D. Nanami's Power Plant

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Nanami likes playing games, and is also really good at it. This day she was playing a new game which involved operating a power plant. Nanami's job is to control the generators in the plant and produce maximum output.

There are n generators in the plant. Each generator should be set to a generating level. Generating level is an integer (possibly zero or negative), the generating level of the i-th generator should be between l_i and r_i (both inclusive). The output of a generator can be calculated using a certain quadratic function $f(x)$, where x is the generating level of the generator. Each generator has its own function, the function of the i-th generator is denoted as $f_i(x)$.

However, there are m further restrictions to the generators. Let the generating level of the i-th generator be x_i. Each restriction is of the form $x_u \le x_v + d$, where u and v are IDs of two different generators and d is an integer.

Nanami found the game tedious but giving up is against her creed. So she decided to have a program written to calculate the answer for her (the maximum total output of generators). Somehow, this became your job.

Input

The first line contains two integers n and m $(1 \le n \le 50;\ 0 \le m \le 100)$ — the number of generators and the number of restrictions.

Then follow n lines, each line contains three integers a_i, b_i, and c_i $(|a_i| \le 10;\ |b_i|, |c_i| \le 1000)$ — the coefficients of the function $f_i(x)$. That is, $f_i(x) = a_i x^2 + b_i x + c_i$.

Then follow another n lines, each line contains two integers l_i and r_i $(-100 \le l_i \le r_i \le 100)$.

Then follow m lines, each line contains three integers u_i, v_i, and d_i $(1 \le u_i, v_i \le n;\ u_i \ne v_i;\ |d_i| \le 200)$, describing a restriction. The i-th restriction is $x_{u_i} \le x_{v_i} + d_i$.

Output

Print a single line containing a single integer — the maximum output of all the generators. It is guaranteed that there exists at least one valid configuration.

Examples

input

```
3 3
0 1 0
0 1 1
0 1 2
0 3
1 2
-100 100
1 2 0
2 3 0
3 1 0
```

output

```
9
```

input

```
5 8
1 -8 20
2 -4 0
-1 10 -10
0 1 0
```

```
0 -1 1
1 9
1 4
0 10
3 11
7 9
2 1 3
1 2 3
2 3 3
3 2 3
3 4 3
4 3 3
4 5 3
5 4 3
```

output

```
46
```

Note

In the first sample, $f_1(x) = x$, $f_2(x) = x + 1$, and $f_3(x) = x + 2$, so we are to maximize the sum of the generating levels. The restrictions are $x_1 \le x_2$, $x_2 \le x_3$, and $x_3 \le x_1$, which gives us $x_1 = x_2 = x_3$. The optimal configuration is $x_1 = x_2 = x_3 = 2$, which produces an output of 9.

In the second sample, restrictions are equal to $|x_i - x_{i+1}| \le 3$ for $1 \le i < n$. One of the optimal configurations is $x_1 = 1$, $x_2 = 4$, $x_3 = 5$, $x_4 = 8$ and $x_5 = 7$.