Imagine you are a tourist in Warsaw and have booked a bus tour to see some amazing attraction just outside of town. The bus first drives around town for a while (a *long* while, since Warsaw is a big city) picking up people at their respective hotels. It then proceeds to the amazing attraction, and after a few hours goes back into the city, again driving to each hotel, this time to drop people off.

For some reason, whenever you do this, your hotel is always the first to be visited for pickup, and the last to be visited for dropoff, meaning that you have to suffer through two not-so-amazing sightseeing tours of all the local hotels. This is clearly not what you want to do (unless for some reason you are *really* into hotels), so let's fix it. We will develop some software to enable the sightseeing company to route its bus tours more fairly—though it may sometimes mean longer total distance for everyone, but fair is fair, right?

For this problem, there is a starting location (the sightseeing company headquarters), h hotels that need to be visited for pickups and dropoffs, and a destination location (the amazing attraction). We need to find a route that goes from the headquarters, through all the hotels, to the attraction, then back through all the hotels again (possibly in a different order), and finally back to the headquarters. In order to guarantee that none of the tourists (and, in particular, you) are forced to suffer through two full tours of the hotels, we require that every hotel that is visited among the first $\lfloor h/2 \rfloor$ hotels on the way to the attraction is also visited among the first $\lfloor h/2 \rfloor$ hotels on the way back. Subject to these restrictions, we would like to make the complete bus tour as short as possible. Note that these restrictions may force the bus to drive past a hotel without stopping there (this is not considered visiting) and then visit it later, as illustrated in the first sample input.

Input

The first line of each test case consists of two integers n and m satisfying $3 \leq n \leq 20$ and $2 \leq m$, where n is the number of locations (hotels, headquarters, attraction) and m is the number of pairs of locations between which the bus can travel.

The n different locations are numbered from 0 to $n-1$, where 0 is the headquarters, 1 through $n-2$ are the hotels, and $n-1$ is the attraction. Assume that there is at most one direct connection between any pair of locations and it is possible to travel from any location to any other location (but not necessarily directly).

Following the first line are m lines, each containing three integers u, v, and t such that $0 \leq u, v \leq n-1, u \neq v, 1 \leq t \leq 3600$, indicating that the bus can go directly between locations u and v in t seconds (in either direction).

Output

For each test case, display the case number and the time in seconds of the shortest possible tour.

Sample Input

```
5 4
0 1 10
1 2 20
2 3 30
3 4 40
4 6
0 1 1
0 2 1
0 3 1
1 2 1
1 3 1
2 3 1
```

Sample Output

```
Case 1: 300
Case 2: 6
```