

Gdzie jest korzeń?

To jest zadanie interaktywne.

Dane jest drzewo o n wierzchołkach. (Drzewo to taki graf, w którym między każdą parą wierzchołków jest dokładnie jedna ścieżka). **Gwarantowane jest, że co najmniej jeden wierzchołek ma stopień co najmniej 3, czyli ma co najmniej trzech sąsiadów.** Jeden z wierzchołków jest *korzeniem* drzewa, ale nie wiadomo który - Twoim zadaniem jest znalezienie tego wierzchołka.

W tym zadaniu zadajesz pytania postaci:

- Dla danego zbioru wierzchołków a_1, a_2, \dots, a_m podaj, czy ich wspólny przodek (LCA) znajduje się w tym zbiorze.

Najniższy wspólny przodek (lowest common ancestor, LCA) zbioru wierzchołków S to najniżej położony (czyli najdalszy od korzenia) wierzchołek v taki, że każda ścieżka od korzenia drzewa do któregośkolwiek wierzchołka z S musi przejść przez v . Za pomocą pytań takich jak powyżej musisz ustalić, który wierzchołek jest korzeniem drzewa.

Interakcja

Twój program powinien rozpocząć interakcję przez wczytanie liczby n ($4 \leq n \leq 500$) - liczby wierzchołków drzewa.

Potem do wczytania jest $n - 1$ wierszy. Wiersz i -ty będzie zawierał dwie liczby a_i, b_i ($1 \leq a_i, b_i \leq n$), oznaczające krawędź w drzewie między wierzchołkami a_i oraz b_i . Jest gwarantowane, że te $n - 1$ krawędzi tworzą drzewo, a przynajmniej jeden wierzchołek ma stopień 3.

Aby zadać pytanie, wypisz na wyjście znak zapytania "?", potem liczbę całkowitą m , a po niej m różnych wierzchołków a_1, a_2, \dots, a_m ($1 \leq m \leq n$, $1 \leq a_i \leq n$, wszystkie a_i są różne) - wierzchołki, dla których dokonujesz sprawdzenia, czy jest wśród nich LCA tego zbioru.

Jako odpowiedź zostanie wypisane "YES" jeśli LCA jest jednym z wierzchołków a_1, a_2, \dots, a_m , zaś "NO" w przeciwnym razie.

Możesz zadać co najwyżej 1000 takich pytań. Wypisanie odpowiedzi nie liczy się do tego limitu. Aby podać odpowiedź, wypisz wiersz zawierający symbol "!", a po nim jedną liczbę v ($1 \leq v \leq n$) -

numer korzenia. Potem zakończ program.

Każde pytanie musi kończyć się znakiem końca wiersza. Musisz też po każdym pytaniu dokonać tzw. wyczyszczenia bufora wyjścia, co robi się używając instrukcji:

- `fflush(stdout)` lub `cout.flush()` w C++;
- `stdout.flush()` w Pythonie;

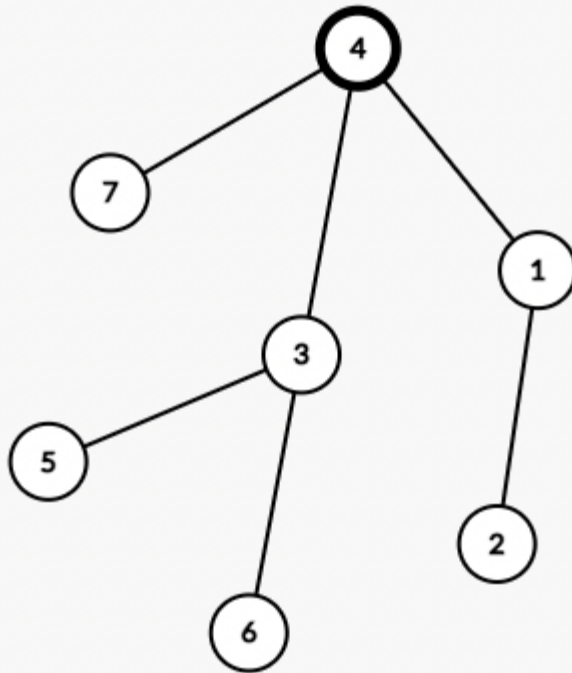
Jest gwarantowane, że struktura drzewa i korzeń są ustalone przed rozpoczęciem interakcji (*sprawdzarka nie jest adaptacyjna*).

Jeszcze raz, **nie zapomnij opróżnić bufora wyjścia** po każdym pytaniu.

Przykład

```
Wejście:
7
4 1
1 2
4 3
3 5
3 6
4 7
Wyjście:
? 2 5 6
Wejście:
NO
Wyjście:
? 3 6 3 5
Wejście:
YES
Wyjście:
? 2 1 7
Wejście:
NO
Wyjście:
? 2 4 6
Wejście:
YES
Wyjście:
! 4
```

Wyjaśnienie



Korzeniem (na razie nieznanym) jest wierzchołek 4.

W pierwszym pytaniu, LCA wierzchołków 5 i 6 to wierzchołek 3, który nie jest wśród wierzchołków 5, 6, a więc odpowiedź jest "NO".

W drugim pytaniu, LCA wierzchołków 3, 5 i 6 to wierzchołek 3, więc odpowiedź jest "YES".

W trzecim pytaniu, LCA wierzchołków 1 i 7 to 4 - odpowiedź "NO".

W czwartym pytaniu, LCA wierzchołków 4 i 6 to 4, a więc odpowiedź to "YES".

Na podstawie tych odpowiedzi zgadujemy, że korzeń to 4, co jest prawidłową odpowiedzią.

Punktacja

1. (7 punktów): $n \leq 9$
2. (10 punktów): $n \leq 30$
3. (max. 83 punktów): $n \leq 500$

W pierwszym i drugim podzadaniu możesz zadać co najwyżej 1000 pytań.

W trzecim podzadaniu, niech k to maksymalna liczba pytań, którą zadał Twój program na jakimkolwiek teście z tego podzadania. Jeżeli $k \leq 9$, otrzymasz 83 punkty. W przeciwnym razie, otrzymasz $\lfloor \max(10, 83 \cdot (1 - \frac{\ln(k-6)}{7})) \rfloor$ punktów.

Kod w C++ do obliczania liczby punktów za trzecie podzadanie:

```
((k <= 9) ? 83 : max(10, int(83 * (1 - log(k - 6.0) / 7))))
```