Problem G

Telecommunication Partners

Source file: partners.pas, partners.c, partners.cc or partners.java

ICPC, an international telecommunication company, wants to improve its relationship with business subscribers, offering a discount on calls made to a fixed set of telephone numbers selected by the client company. To help ICPC decide on the cost for this new service, they searched their database and produced a list of calls made last year by one company to another. If a company communicated with another company (making or receiving a call) during last year, we will say they are *Business Partners*.

You have been hired by ICPC to process the list of calls from last year and determine the size (in number of companies) of the largest set of companies that are Business Partners of at least K other companies in the same set. That is, you must find a set S of companies such that every company in S has at least K business partners that are also in S (and possibly partners that are outside S), where K is a parameter chosen by the ICPC.

Input

Your program should process several test cases. The first line of a test case contains three integers N, P and K. N represents the total number of companies subscribed to ICPC ($1 \leq N \leq 1000$); companies are identified by numbers between 1 and N. P represents the total number of business partner pairs, produced from last year calls; and K is the minimum number of business partners a company must have in the final set ($1 \leq K \leq N{-}1$), as described above. The next P lines describe each a business partner pair, represented as two integers X and Y, where X and Y are companies ($1 \leq X \leq N$, $1 \leq Y \leq N$ and $X \neq Y$). The value $N = 0$ indicates the end of input.

The input must be read from standard input.

Output

For each test case from the input, your program should print a single line, containing the size of the largest set of companies found by your program.

The output must be written to standard output.

Sample input

Output for the sample input

```
5 3 1
1 2
4 3
4 5
5 3 2
1 2
4 3
4 5
10 11 2
1 2
1 3
3 2
3 5
5 4
5 6
9 10
8 9
8 7
6 7
6 8
0 0 0
```

```
3
0
7
```