**Problem 3. Paths**

Ivancho really likes travelling. However, he doesn’t like every travel, but only the travels with duration of exactly K hours. This is the reason why his mom let him chose the path for the following travel – from their city to the city where lives his grandma. The map that Ivancho has contains N cities and M two-way roads. The time for going through each road is exactly one hour.

Now Ivancho is wondering in how many different ways he can chose a path for his travel so that he would arrive to his grandma’s city exactly K hours after he leave his city. Ivancho can go through one city or road more than once.

Your task is to write a programme **paths** that reads information about the map and calculates the number of the different paths that Ivancho would chose. As this number could be quite large, you should calculate it module 106+1. Two paths are considered to be different if it does exist a moment after going that Ivancho would be in different cities according to the paths. There is no road that starts and ends in same city and there is not more than one road between any two cities.

**Input**: The first line of the input file **paths.in** contains two integers N and M – the number of cities and the number of roads. Follow M lines containing two integers – the numbers of the cities that connect the i-th road. In the end of the file there are three integers – the number of the city where is Ivancho, the number of the city where is his grandma and K.

**Output**: The output file **paths.out** must contain a single integer – the number of the paths by module 106+1.

**Constraints:**

1N1000

1M100000

1 K100

**TIME LIMIT – 2 sec**

**Note**:You can scan from a file and print into a file by using an operator for that. You can use operator freopen including fstream library and adding the following two lines in the beginning of your main function:

freopen ( "paths.in", "r", stdin );

freopen ( “paths.out”, “w” , stdout );

**Example:**

|  |  |
| --- | --- |
| **paths.in** | **paths.out** |
| 3 3  1 2  3 1  2 3  1 3 4 | 5 |