Police Query

To help capture criminals on the run, the police are introducing a new computer system. The area covered by the police contains **N** cities and **E** bidirectional roads connecting them. The cities are labelled 1 to **N**. The police often want to catch criminals trying to get from one city to another. Inspectors, looking at a map, try to determine where to set up barricades and roadblocks.

The new computer system should answer the following two types of queries:

1. Consider two cities **A** and **B**, and a road connecting cities $\mathbf{G_1}$ and $\mathbf{G_2}$. Can the criminals get from city **A** to city **B** if that one road is blocked and the criminals can't use it?

2. Consider three cities **A**, **B** and **C**. Can the criminals get from city **A** to city **B** if the entire city **C** is cut off and the criminals can't enter that city?

Write a program that implements the described system.

Input

The first line contains two integers **N** and **E** ($2 \le N \le 10^5$, $1 \le E \le 5*10^5$), the number of cities and roads. Each of the following **E** lines contains two distinct integers between 1 and **N** – the labels of two cities connected by a road. There will be at most one road between any pair of cities.

The following line contains the integer **Q** ($1 \le Q \le 10^5$), the number of queries the system is being tested on. Each of the following **Q** lines contains either four or five integers. The first of these integers is the type of the query – **1** or **2**.

If the query is of **type 1**, then the same line contains four more integers **A**, **B**, $\mathbf{G_1}$ and $\mathbf{G_2}$ as described earlier. **A** and **B** will be different. $\mathbf{G_1}$ and $\mathbf{G_2}$ will represent an existing road.

If the query is of **type 2**, then the same line contains three more integers **A**, **B** and **C**. **A**, **B** and **C** will be distinct integers.

The test data will be such that it is initially possible to get from each city to every other city.

Output

Output the answers to all **Q** queries, one per line. The answer to a query can be *da* (yes) or *ne* (no).

Example

Input:
13 15
1 2
2 3
3 5
2 4
4 6
2 6

1 4
1 7
7 8
7 9
7 10
8 11
8 12
9 12
12 13
5
1 5 13 1 2
1 6 2 1 4
1 13 6 7 8
2 13 6 7
2 13 6 8

Output:
da
da
ne
da