Escape Route

In IOI Kingdom, they use *Byou* as the unit of time. A day in IOI Kingdom is divided into S units of time. The moment x Byous ($0 \le x < S$) after the beginning of a day is called time x. The IOI Kingdom consists of N cities, numbered from 0 to $N - 1$. There are M roads connecting cities, numbered from 0 to $M - 1$. You can travel from any city to any other city by passing through some of the roads. The road i ($0 \le i \le M - 1$) connects the city A_i and the city B_i bidirectionally. It takes L_i Byous to pass through the road i from an endpoint to the other endpoint. Every day, a strict security inspection is performed on the road i from time C_i to the end of the day.

JOI Group is one of the secret sects in IOI Kingdom. Since it is a secret sect, the members of JOI Group is strictly confidential. This means members of JOI Group should not encounter a strict security inspection performed every day. Therefore, if a member of JOI Group wants to pass through the road i, the member should depart from either the city A_i or the city B_i at time x and arrive at the other city at time $x + L_i$ for some $0 \le x \le C_i - L_i$. Since a strict security inspection is not performed in the cities, a member of JOI Group may be at in either the city A_i or the city B_i when a strict security inspection is performed on the road i.

There are Q members in JOI Group, numbered from 0 to $Q - 1$. The member j ($0 \le j \le Q - 1$) departs from the city U_j at time T_j on some day and starts traveling to the city V_j. Members are allowed to stay in cities on the way for a while. It may take multiple days for the member j to arrive at the city V_j.

Write a program which, given the information of the cities and the roads of IOI Kingdom, the information of strict security inspections, and the information of the members of JOI Group, calculates for each j ($0 \le j \le Q - 1$) the minimum amount of time required for the member j to travel from the city U_j to the city V_j.

Implementation Details

In order to speed up input and output, this task is graded with `grader`.

You need to submit one file. The name of the file you submit is `escape_route.cpp`. It should implement the following function. The program should include `escape_route.h` using the preprocessing directive `#include`.

- `std::vector<long long> calculate_necessary_time(`
  `int N, int M, long long S, int Q, std::vector<int> A, std::vector<int> B,`
  `std::vector<long long> L, std::vector<long long> C, std::vector<int> U,`
  `std::vector<int> V, std::vector<long long> T)`

  This function is called exactly once for each test case.

- The parameter `N` is the number of cities in IOI Kingdom.
- The parameter `M` is the number of roads in IOI Kingdom.
- The parameter `S` means a day in IOI Kingdom is S Byous.
- The parameter `Q` is the number of members of JOI Group.
- The parameters `A, B, L, C` are arrays of length M. They mean the road i ($0 \leq i \leq M - 1$) connects the city `A[i]` and the city `B[i]`, it takes `L[i]` Byous to pass through the road i, and a strict security inspection on the road i starts at time `C[i]`.
- The parameters `U, V, T` are arrays of length Q. They mean the member j ($0 \leq j \leq Q - 1$) departs from the city `U[j]` at time `T[j]` and starts traveling to the city `V[j]`.
- This function should return an array `answer` of `long long` type and of length Q. It means, for each $0 \leq j \leq Q - 1$, the minimum amount of time required for the member j to travel from the city `U[j]` to the city `V[j]` is `answer[j]` Byous.

Important Notices

- Your program must not use the standard input and the standard output. Your program must not communicate with other files by any methods. However, your program may output debugging information to the standard error.
- The grader does not check the format of input. If the format of input is incorrect, the behavior of the grader is not guaranteed.
- If the length of the array returned by `calculate_necessary_time` is different from Q or the array contains negative integers, the behavior of the grader is not guaranteed.

Compilation and Test Run

You can download an archive file from the contest webpage which contains the grader to test your program. The archive file also contains a sample source file of your program.

The grader is the file `grader.cpp`. In order to test your program, put `grader.cpp`, `escape_route.cpp`, `escape_route.h` in the same directory, and run the following command to compile your programs.

```
g++ -std=gnu++17 -O2 -fsigned-char -o grader grader.cpp escape_route.cpp
```

When the compilation succeeds, the executable file `grader` is generated.

In this task, the grader available from the contest webpage is the same as the actual grader used for grading in CMS.

Input for the Grader

The grader reads the following data from the standard input. Given values are all integers.

$N \ M \ S \ Q$

$A_0 \ B_0 \ L_0 \ C_0$

$\vdots$

$A_{M-1} \ B_{M-1} \ L_{M-1} \ C_{M-1}$

$U_0 \ V_0 \ T_0$

$\vdots$

$U_{Q-1} \ V_{Q-1} \ T_{Q-1}$

Output of the Grader

The grader outputs Q lines to the standard output. The $(k + 1)$-th line $(0 \leq k \leq Q - 1)$ of output contains `answer[k]`.

Constraints

- $2 \leq N \leq 90$.
- $N - 1 \leq M \leq \dfrac{N(N-1)}{2}$.
- $2 \leq S \leq 1\,000\,000\,000\,000\,000 = 10^{15}$.
- $1 \leq Q \leq 3\,000\,000$.
- $0 \leq A_i \leq N - 1 \ (0 \leq i \leq M - 1)$.
- $0 \leq B_i \leq N - 1 \ (0 \leq i \leq M - 1)$.
- $A_i \neq B_i \ (0 \leq i \leq M - 1)$.
- $(A_i, B_i) \neq (A_k, B_k), (A_i, B_i) \neq (B_k, A_k) \ (0 \leq i < k \leq M - 1)$.
- $1 \leq L_i < S \ (0 \leq i \leq M - 1)$.
- $L_i \leq C_i < S \ (0 \leq i \leq M - 1)$.
- You can travel from any city to any other city by passing through some of the roads.
- $0 \leq U_j \leq N - 1 \ (0 \leq j \leq Q - 1)$.
- $0 \leq V_j \leq N - 1 \ (0 \leq j \leq Q - 1)$.
- $U_j \neq V_j \ (0 \leq j \leq Q - 1)$.
- $0 \leq T_j < S \ (0 \leq j \leq Q - 1)$.

Subtasks

1. (5 points) $N \leq 40,\ \ Q \leq 1\,000$,
2. (20 points) $N \leq 40,\ \ U_j = 0\ (0 \leq j \leq Q - 1)$.
3. (10 points) $N \leq 40$.
4. (35 points) $N \leq 60$.
5. (30 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
4 5 20 6	3
0 1 3 19	8
0 2 2 8	14
1 2 4 15	2
1 3 5 14	5
2 3 1 18	7
0 3 5	
0 3 7	
0 3 9	
2 0 6	
3 1 10	
1 2 15	

This sample input satisfies the constraints of Subtasks 1, 3, 4, 5.

At time 5, the member 0 starts traveling from the city 0 to the city 3. If the member 0 travels in the following way, it takes 3 Byous.

- Using the road 1, the member 0 departs from the city 0 at time 5, and arrives at the city 2 at time 7.
- Using the road 4, the member 0 departs from the city 2 at time 7, and arrives at the city 3 at time 8.

Since this is the minimum value, we have `answer[0]` = 3.

At time 7, the member 1 starts traveling from the city 0 to the city 3. If the member 1 travels in the following way, it takes 8 Byous.

The 20th Japanese Olympiad in Informatics (JOI 2020/2021)
Spring Training Camp/Qualifying Trial
March 20–23, 2021 (Komaba, Tokyo)

Contest Day 2 – Escape Route

- Using the road 0, the member 1 departs from the city 0 at time 7, and arrives at the city 1 at time 10.
- Using the road 2, the member 1 departs from the city 1 at time 10, and arrives at the city 2 at time 14.
- Using the road 4, the member 1 departs from the city 2 at time 14, and arrives at the city 3 at time 15.

Since this is the minimum value, we have `answer[1]` = 8.

At time 9, the member 2 starts traveling from the city 0 to the city 3. If the member 2 travels in the following way, the member 2 arrives at the city 3 at time 3 on the next day. In total, it takes 14 Byous.

- The member 2 stays in the city 0 until time 0 on the next day.
- Using the road 1, the member 2 departs from the city 0 at time 0, and arrives at the city 2 at time 2.
- Using the road 4, the member 2 departs from the city 2 at time 2, and arrives at the city 3 at time 3.

Since this is the minimum value, we have `answer[2]` = 14.

Sample Input 2	Sample Output 2
6 10 100 9	42
5 3 4 29	32
1 0 6 26	4
0 4 2 7	93
0 5 18 18	99
2 0 79 82	6
3 4 35 46	102
1 2 15 57	60
2 4 3 6	39
4 1 21 83	
3 2 47 53	
0 2 63	
0 4 70	
0 4 98	
0 5 25	
0 5 19	
0 4 96	
0 5 2	
0 3 62	
0 3 83	

This sample input satisfies the constraints of all Subtasks.

Sample Input 3	Sample Output 3
8 12 1000000000000000 13	72937946261976
2 0 4451698272827 120985696255786	929038398222642
6 5 78520421713825 342652131468508	702857945988825
2 1 185377268405175 382583457603811	272921388674172
0 4 54350742205838 133614919589507	580895059624855
7 0 68486247989149 651590905094148	181808439529442
0 6 85177550834829 299184420663240	117602869946965
5 2 442329739732459 926608308293721	569788353034530
3 7 78020232822359 913548478810253	1181546234307589
1 3 267796317244889 687571310475622	244230056736534
5 4 90590208828121 910324397566584	513790925121797
5 7 8414633059584 17796117322043	617759130113052
4 6 45682367792138 204548471584556	674500988551485
7 2 44779065000162	
3 5 79376234836942	
4 7 305556687070759	
4 3 927935834343174	
5 1 663284649258985	
2 5 967584209777344	
5 2 963749709374595	
7 4 484562389171308	
1 5 446160773830045	
6 4 801452311055604	
3 1 744524289545354	
0 6 467418420721777	
5 6 371181379240653	

This sample input satisfies the constraints of Subtasks 1, 3, 4, 5.