## Problem A. ACM Rank Table

Author: A. Klenin

|  |  |
| --- | --- |
| Input file: | input.txt |
| Output file: | output.txt |
| Time limit: | 2 sec |
| Memory limit: | 16 Mb |

### *Statement*

ACM contests, like the one you are participating in, are hosted by the special software. That software, among other functions, preforms a job of accepting and evaluating teams' solutions (runs), and displaying results in a rank table. The scoring rules are as follows:

1. Each run is either accepted or rejected.
2. The problem is considered solved by the team, if one of the runs submitted for it is accepted.
3. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submission of the first accepted run for this problem (in minutes) plus 20 minutes for every other run for this problem before the accepted one. For an unsolved problem consumed time is not computed.
4. The total time is the sum of the time consumed for each problem solved.
5. Teams are ranked according to the number of solved problems. Teams that solve the same number of problems are ranked by the least total time.
6. While the time shown is in minutes, the actual time is measured to the precision of 1 second, and the the seconds are taken into account when ranking teams.
7. Teams with equal rank according to the above rules must be sorted by increasing team number.

Your task is, given the list of *N* runs with submission time and result of each run, compute the rank table for *C* teams.

### *Constraints*

1 ≤ *C, N* ≤ 1000, 1 ≤ *ci* ≤ *C*, 1 ≤ *pi* ≤ 20, 1 ≤ *ti* ≤ 36000.

### *Input file format*

Input contains integer numbers *C* *N*, followed by *N* quartets of integes *ci* *pi* *ti* *ri*, where *ci* -- team number, *pi* -- problem number, *ti* -- submission time *in seconds*, *ri* -- 1, if the run was accepted, 0 otherwise.

### *Output file format*

Output file must contain *C* integers -- team numbers sorted by rank.

#### *Sample input #1*

2 3

1 1 5000 0

1 1 500 1

2 1 10000 1

#### *Sample output #1*

1 2

#### *Sample input #2*

3 3

1 2 3000 0

1 2 3100 1

2 1 4200 1

#### *Sample output #2*

2 1 3