

3. Jak szybko dojadę?

Nikt w niego nie wiedział. Z każdym kolejnym bankructwem ludzie się od niego odwracali. Przedłużające się terminy zniechęcały nawet jego największych zwolenników.

Oto jednak jest - system Polskich Kolei Próżniowych na miarę naszych czasów i możliwości! Kapsuły mkną między wszystkimi ważnymi miastami i pozwalają ludziom podróżować w czasie krótszym niż kiedykolwiek.

Aby zamknąć krytykom usta, zarząd spółki zlecił ci opracowanie aplikacji, która będzie odpowiadała na ważne pytanie "jak szybko dojadę z miasta A do miasta B"? Otrzymałeś wykaz wszystkich linii kolei próżniowych wraz z czasem przejazdu między kolejnymi przystankami dla każdej z nich. Twoim zadaniem jest opracować algorytm, który na podstawie tych danych możliwie szybko będzie obliczał czas podróży między dwoma dowolnymi stacjami systemu.

Zarząd spółki ma dla ciebie jeszcze jedną dobrą wiadomość - kapsuły kolei próżniowych poruszają się po liniach z tak wielką częstotliwością, że nie musisz w swoich obliczeniach uwzględniać czasu na przesiadkę między liniami.

1 Zadanie

Napisz program, który dla zadanego układu linii kolei próżniowych obliczy czas podróży między każdymi dwoma stacjami.

2 Wejście

Pierwsza linia zawiera trzy liczby całkowite T , N Q będące odpowiednio liczbą stacji w systemie, liczbą linii i liczbą zapytań od użytkowników aplikacji.

W kolejnych N wierszach znajdują się opisy poszczególnych linii w następującym formacie: pierwsza liczba oznacza liczbę stacji na trasie linii, a następnie występują na zmianę: numer stacji i czas podróży do następnej stacji, przy czym pierwsza liczba to numer stacji początkowej, a ostatnia to numer stacji

końcowej.

Następne Q wierszy zawiera zapytania od użytkowników. Każde z zapytań składa się z dwóch liczb - numerów stacji początkowej i końcowej.

W szablonie programu znajdziesz kilka miejsc do uzupełnienia.

3 Wyjście

Dla każdego zapytania wypisz czas podróży między stacją początkową i stacją końcową.

4 Przykład

4.1 Wejście

```
5 2 3
4 2 12 3 1 4 20 0
4 3 3 4 7 1 22 0
3 4
0 2
3 0
```

4.2 Wyjście

```
1
33
21
```