3178 Take a Walk

A *walk* W in a graph G is a finite sequence

$$v_o e_1 v_1 e_2 v_2 \ldots v_{k-1} e_k v_k$$

whose terms are alternately vertices and edges such that, for $1 \leq i \leq k$, the edge e_i has end vertices v_{i-1} and v_i. If the edges $e_1, e_2, \ldots, e_k$ of the walk are distinct, then W is called a *trail*. A trail with $v_o \neq v_k$ is an *open trail*.

If $v_o = v_k$, then W is a closed walk. A tour of G is a closed walk of G that includes every edge of G at least once.

Write a program that determines whether for a graph G:

1. there exists an open trail that includes every edge of G, or not; and
2. there exists a tour that includes every edge of G exactly once, or not

where graph G is undirected, has at least 2 edges, has no self-loops (i.e., edges (v_i, v_i)), but may contain parallel edges (i.e., 2 or more edges having the same end vertices).

Input

The input file consists of several test cases, each with a case number, the set of vertices in a graph, and the set of edges in the graph, as shown in the samples. Assume the vertices are single letters only.

Output

For each of the test cases, output '`Yes`' if the graph has at least one open trail that includes every edge of the graph, and '`No`', if not; and output '`Yes`' if the graph has at least one tour that includes every edge of the graph exactly once, and '`No`' if not.

Sample Input

```
Case 1: { a, b, c, d, e } { (a,b), (b,c), (c,d), (d,a), (b,e), (c,e) }
Case 2: { a, b, c, d, e } { (a,b), (a,c), (b,e), (b,d), (b,c), (d,c),
(d,e), (d,e), (e,c) }
Case 3: { A, B, c, d } { (A,B), (c,d) }
```

Sample Output

```
Yes No
No Yes
No No
```