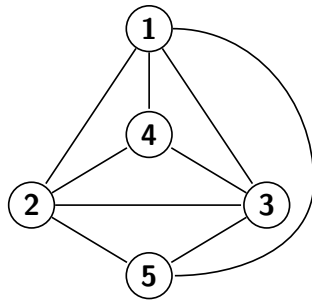


B: A Subway for Boroughgraph

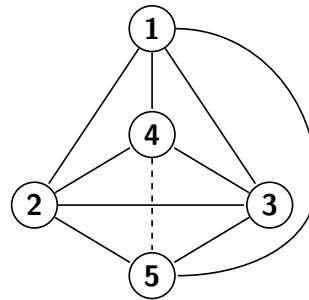
Source file name: `subway.c`, `subway.cpp`, or `subway.java`

The mayor of Boroughgraph has been promising its inhabitants a new subway system for quite a few years, and the people are growing impatient. With barely enough money to complete the construction and even less time, he has hired you, a brilliant Boroughgraphian programmer, to help him with this project.

The mayor, of course, wants the new subway to favor mostly those whom he thinks will vote for him, so he has already determined the subway plan. The only problem is that Boroughgraph was built over a series of swamps, so the subway tunnels can be drilled only at a certain fixed depth below ground without risking collapse. Since there is no leftover money to invest in overpasses or ground-level lines, two subway lines can never intersect (except possibly at their endpoints), but their paths need not be straight.



The subway can be built



The subway cannot be built

This is where you come in. Given a specification of a subway network, can you write a program that tells the mayor whether it is possible to build it in Boroughgraph?

Input

The input consists of several subway network specifications. Each specification begins with a line containing a single integer N indicating the number of subway stations ($2 \leq N \leq 64$), which are numbered from 1 to N . Then follow N lines indicating the layout of the subway system: line i contains exactly d_i blank-separated integers from 1 to N (excluding i), indicating the stations to which station i should be connected ($1 \leq d_i \leq N-1$). The subway network is bidirectional, so if station i appears in station j 's line, then it is guaranteed that station j will appear in station i 's line.

The input must be read from standard input.

Output

For each specification, print a line with the character 'Y' if the subway can be built in Boroughgraph, or with the character 'N', otherwise.

The output must be written to standard output.

Sample Input	Sample Output
5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 1 2 3 5 2 3 4 5 1 3 4 5 1 2 4 5 1 2 3 5 1 2 3 4	Y N