $s_i = 0$ if $i \in A$ and $s_i = 1$ if $i \in B$ in the maximum cut. If there are several possible answers, print any one of them.

Example

standard input	standard output
4 5	21
0 0	0 0 1 1
2 0	
0 2	
2 2	
1 2 3	
2 4 6	
3 4 4	
1 3 7	
2 3 8	