

A - Andy's First Dictionary

Source file name: `andy.py`

Time limit: 1 second

Andy, 8, has a dream -he wants to produce his very own dictionary. This is not an easy task for him, as the number of words that he knows is, well, not quite enough. Instead of thinking up all the words himself, he has a brilliant idea. From his bookshelf he would pick one of his favourite story books, from which he would copy out all the distinct words. By arranging the words in alphabetical order, he is done! Of course, it is a really time-consuming job, and this is where a computer program is helpful.

You are asked to write a program that lists all the different words in the input text. In this problem, a word is defined as a consecutive sequence of alphabets, in upper and/or lower case. Words with only one letter are also to be considered. Furthermore, your program must be CaSe InSeNsItIvE. For example, words like "Apple", "apple", or "APPLE" must be considered the same.

Input

The input is a text with no more than 5000 lines. An input line has at most 200 characters.

The input must be read from standard input.

Output

Your output should give a list of different words that appears in the input text, one in a line. The words should all be in lower case, sorted in alphabetical order.

The output must be written to standard output.

Sample Input	Sample Output
Adventures in Disneyland	a
	adventures
Two blondes were going to Disneyland when they came to a fork in the	blondes
road. The sign read: "Disneyland Left."	came
	disneyland
So they went home.	fork
	going
	home
	in
	left
	read
	road
	sign
	so
	the
	they
	to
	two
	went
	were
	when

B - X-Plosives

Source file name: `xplosives.py`

Time limit: 1 second

A secret service developed a new kind of explosive that attain its volatile property only when a specific association of products occurs. Each product is a mix of two different simple compounds, to which we call a *binding pair*. If $N > 2$, then mixing N different binding pairs containing N simple compounds creates a powerful explosive. For example, the binding pairs $A + B$, $B + C$, $A + C$ (three pairs, three compounds) result in an explosive, while $A + B$, $B + C$, $A + D$ (three pairs, four compounds) does not.

You are not a secret agent but only a guy in a delivery agency with one dangerous problem: receive binding pairs in sequential order and place them in a cargo ship. However, you must avoid placing in the same room an explosive association. So, after placing a set of pairs, if you receive one pair that might produce an explosion with some of the pairs already in stock, you must refuse it, otherwise, you must accept it. Compute the number of refusals given a sequence of binding pairs.

Lets assume you receive the following sequence: $A + B$, $G + B$, $D + F$, $A + E$, $E + G$, $F + H$. You would accept the first four pairs but then refuse $E + G$ since it would be possible to make the following explosive with the previous pairs: $A + B$, $G + B$, $A + E$, $E + G$ (4 pairs with 4 simple compounds). Finally, you would accept the last pair, $F + H$.

Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line. Instead of letters we will use integers to represent compounds. The input contains several lines. Each line (except the last) consists of two integers (each integer lies between 0 and 10^5) separated by a single space, representing a binding pair. Each test case ends in a line with the number `-1`. You may assume that no repeated binding pairs appears in the input.

The input must be read from standard input.

Output

For each test case, output a single line with the number of refusals.

The output must be written to standard output.

Sample Input	Sample Output
1 2 3 4 3 5 3 1 2 3 4 1 2 6 6 5 -1	3

C - Conformity

Source file name: `conformity.py`

Time limit: 1 second

Frosh commencing their studies at Waterloo have diverse interests, as evidenced by their desire to take various combinations of courses from among those available.

University administrators are uncomfortable with this situation, and therefore wish to offer a *conformity prize* to frosh who choose the most popular combination of courses. How many frosh will win the prize? The popularity of a combination is the number of frosh selecting exactly the same combination of courses. A combination of courses is considered most popular if no other combination has higher popularity.

Input

The input consists of several test cases followed by a line containing 0. Each test case begins with an integer $1 \leq n \leq 10000$, the number of frosh. For each frosh, a line follows containing the course numbers of five distinct courses selected by the frosh. Each course number is an integer between 100 and 499.

The input must be read from standard input.

Output

For each line of input, you should output a single line giving the total number of students taking some combination of courses that is most popular.

The output must be written to standard output.

Sample Input	Sample Output
3	2
100 101 102 103 488	3
100 200 300 101 102	
103 102 101 488 100	
3	
200 202 204 206 208	
123 234 345 456 321	
100 200 300 400 444	
0	