UVa Online Judge

11358 Faster Processing Feasibility

The era of technology is so hectic at times! Each & every single day, the crying needs for faster & faster processors are becoming more important than before with the continuously increasing complexity of applications & tasks. One of the solutions that the processor manufacturers are successfully employing nowadays is parallel processing. Scheduling of tasks is a prime concern while designing faster processors having several components working in parallel. We are currently building a new microprocessor that has P logical sub-processors in it. Each of the logical sub-processors can act as an individual processor & process one of the available tasks during a time slot. Note that, a logical sub-processor can process only a single task in one time slot & a task can not be processed by more than one processor during a time slot. Now, you are given the arrival time (the time when a task becomes available to the processors), processing time (the number of time slots necessary to complete processing the task) & deadline (the earliest time when the processing of this task is required to be completed) for T tasks. You have to figure out if it is possible to schedule the tasks using the available resources in such a way that all tasks are completed before their respective deadlines.

Let us be a bit more specific. From now on we shall assume, each time slot equals 1 micro-second. The span of the first time slot is 0 to 1 micro-second while the second time slot is 1 to 2 micro-second & so on.

For example, consider a multi-processor system with 2 logical sub-processors. You are needed to schedule 3 tasks A, B & C with the following data.

Task	*Arrival Time*	*Processing Time*	*Deadline*
A	0	2	2
B	0	3	4
C	1	2	3

It is possible to schedule these 3 tasks in the given system so that all the tasks meet their respective deadlines i.e. processing of task A, B & C are completed at (or before) time 2 micro-second, 4 micro-second and 3 micro-second respectively. Look at the following table for a possible schedule.

Time (micro-second)	*Available tasks*	*Assigned to processor -1*	*Assigned to processor -2*	*Completed tasks*	*Due tasks*
0	A,B	A	B	-	-
1	A,B,C	A	C	-	-
2	B,C	B	C	A	A
3	B	B	-	A,C	A,C
4	-	-	-	A,C,B	A,C,B

Input

The first line of the input is the number of test cases C $(1 \leq C \leq 50)$. C test cases follow. A test case begins with 2 integers P $(1 \leq P \leq 40)$ and T $(1 \leq T \leq 40)$, as described earlier. Each of the next T lines gives a task detail. A task detail consists of 3 integers — arrival time, A_i $(0 \leq A_i \leq 1000)$, processing time, R_i $(1 \leq R_i \leq 5000)$ and deadline, $D_i (A_i + R_i \leq D_i \leq 10000)$.

Output

For each test case, print '`FEASIBLE`' in a line if there is a possible schedule to meet all the deadlines. Otherwise, print '`NO WAY`'.

Sample Input

```
2
2 3
0 2 2
0 3 4
1 2 3
2 3
0 2 2
0 3 3
1 2 3
```

Sample Output

```
FEASIBLE
NO WAY
```