OJ 10308

Roads in the North

Prof. Edson Alves

Faculdade UnB Gama

Building and maintaining roads among communities in the far North is an expensive business. With this in mind, the roads are built in such a way that there is only one route from a village to a village that does not pass through some other village twice.

Given is an area in the far North comprising a number of villages and roads among them such that any village can be reached by road from any other village. Your job is to find the road distance between the two most remote villages in the area.

The area has up to $10,000$ *villages connected by road segments. The villages are numbered from* 1*.*

Construir e manter estradas entre comunidades no Norte é um negócio caro. Com isto em mente, as estradas foram construídas de tal modo que há uma única rota entre uma vila e outra e que não passa por uma mesma vila duas vezes.

É dada uma área no Norte composta por um certo número de vilas e estradas que as conectam, de modo que pode-se chegar a qualquer vila a partir de qualquer vila. Seu trabalho é encontrar a distância da rota entre as duas vilas mais remotas desta área.

A área tem no máximo 10.000 **vilas conectadas por estradas. As vilas são numeradas a partir de** 1**.**

Input

The input contains several sets of input. Each set of input is a sequence of lines, each containing three positive integers: the number of a village, the number of a different village, and the length of the road segment connecting the villages in kilometers. All road segments are two-way. Two consecutive sets are separated by a blank line.

Output

For each set of input, you are to output a single line containing a single integer: the road distance between the two most remote villages in the area.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste é formado por uma sequência de linhas, cada uma contendo três inteiros positivos: o número de uma vila, o número de uma vila diferente e o comprimento da estrada que conecta estas duas vilas, em kilômetros. Todas as estradas são bidirecionais. Dois casos de teste consecutivos são separados por uma linha em branco.

Saída

Para cada caso de teste, você deve imprimr uma única linha contendo um único inteiro: a distância da rota entre as duas vilas mais remotas da área.

Exemplo de entrada e saída

Exemplo de entrada e saída

```
5 1 6
```

Exemplo de entrada e saída

5 1 6

Exemplo de entrada e saída

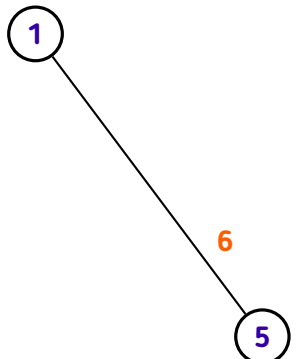
5 1 6

Exemplo de entrada e saída

```
5 1 6
```

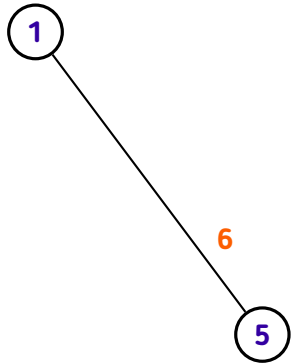
Exemplo de entrada e saída

```
5 1 6
```



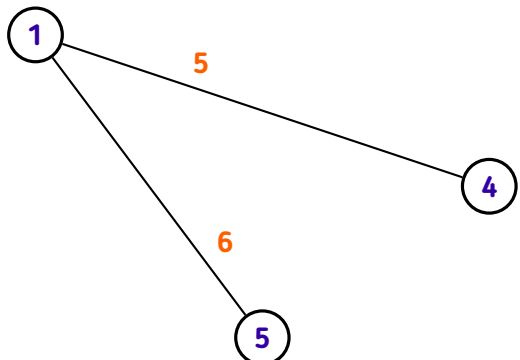
Exemplo de entrada e saída

```
5 1 6
1 4 5
```



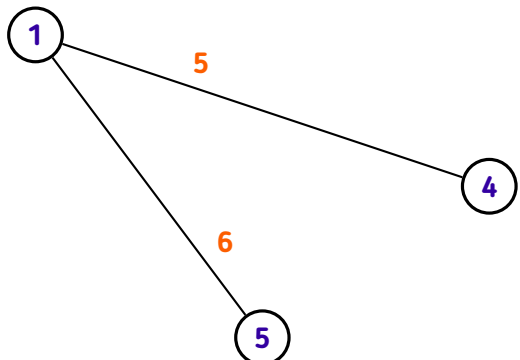
Exemplo de entrada e saída

```
5 1 6
1 4 5
```



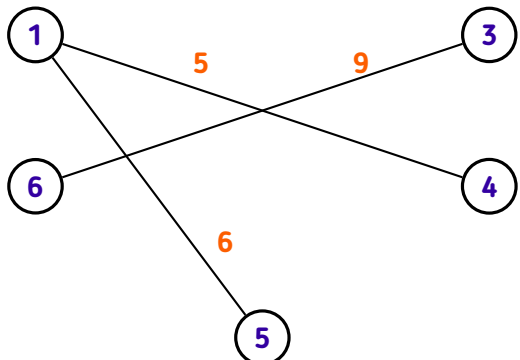
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
```



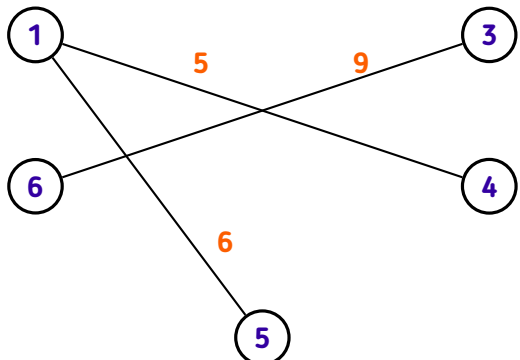
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
```



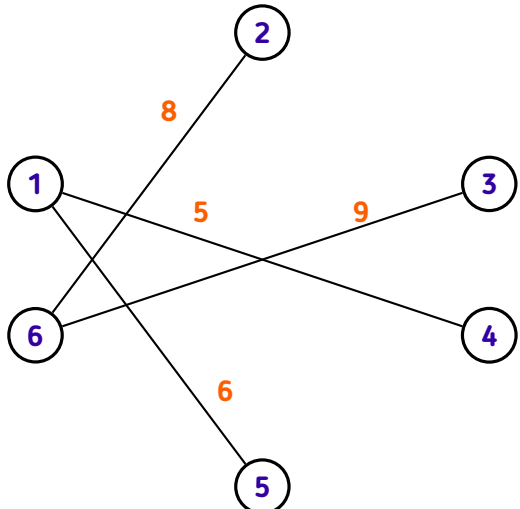
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
2 6 8
```



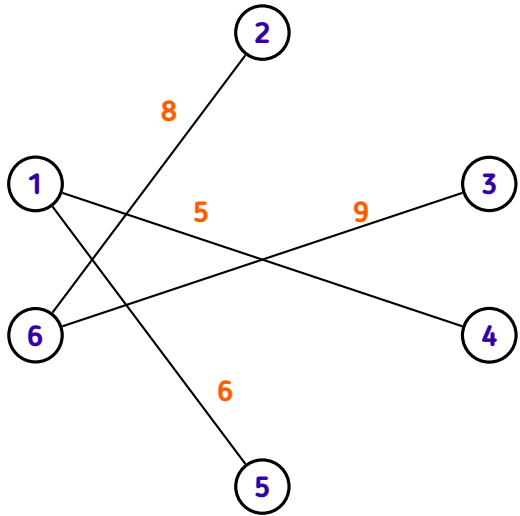
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
2 6 8
```



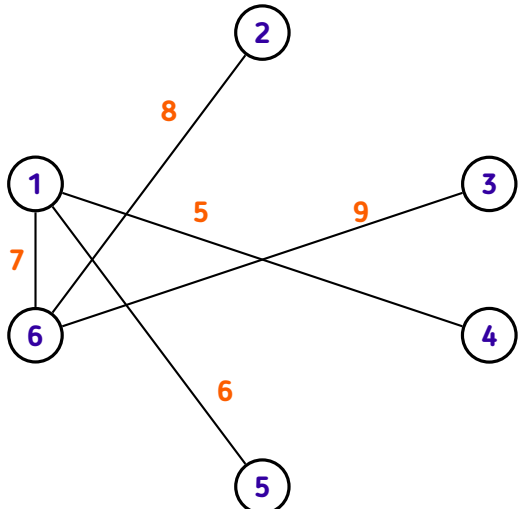
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
2 6 8
6 1 7
```



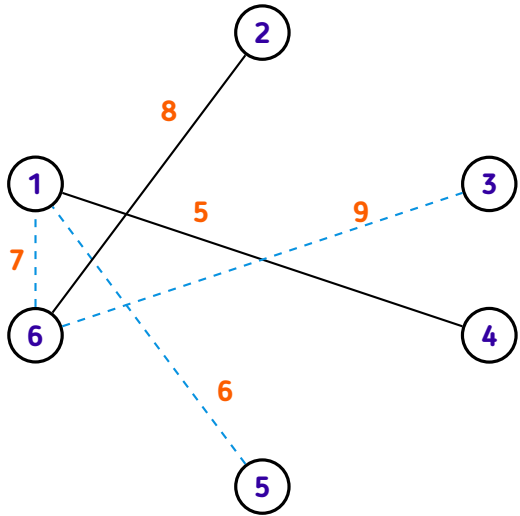
Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
2 6 8
6 1 7
```



Exemplo de entrada e saída

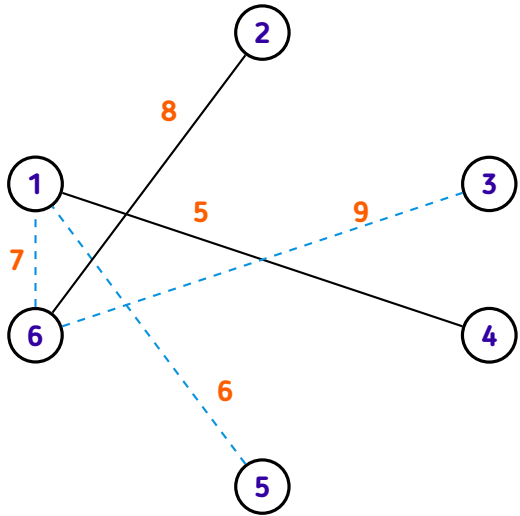
```
5 1 6
1 4 5
6 3 9
2 6 8
6 1 7
```



Exemplo de entrada e saída

```
5 1 6
1 4 5
6 3 9
2 6 8
6 1 7
```

6 + 7 + 9 = 22



Solução

Solução

- **As vilas mais remotas são as mais distantes entre si**

Solução

- **As vilas mais remotas são as mais distantes entre si**
- **Neste problema, as vilas são os vértices e as estradas são arestas**

Solução

- As vilas mais remotas são as mais distantes entre si
- Neste problema, as vilas são os vértices e as estradas são arestas
- As características dadas no texto tornam o grafo uma árvore

Solução

- ⋆ **As vilas mais remotas são as mais distantes entre si**
- ⋆ **Neste problema, as vilas são os vértices e as estradas são arestas**
- ⋆ **As características dadas no texto tornam o grafo uma árvore**
- ⋆ **Atenção: Como o grafo é ponderado, o último vértice visitado na BFS não é, necessariamente, o mais distante da origem**

```
ii bfs(int s, int N)
{
    vector<int> dist(N + 1, oo); dist[s] = 0;
    queue<int> q; q.push(s);

    while (not q.empty()) {
        auto u = q.front(); q.pop();

        for (auto [v, w] : adj[u]) {
            if (dist[v] == oo) {
                dist[v] = dist[u] + w;
                q.push(v);
            }
        }
    }

    auto w = (int) (max_element(dist.begin() + 1, dist.end()) - dist.begin());

    return { w, dist[w] };
}
```

```
int diameter(int N)
{
    auto [v, _] = bfs(1, N);
    auto [w, D] = bfs(v, N);

    return D;
}

int solve(int N)
{
    auto D = diameter(N);

    return D;
}
```