D. Fight Against Traffic

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Little town Nsk consists of n junctions connected by m bidirectional roads. Each road connects two distinct junctions and no two roads connect the same pair of junctions. It is possible to get from any junction to any other junction by these roads. The distance between two junctions is equal to the minimum possible number of roads on a path between them.

In order to improve the transportation system, the city council asks mayor to build one new road. The problem is that the mayor has just bought a wonderful new car and he really enjoys a ride from his home, located near junction s to work located near junction t. Thus, he wants to build a new road in such a way that the distance between these two junctions won't decrease.

You are assigned a task to compute the number of pairs of junctions that are not connected by the road, such that if the new road between these two junctions is built the distance between s and t won't decrease.

Input

The firt line of the input contains integers n, m, s and t ($2 \leq n \leq 1000$, $1 \leq m \leq 1000$, $1 \leq s, t \leq n$, $s \neq t$) — the number of junctions and the number of roads in Nsk, as well as the indices of junctions where mayors home and work are located respectively. The i-th of the following m lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), meaning that this road connects junctions u_i and v_i directly. It is guaranteed that there is a path between any two junctions and no two roads connect the same pair of junctions.

Output

Print one integer — the number of pairs of junctions not connected by a direct road, such that building a road between these two junctions won't decrease the distance between junctions s and t.

Examples

input

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

output

```
0
```

input

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

output

```
5
```

input

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

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

output

```
3
```