

In the history of Asia there is an ancient empire called Mayo which was one of the most powerful countries of its own time. At first the empire consisted of only one city (which was the capital) but its people were great warriors so they started to invade their neighbors and increase the area of their lands.

Each time they defeated a neighbor, they razed all their cities except the biggest one, which is added to their kingdom. They built a road from that city to one of their own cities, so that they could travel between all cities of their lands. If the added city was big enough it would become the capital of Mayo.

The vulnerability of a city in Mayo was defined the number of roads that one had to pass to travel from that city to the capital of Mayo. We have a list of all kingdoms that people of Mayo invaded and the description of the roads that they have built.

We want to know the maximum vulnerability of the cities, after each city was added to Mayo. To simplify the output of the program, report the sum of $v_1 + v_2 + \dots + v_n$, where v_i is the maximum vulnerability of the cities after adding the i -th city.

Input

There are several test case in the input. The first line of each test case contains n ($1 \leq n \leq 100,000$) which is the number of cities in Mayo. For each test case, the cities are numbered from 1 to n ordered by the time they had been added to Mayo. So, the city 1 is the first city of Mayo (and its initial capital). Each of next $n - 1$ lines of the test case contains two integers j and c . This means that when the city $i + 1$ is added to Mayo, a road is built between the city $i + 1$ and the city j . The number c is non-zero if the new city (city $i + 1$) had become the new capital of Mayo at the time of its join to Mayo, and is zero otherwise. The last line of the input contains a single '0'.

Output

For each test case, output a single line containing the sum of maximum vulnerabilities as explained in the problem description.

Sample Input

```
3
1 1
2 1
3
1 1
2 0
0
```

Sample Output

```
3
2
```