

# Corey's Party

**Input File:** *partyin.txt*

**Output File:** *partyout.txt*

**Time Limit:** 1 second

Your friend Corey wants to organise a party and invite a few mates.

In the past, Corey's parties haven't always ended well. Some guests get bored because they don't know anyone, then start causing trouble. Other guests get bored because they already know everyone, then start causing trouble.

Thus, Corey's new party invitation strategy is to invite a set of people to the party such that every guest already knows at least  $A$  other guests (not including Corey), and **doesn't** know at least  $B$  other guests. This way, everyone will have at least  $A$  friends to chat to and at least  $B$  new people to meet.

Corey will give you a list of the  $M$  existing friendships between his  $N$  friends. You must write a program to output the largest number of friends he can invite to his party satisfying the above constraints.

## Input

Your program should read from the file `partyin.txt`. The first line of input will contain four space-separated integers in this order:

- $N$ : the number of Corey's friends
- $M$ : the number of friendships between Corey's friends
- $A$ : the number of guests that each guest must already know
- $B$ : the number of guests that each guest must not already know

The following  $M$  lines of input will each contain two space-separated integers  $x$  and  $y$ , representing that Corey's friend  $x$  and Corey's friend  $y$  are already friends with each other. Friends are represented as integers between 1 and  $N$ .

## Output

Your program should write to the file `partyout.txt`. Your output file should contain one line with one integer: the maximum number of friends Corey can invite to his party such that every invited guest knows at least  $A$  other guests and doesn't know at least  $B$  other guests. Note that it may not always be possible to invite a set of guests that satisfies these constraints, in which case you should output 0.

**Sample Input 1**

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

**Sample Output 1**

```
4
```

**Sample Input 2**

```
7 8 2 0
1 2
4 5
1 6
2 6
3 6
4 6
5 6
7 1
```

**Sample Output 2**

```
5
```

**Explanation**

In the first sample case, if Corey invites friends 2, 3, 4 and 5 to a party, each will already know exactly 2 other guests and not know exactly 1 other guest. This satisfies Corey's constraint that every guest must know at least 2 other guests and not know at least 1 other guest. It is not possible to organise a party that includes friends 1 or 6, thus 4 is the maximum number of guests Corey can invite.

In the second sample case, Corey can invite friends 1, 2, 4, 5 and 6. Each guest would already know at least two other guests, thus satisfying  $A = 2$ . Since  $B = 0$ , there is no constraint on the number of guests each invited guest *doesn't* already know.

**Constraints**

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $1 \leq N \leq 1000$
- $1 \leq M \leq 100,000$
- $0 \leq A, B \leq N - 1$

Furthermore, for 50% of available marks,  $B = 0$ . That is, there is no constraint on the number of people an invited guest must not already know.