

Machine Cooling II

Duck is a busy guy who has N tasks to do in a single day, each task has four sub tasks. The i -th task is scheduled at the S_i second starting from 00:00:00, for example, 65 is 00:01:05 and 82800 is 23:00:00, and its j -th sub task requires $D_{i,j}$ seconds to complete. That is, the j -th sub task of i -th task ends at $S_i + D_{i,j}$ second. Duck plans to buy some machines to help him complete the tasks automatically. One machine can only deal with one task at the same time, but can deal with all four sub tasks simultaneously. Therefore, if any of the sub tasks of that task is not finished, the machine cannot deal with next task. If the whole task is finished, it can deal with the next task immediately without waiting.

However, Duck doesn't want that happens because he wants the machines to be more durable. He wants to maximize the cooling time of each machine whenever it completes a task. Given that the maximum cooling time of each machine is 86400, no matters at what second a machine completes a task, but the cooling time must be the same for all machines. What is the maximum cooling time that every machine can have if Duck uses as few machines as possible to complete all tasks by distributing it optimally?

Input

The first line is the number of test cases T . ($1 \leq T \leq 20$)

For each test case, it starts with the number of tasks N . ($1 \leq N \leq 100$)

Following N lines, each consisting of five integers: the second counting from 00:00:00 that the i -th task is scheduled at S_i , the seconds required to complete the j -th sub task D_1, D_2, D_3, D_4 . ($1 \leq S_i \leq 86399, 1 \leq D_{i,j} \leq 86400 - S_i$)

Output

Output the maximum cooling time that every machine can have.

Example

Input:

```
2
6
39999 7643 9987 13924 694
2000 3100 3804 2010 1999
4900 15238 28098 27777 27777
28813 11186 15742 886 20016
70000 200 300 400 500
51234 3555 30 7000 24567
3
52024 10000 7321 8864 20
62024 7321 10000 20 8864
72024 20 8864 10000 7321
```

Output:

```
2405
0
```

Explanation

In case 1, Duck needs two machines for task {2, 4, 6} and {1, 3, 5}, the answer is then 2405 choosing the cooling time of 2405 and 7322.

In case 2, one machine is enough but there is no gap between each task.