Fair-share Scheduling?

Description

You are a busy earthworm loving your girl friend so much that you want to accompany her as much time as you can. However, you always try to get all of your projects done before meeting her, thus you sometimes work for a whole week without seeing her. Your girl friend had complained about that several times, so you decide to change your policy this summer to let you meet her every week.

You partition each project P_i into Q_i parts, deciding to start working on it at the S_i-th day of summer. You will need C_i days to finish one part, and you plan to finish a part every D_i days. To make things easier, you do not need to work on the project at S_i-th day if you can finish the first part in the next $D_i - 1$ days.

Now, you want to maximize the minimum days you can accompany your girl friend of each week (so you can accompany your girl friend every week :p). Note that you hate to do many things in one day, hence you can only do **one** thing a day, including accompany your girl friend.

Input

The first line contains an integer T, the number of test cases. The first line of a test case will contains three integers N, M, K, indicating the number of weeks of this summer, how many days there are in a week and the number of projects you need to do. Each of the following K lines contains four integers S_i, Q_i, D_i, C_i, which are described above.

- $1 \leq T \leq 30$
- $1 \leq N, M$
- $N \times M \leq 500$
- $0 \leq K \leq 50$
- $1 \leq S_i \leq S_i + Q_i \times D_i - 1 \leq N \times M$
- $1 \leq Q_i$
- $1 \leq C_i \leq D_i$

Output

Output a number for each test case, representing the minimum number of days you can accompany your girl friend of each week if you schedule your projects in the optimal way.

Sample Input

```
2
2 3 1
3 2 2 1
1 1 2
1 1 1 1
1 1 1 1
```

Sample Output

```
2
0
```