

Rury (Pipes)

W Bajtogradzie grasuje znany złoczyńca Bitold. Aby uprzykrzyć życie mieszkańcom, Bitold postanowił pomajstrować przy kanalizacji. Woda przeznaczona na użytek miasta jest przechowywana w N zbiornikach połączonych układem M rur. Z każdego zbiornika do każdego innego istnieje ścieżka złożona z jednej lub większej liczby rur. Każda z rur łączy dwa różne zbiorniki, a każda para zbiorników jest połączona co najwyżej jedną rurą.

Bitold podpiał się do niektórych rur i zaczął odciągać z nich wodę. Nie wiedząc czemu, z każdej z tych rur odciąga zawsze parzystą liczbę metrów sześciennych wody na sekundę (m^3/s). Jeśli Bitold odciąga $2d$ m^3/s wody z rury łączącej zbiorniki u oraz v , to w efekcie każdy ze zbiorników u, v traci d m^3/s wody.

Celem Bitolda nie jest jednak wzbogacenie się w wodę, tylko mydlenie ludziom oczu. Dlatego do niektórych kontrolowanych przez siebie rur Bitold pompuje wodę, zamiast ją odciągać. Do każdej z tych rur Bitold pompuje parzystą liczbę m^3/s wody. Jeśli Bitold pompuje $2p$ m^3/s wody do rury łączącej zbiorniki u oraz v , to w efekcie każdy ze zbiorników u, v zyskuje p m^3/s wody. Łączna zmiana stanu wody w zbiorniku jest sumą zysków i strat pochodzących ze wszystkich rur dochodzących do tego zbiornika. Tak więc jeśli do zbiornika są podłączone rury, z których jest odpompowywane, odpowiednio, $2d_1, 2d_2, \dots, 2d_a$ m^3/s wody oraz rury, do których jest pompowane, odpowiednio, $2p_1, 2p_2, \dots, 2p_b$ m^3/s wody, to zmiana stanu wody w tym zbiorniku jest równa $p_1 + p_2 + \dots + p_b - d_1 - d_2 - \dots - d_a$.

Burmistrz Bajtogradu, Bajtazar, zainstalował w kanalizacji system czujników. Niestety, czujniki znajdują się w zbiornikach, ale nie w rurach. Tak więc Bajtazar może wyznaczyć zmianę stanu wody w każdym zbiorniku, jednak nie wie, ile wody jest odciągane z bądź pompowane do poszczególnych rur.

Twoim zadaniem jest napisanie programu, który pomoże Bajtazarowi rozgryźć plan Bitolda. Znając układ sieci zbiorników oraz zmiany stanu wody w poszczególnych zbiornikach, Twój program powinien stwierdzić, czy na tej podstawie da się jednoznacznie odtworzyć plan Bitolda. Powiemy, że plan można odtworzyć jednoznacznie, jeśli można jednoznacznie ustalić, ile wody jest odciągane z i pompowane do poszczególnych rur. Zauważ, że objętość wody odciąganej lub pompowanej nie musi być taka sama dla wszystkich rur. Jeśli plan można odtworzyć jednoznacznie, Twój program powinien go wypisać.

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite: liczbę zbiorników N oraz liczbę rur M . Każdy z kolejnych N wierszy zawiera jedną liczbę całkowitą c_i : zmianę stanu wody w zbiorniku numer i ($1 \leq i \leq N$). Każdy z kolejnych M wierszy zawiera dwie liczby całkowite u_i oraz v_i ($1 \leq i \leq M$). Oznaczają one, że zbiorniki u_i oraz v_i ($1 \leq u_i, v_i \leq N$) są połączone rurą.

Możesz założyć, że podany na wejściu opis zawsze odpowiada jakiemuś planowi Bitolda.

Wyjście

Jeśli nie da się jednoznacznie odtworzyć planu Bitolda, Twój program powinien wypisać jeden wiersz zawierający liczbę 0. W przeciwnym razie Twój program powinien wypisać M wierszy zawierających po jednej liczbie x_i ($1 \leq i \leq M$). Jeśli Bitold odciąga d_i m^3/s wody z rury łączącej zbiorniki u_i oraz

v_i , powinno zachodzić $x_i = -d_i$. Jeśli Bitold pompuje p_i m^3/s wody do rury łączącej zbiorniki u_i oraz v_i , powinno zachodzić $x_i = p_i$. Jeśli zaś Bitold nie zaburza przepływu w rurze łączącej zbiorniki u_i oraz v_i , powinno zachodzić $x_i = 0$.

Ograniczenia

$$1 \leq N \leq 100\,000$$

$$1 \leq M \leq 500\,000$$

$$-10^9 \leq c_i \leq 10^9$$

Jeśli jednoznaczne odtworzenie planu Bitolda jest możliwe, $-10^9 \leq x_i \leq 10^9$.

W przypadkach testowych wartych 30 punktów, sieć kanalizacyjna jest drzewem.

Przykłady

Wejście	Wyjście
4 3 -1 1 -3 1 1 2 1 3 1 4	2 -6 2
4 5 1 2 1 2 1 2 2 3 3 4 4 1 1 3	0

Ograniczenia

Limit czasu: 1 sek na test

Limit pamięci: 128 MB na test