There are *N* cities in one country that are connected with two-way air links. One crazy airline president, Ronald Krump, often changes the flight schedule. More precisely, every day he does the following:

- chooses one of the cities,
- introduces flights from that city towards all other cities where these flights do not currently exist, and at the same time cancels all existing flights from that city

For instance, if from city 5 flights exist towards cities 1 and 2, but not towards cities 3 and 4, after Krump's change, there will exist flights from city 5 towards cities 3 and 4, but not towards cities 1 and 2.

The citizens of this country are wondering if a day could come when the flight schedule will be complete. In other words, when between each two different cities a (direct) flight will exist. Write a programme that will, based on the current flight schedule, determine whether it is possible to have a *Complete Day*, or whether this will never happen, no matter what moves Krump makes.

INPUT

The first line of input contains the integer *N* ($2 \leq N \leq 1000$), the number of cities. The cities are labeled with numbers from 1 to *N*.
The second line contains the integer *M* ($0 \leq M < N*(N-1)/2$), the number of current flights. Each of the following *M* lines contains two different numbers, the labels of the cities that are currently connected.

OUTPUT

The first and only line of output must contain DA (Croatian for "yes") or NE (Croatian for "no").

SAMPLE TESTS

input	input	input
2 0	3 2 1 2 2 3	4 2 1 3 2 4
output	**output**	**output**
DA	NE	DA

Clarification of the first test case: In the first step, Krump will introduce the (only possible) line 1-2.
Clarification of the third case: If Krump first chooses city 1, flights 1-2, 1-4 and 2-4 will exist. If he then chooses city 3, the flight schedule will become complete.