

Blockade

There are exactly N towns in Byteotia. Some towns are connected by bidirectional roads. There are no crossroads outside towns, though there may be bridges, tunnels and flyovers. Each pair of towns may be connected by at most one direct road. One can get from any town to any other directly or indirectly.

Each town has exactly one citizen. For that reason the citizens suffer from loneliness. It turns out that each citizen would like to pay a visit to every other citizen (in his host's hometown), and do it exactly once. So exactly $N*(N-1)$ visits should take place. That's right, *should*. Unfortunately, a general strike of programmers, who demand an emergency purchase of software, is under way. As an act of protest, the programmers plan to block one town of Byteotia, preventing entering it, leaving it, and even passing through. As we speak, they are debating which town to choose so that the consequences are most severe.

Task

Write a programme that:

- reads the Byteotian road system's description from the standard input,
- for each town determines, how many visits could take place if this town were not blocked by programmers,
- writes out the outcome to the standard output.

Input

In the first line of the standard input there are two positive integers: N and M ($1 \leq N \leq 100000$, $1 \leq M \leq 500000$) denoting the number of towns and roads, respectively. The towns are numbered from 1 to N . The following M lines contain descriptions of the roads. Each line contains two integers X_i and Y_i ($1 \leq X_i, Y_i \leq N$) and denotes a direct road between towns numbered X_i and Y_i .

Output

Your programme should write out exactly N integers to the standard output, one number per line. The i -th line should contain the number of visits that could not take place if the programmers blocked the town no. i .

Example

For the input data:

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

1 3
3 4
4 5

the correct result is:

8
8
16
14
8