

Meeting



SEASON 9 – SECOND ROUND

Three friends live in a country with N cities, connected by $N-1$ two-way roads. Naturally, any city can be reached by any other. Travelling through a road costs a certain amount of money (after all, fuel is not free). The three friends live in three different cities A , B and C . They want to meet up. They want to choose a city D (which could happen to be one A , B or C), so that the sum of the costs for each of their trips to that city is as low as possible. Write a program that answers several queries (A,B,C) and for each query outputs the optimal city D and how much the friends have to pay to get there.

Input

From the first line of the input file `meeting.in` the number of cities N is entered.

From the next $N-1$ lines F_i , T_i , C_i are entered - a road between cities F_i and T_i and its cost

Next the number of queries Q is entered.

For each of the queries on a separate line the towns of the three friends A , B and C are entered.

Output

In the output file `meeting.out` for each of the queries print two numbers D and S - the city with the lowest sum of the costs needed to get to it from A , B and C and the sum itself.

Constraints

$$1 \leq N \leq 100000$$

$$0 \leq F, T, A, B, C, D < N$$

$$1 \leq Q \leq 1000$$

Time limit: 1 second

Memory limit: 256 MB

Meeting

SEASON 9 – SECOND ROUND



Example

Input (meeting.in)	Output (meeting.out)	Explanation
8 0 1 10 0 2 13 2 5 27 2 3 5 2 4 8 3 6 4 3 7 1 2 3 4 5 1 0 7	2 40 0 29	For the first query: Going from city 3, 4 and 5 to city 2 costs 5, 8 and 27 respectively. The sum is therefore $5+8+27=40$, which is the minimum possible. For the second query: Going from cities 1, 0 and 7 to city 0 costs 10, 0 and 19 respectively.