

Problem C

Confederation

File: confederacao.[c|cpp|java]

The Galactic Confederation is planning an administrative reform, to better manage its resources. For that, the Confederation divided the whole space into regions. To define the regions, initially a set of planes was specified, and the regions were defined by the cuts these planes made in the space. Notice that some regions are unlimited, but there may be limited regions. The set of planes was chosen so that no plane intercepts the orbit of a planet, and therefore each planet moves within only one region during its orbit (that is, a planet inside a region will never cross a plane to another region).

Your task is to determine, given the equations of the planes and the positions of the planets, how many planets exist within the region with the largest number of planets (in other words, what is the maximum number of planets inside any region).

Input

The input contains several test cases. The first line of a test case contains two integers M ($1 \leq M \leq 500$) and N ($1 \leq N \leq 10000$), indicating respectively the number of planes and the number of planets. Each of the M following lines contains four integers A, B, C and D ($-10000 \leq A, B, C, D \leq 10000$), the coefficients and the free term of the equation $Ax + By + Cz = D$ which defines each plane. Each of the following N lines contains three integers X, Y and Z ($-10000 \leq X, Y, Z \leq 10000$), representing the position (X, Y, Z) of a planet.

Output

For each test case in the input your program must produce a single line containing a single integer, the number of planets in the region which contains the largest number of planets.

Example

Input	Output
2 5	3
1 0 0 1	5
2 0 0 8	
0 1 0	
2 2 2	
3 3 3	
5 5 5	
2 18 4	
4 8	
0 0 1 1	
1 0 1 2	
-1 1 1 3	
-1 -1 1 3	
0 0 5	
0 0 4	
0 0 -2	
1 0 5	
40 19 104	
13 26 84	
89 -45 18	
3 1 0	