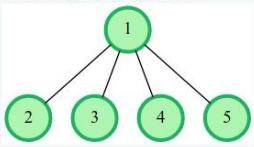
1. Hotel Construction

There are a certain number of cities in a country, some of which are connected with bidirectional roads. The number of roads is one less the number of cities, and it is possible to travel between any pair of cities using the roads. The distance between cities is the minimum number of roads one has to cross when traveling between them. How many ways are there to build exactly 3 hotels, each in a different city, such that the distance between every pair of hotels is equal?

For example, let's say there are *n = 5* cities, and *roads = [[1, 2], [1, 3], [1, 4], [1, 5]]*. This means that city 1 is connected with roads to all other cities, as seen below:



There are 4 ways to build exactly 3 hotels, each in a different city, so that the distance between every pair of hotels is equal:

1. Build hotels in cities 2, 3, and 4.
2. Build hotels in cities 2, 3, and 5.
3. Build hotels in cities 2, 4, and 5.
4. Build hotels in cities 3, 4, and 5.

In all these cases, the distance between every pair of hotels is 2. Because there are 4 ways to accomplish this, the answer is 4.

Function Description
Complete the function *numberOfWays* in the editor below. The function must return an integer denoting the number of ways to build 3 hotels in such a way that the distance between every pair of hotels is equal.

numberOfWays has the following parameter:
int *roads[n-1][2]*: **a 2-dimensional array of integers, 0-indexed,** such that *roads[i][0]* and *roads[i][1]* denote *cities that are connected by* the i^{th} road
Returns:
int: the number of ways to build 3 hotels in such a way that the distance between every pair of hotels is equal

Constraints

- $4 \le n \le 50$
- $1 \le roads[i][0] \le n$
- $1 \le roads[i][1] \le n$
- $roads[i][0] \ne roads[i][1]$

▼ Input Format For Custom Testing

In the first line, there is a single integer, *n - 1*, denoting the number of roads.
In the second line, there is a single integer, 2.
Then, *n -1* lines follow. The i^{th} of them contains two space-separated integers, *roads[i][0]* and *roads[i][1]*, denoting the cities that are connected by roads.

▼ Sample Case 0

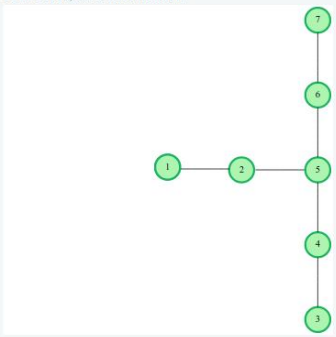
Sample Input For Custom Testing

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

Sample Output

```
2
```

Explanation
In this case, there are n = 7 cities, *n - 1= 6* and *roads = [[1, 2], [2, 5], [3, 4], [4, 5], [5, 6], [7,6]]*. This means there are 7 cities connected, as shown below:



There are 2 ways to build 3 hotels under the conditions:

1. Build hotels in cities 2, 4, and 6. The distance between any pair of hotels is 2 in this case.
2. Build hotels in cities 1, 3, and 7. The distance between any pair of hotels is 3 in this case.

▼ Sample Case 1

Sample Input For Custom Testing

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

Sample Output

```
0
```

Explanation
In this case, *n = 5* and *roads = [[1, 2], [2, 3], [3, 4], [4, 5]]*. This means there are 5 cities connected to form a single path, as shown below.

In this case, it is impossible to build 3 hotels under the conditions. Therefore, the answer is 0.