Zadanie: RAILWAY
Bajtockie Koleje Państwowe

dzień 1, plik źródłowy `railway.*`, dostępna pamięć 32 MB *CPSPC 2004, 7 czerwca*

Wiele lat temu, władze Bajtocji doszły do wniosku, że stworzenie sieci połączeń kolejowych ma szansę w istotny sposób wpłynąć na jakoś komunikacji publicznej w kraju. Ze względów finansowych zdecydowano się jednak na następujący "wariant minimum":

- Każdy pociąg łączy bezpośrednio dwa miasta i nie zatrzymuje się po drodze.
- Każdy pociąg jeździ w obie strony (tzn. pociąg z miasta A do miasta B jeździ także z miasta B do miasta A), a ceny obu połączeń są takie same.
- Z dowolnego miasta można dojechać do dowolnego innego miasta (może się to jednak wiązać z koniecznością przesiadek).
- Liczba pociągów jest możliwie najmniejsza.

Po dwóch latach wytężonej pracy sieć kolejową udało się uruchomić i podróżni mieli okazję poznać szybkość i solidność Bajtockich Kolei Państwowych. Pierwszy poważny problem dotyczył wydawania biletów: Aby obliczyć cenę biletu upoważniającego do podróży z 42 przesiadkami, kasjer musiał zsumować ceny biletów na wszystkie pociągi w niej występujące, co nie jest czynnością zbyt pasjonującą. Poza tym, oczywiście łatwo się w takich obliczeniach pomylić. Zostałeś zatrudniony przez BKP, aby napisać program obliczający ceny biletów.

Zadania

Napisz program, który:

- wczyta opis sieci połączeń kolejowych i ceny biletów na poszczególne pociągi,
- odpowie na ciąg pytań dotyczących cen biletów: każde pytanie składa się z dwóch numerów miast x_i, y_i, a odpowiedzią na takie pytanie jest cena biletu z x_i do y_i.

Wejście

Pierwszy wiersz pliku wejściowego `railway.in` zawiera dwie liczby całkowite N, Q (w tej kolejności), rozdzielone pojedynczym odstępem — N jest liczbą miast, $1 \leq N \leq 100\,000$, a Q liczbą pytań, $1 \leq Q \leq 2\,000\,000\,000$. Dla uproszczenia, miasta są ponumerowane liczbami $1, \ldots, N$. W kolejnych $N-1$ wierszach znajduje się opis połączeń kolejowych: każdy z tych wierszy zawiera trzy liczby całkowite a_i, b_i, c_i ($1 \leq a_i, b_i \leq N$, $0 \leq c_i \leq 2\,000\,000\,000$). Oznaczają one, że z miasta a_i jeździ pociąg do miasta b_i oraz, że cena biletu na ten pociąg wynosi c_i. Możesz założyć, że sieć połączeń jest poprawna, tzn. jest drzewem. W kolejnych Q wierszach znajdują sie zapytania: każdy z tych wierszy zawiera dwie liczby całkowite x_i, y_i ($1 \leq x_i, y_i \leq N$, $x_i \neq y_i$) – parę miast dla której chcemy znać cenę biletu.

Wyjście

Plik wyjściowy `railway.out` powinien składać się z dokładnie Q wierszy. i-ty wiersz powinien zawierać pojedynczą liczbę całkowitą — cenę biletu za przejazd z x_i do y_i. Możesz założyć, że wszystkie ceny biletów mieszczą się w 32-bitowej liczbie całkowitej ze znakiem. Znaczy to, że liczby te mieszczą się w typie `int` w C/C++ lub typie `longint` w Pascalu.

Przykład

Dla pliku wejściowego `railway.in`:

```
4 3
1 2 5
2 3 6
2 4 1
1 2
2 3
1 3
```

poprawnym wynikiem jest plik wyjściowy `railway.out`:

```
5
6
11
```