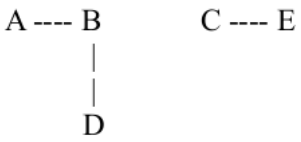
Problem J. Connected Components

Source file name:	J.c, J.cpp, J.java, J.py
Input:	`Standard`
Output:	`Standard`

A graph $G = (V, E)$ is connected if a path can be found in 0 or more intermediate steps between any pair of nodes in G. The following graph is not connected:



It contains, however, many connected subgraphs, which are: $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, $\{A, B\}$, $\{B, D\}$, $\{C, E\}$, $\{A, B, D\}$.

A connected subgraph is maximal if there are no nodes and edges in the original graph that could be added to the subgraph and still leave it connected. In the previous example, $\{C, E\}$ and $\{A, B, D\}$ are maximal.

Your job is to find, for the given graphs, how many connected subsets there are.

Input

There will be many input cases, each of which will be separated by a blank line. The last input case will be followed by EOF.

Each of the input cases starts with a line that contains a single upper case alphabetic character, which represents the largest node name in the graph. Each successive line contains a pair of upper case alphabetic characters denoting an edge in the graph. The graph represented by the input is undirected.

Output

For each test case print a single line with the number of maximal connected subgraphs.

Example

Input	Output
E AB CE DB B AB C	2 1 3