

E - Go Sightseeing

When you travel, you like to spend time sightseeing in as many cities as possible, but sometimes you might not be able to because you need to catch the bus to the next city. To maximize your travel enjoyment, you decide to write a program to optimize your schedule.

You begin at city 1 at time 0 and plan to travel to cities 2 to N in ascending order, visiting every city. There is a bus service from every city i to the next city $i + 1$. The i -th bus service runs on a schedule that is specified by 3 integers: S_i , F_i and D_i , the start time, frequency and ride duration. Formally, this means that there is a bus leaving from city i at all times $S_i + xF_i$, where x is an integer and $x \geq 0$, and the bus takes D_i time to reach city $i + 1$.

At each city between 1 and $N - 1$, inclusive, you can decide to spend at least T_s time sightseeing before waiting for the next bus, or you can immediately wait for the next bus. You cannot go sightseeing multiple times in the same city. You may assume that boarding and leaving buses takes no time. You must arrive at city N by time T_f at the latest. (Note that you cannot go sightseeing in city N , even if you arrive early. There's nothing to see there!)

What is the maximum number of cities you can go sightseeing in?

Input

The input starts with a line indicating the number of test cases. Each test case that follows begins with a line containing 3 integers, N , T_s and T_f , representing the number of cities, the minimum time required for sightseeing in any city, and the latest time you can arrive in city N . This is followed by $N - 1$ lines. On the i -th line, there are 3 integers, S_i , F_i and D_i , indicating the start time, frequency, and duration of buses travelling from city i to city $i + 1$.

Limits on input sizes: $2 \leq N \leq 16$, $1 \leq S_i \leq 5000$, $1 \leq F_i \leq 5000$, $1 \leq D_i \leq 5000$, $1 \leq T_s \leq 5000$, $1 \leq T_f \leq 5000$.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the maximum number of cities you can go sightseeing in such that you can still arrive at city N by time T_f at the latest. If it is impossible to arrive at city N by time T_f , output Case #x: IMPOSSIBLE.

Sample Input	Sample Output
4 4 3 12 3 2 1 6 2 2 1 3 2 3 2 30 1 2 27 3 2 1 4 1 11 2 1 2 4 1 5 8 2 2 5 10 5000 14 27 31 27 11 44 30 8 20 2000 4000 3	Case #1: 2 Case #2: 0 Case #3: IMPOSSIBLE Case #4: 4

In the first test case, you can go sightseeing in city 1, catching the bus leaving at time 3 and arriving at time 4. You can go sightseeing in city 2, leaving on the bus at time 8. When you arrive in city 3 at time 10 you immediately board the next bus and arrive in city 4 just in time at time 12.