



Задача F. РАЗРЯЗВАНЕ НА ГРАФ

Даден е неориентиран граф без тегла, без примки и кратни ребра. За него се изпълняват два вида заявки:

- *cut* – разрязване на графа, т.е. премахване на едно ребро от него;
- *ask* – проверка дали два върха в графа се намират в една и съща компонента на свързаност.

Знае се, че след изпълнение на всички операции *cut*, няма да останат ребра в графа. Напишете програма, която определя резултата от всяка заявка *ask*.

Вход

Входът се състои от няколко набора входни данни. За всеки набор входните данни са в следния формат: На първия ред са дадени три цели числа n ($1 \leq n \leq 50000$), m ($1 \leq m \leq 100000$) и k ($1 \leq k \leq 150000$) – броят на върховете, броят на ребрата и броят на заявките. Следват m реда, задаващи ребрата на графа. Всеки от тези редове съдържа две цели числа u_i и v_i ($1 \leq u_i, v_i \leq n$) задаващи поредното ребро. Върховете са номерирани с целите числа от 1 до n .

Следват k реда описващи заявките. Операцията от тип *cut* се задава със следния ред “*cut u v*”, ($1 \leq u, v \leq n$), което означава, че от графа трябва да се премахне реброто между върховете u и v . Операцията от типа *ask* се задава със следния ред “*ask u v*”, ($1 \leq u, v \leq n$), което означава, че е необходимо да се отговори на дали в този момент u и v са в една компонента на свързаност. Гарантирано е, че всяко от ребрата ще се срещне в операцията *cut* точно веднъж.

Изход

За всяка от заявките *ask*, за всеки набор от входни данни, да се изведе на отделен ред низът “*YES*”, без кавичките, ако двата върха лежат в една компонента на свързаност, или “*NO*” в противен случай. Редът на отговорите трябва да съответства на реда на питанията.

Пример

Вход	Изход
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	