F. Variables and Operations

time limit per test: 5 seconds
memory limit per test: 512 megabytes

There are n variables; let's denote the value of the i-th variable as a_i.

There are also m operations which will be applied to these variables; the i-th operation is described by three integers x_i, y_i, z_i. When the i-th operation is applied, the variable x_i gets assigned the following value: $\min(a_{x_i}, a_{y_i} + z_i)$.

Every operation will be applied **exactly once**, but their order is not fixed; they can be applied in any order.

Let's call a sequence of initial variable values $a_1, a_2, \ldots, a_n$ **stable**, if no matter in which order we apply operations, the resulting values will be the same. If the resulting value of the i-th variable depends on the order of operations, then the sequence of initial variable values is called i**-unstable**.

You have to process q queries. In each query, you will be given initial values $a_1, a_2, \ldots, a_n$ and an integer k; before applying the operations, you can at most k times choose a variable and decrease it by 1. For every variable i, you have to independently determine if it is possible to transform the given values into an i-unstable sequence.

Input

The first line contains two integers n and m ($2 \le n \le 500$; $1 \le m \le 4 \cdot 10^5$) — the number of variables and operations, respectively.

Then, m lines follow. The i-th of them contains three integers x_i, y_i, z_i ($1 \le x_i, y_i \le n$; $x_i \ne y_i$; $0 \le z_i \le 10^5$) — the description of the i-th operation.

The next line contains one integer q ($1 \le q \le 1000$) — the number of queries.

Each query consists of two lines:

- the first line contains one integer k ($0 \le k \le 10^9$) — the maximum number of times you can choose a variable and decrease it by 1;
- the second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the initial values of the variables.

Output

For each query, print a string of n zeroes and/or ones. The i-th character should be `1` if it is possible to obtain an i-unstable sequence, or `0` otherwise.

Examples

input Copy

```
4 5
2 1 10
3 2 5
1 4 8
1 2 6
3 1 17
3
0
20 0 15 5
10
20 0 15 5
30
20 0 15 5
```

Educational Codeforces Round 180 (Rated for Div. 2)

Finished

Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
325961650	Jun/25/2025 08:48	Accepted
325713636	Jun/23/2025 17:43	Wrong answer on test 1

→ Problem tags

graphs greedy shortest paths

No tag edit access

→ Contest materials

- Announcement

output Copy

```
0000
0000
0110
```

input Copy

```
3 5
1 2 100
1 2 10
1 3 5
1 2 100
3 2 5
1
1000000000
0 0 0
```

output Copy

```
000
```

input Copy

```
3 4
2 3 5
1 2 0
3 1 4
1 3 4
10
5
7 5 3
2
5 7 0
1
1 1 1
5
3 0 1
0
5 3 5
5
6 0 4
5
1 5 6
1
7 7 2
1
1 6 6
4
7 7 2
```

output Copy

```
000
000
000
001
000
001
001
000
000
000
```

Note

Consider the first example. If the initial variable values are $[20, 0, 15, 5]$, the resulting values will be $[6, 0, 5, 5]$ with any order of operations. Decreasing the variables 10 times is not enough. However, if we can apply no more than 30 changes, we can decrease the 1-st variable by 2, and the 4-th variable by 25, we get initial values equal to $[18, 0, 15, -20]$, and this sequence is 2-unstable and 3-unstable:

- if you apply the operations in the order they are given, you will get $[-12, 0, 5, -20]$;
- however, if you apply the operations in order $[3, 2, 4, 1, 5]$, you will get $[-12, -2, 5, -20]$;
- and if you apply the operations in order $[3, 4, 5, 1, 2]$, you will get $[-12, -2, 3, -20]$.

Server time: Jun/25/2025 12:48:46^{UTC+7} (n2).
Desktop version, switch to mobile version.
Privacy Policy | Terms and Conditions

Supported by