Zadanie: BLO
Blokada

Etap II. Dzień próbny. Plik źródłowy `blo.*` *6–02–2008*

Dostępna pamięć: 32 MB.

Na obszarze Bajtocji znajduje się dokładnie n miast. Między niektórymi parami miast biegną dwukierunkowe drogi. Poza miastami nie ma skrzyżowań, lecz mogą istnieć mosty, tunele i estakady. Między każdymi dwoma miastami może istnieć co najwyżej jedna droga. Z każdego miasta da się dojechać — bezpośrednio bądź też pośrednio — do każdego innego.

W każdym mieście mieszka dokładnie jeden mieszkaniec. Mieszkańcom doskwiera samotność. Okazuje się, że każdy z mieszkańców chce odwiedzić każdego innego w jego mieście, i to dokładnie raz. A zatem powinno odbyć się $n \cdot (n-1)$ spotkań. Tak, tak, *powinno*. Niestety w Bajtocji trwają protesty programistów, domagających się wprowadzenia interwencyjnego skupu oprogramowania. Programiści planują, w ramach protestu, zablokowanie możliwości wjeżdżania, wyjeżdżania i przejeżdżania przez jedno z miast Bajtocji. Zastanawiają się teraz, które miasto wybrać tak, aby jak najbardziej uprzykrzyć życie mieszkańcom Bajtocji.

Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia opis sieci dróg Bajtocji,
- dla każdego miasta obliczy, ile spotkań nie mogłoby się odbyć, gdyby owe miasto zostało zablokowane przez programistów,
- wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby naturalne oddzielone pojedynczym odstępem: n oraz m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 500\,000$) oznaczające odpowiednio liczbę miast oraz liczbę dróg. Miasta są ponumerowane od 1 do n. W kolejnych m wierszach znajdują się opisy dróg. Każdy opis składa się z dwóch liczb a oraz b ($1 \leq a < b \leq n$) oddzielonych pojedynczym odstępem i oznacza, że istnieje droga między miastami o numerach a i b.

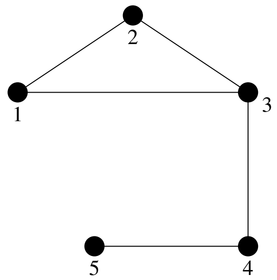
Wyjście

Na standardowe wyjście Twój program powinien wypisać dokładnie n liczb naturalnych, po jednej w wierszu. W i-tym wierszu powinna znaleźć się liczba spotkań, które nie odbyłyby się, gdyby programiści zablokowali miasto nr i.

Przykład

Dla danych wejściowych:

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



poprawnym wynikiem jest:

```
8
8
16
14
8
```