'''Mr Parandhamayya working with words.

He is given a word W, you need to divide the word into N non-empty parts,

such that all the newly formed words should be distinct, and

if you append all those words should form original word W.

Your task is to help Mr Parandhamayya to divide the word into N parts,

such that, the value of N should be maximized, and print N.

Input Format:

-------------

Line-1: A string W, a word.

Output Format:

--------------

Print an integer result, the value of N.

Sample Input-1:

---------------

banana

Sample Output-1:

----------------

4

Explanation:

------------

One way to divide the word is "b","a","n","ana".

If you divide it like "b","a","n","an","a".The word "a" will be repeated.

So it is not allowed to divide like the second way.

Sample Input-2:

---------------

mississippi

Sample Output-2:

----------------

7

Explanation:

------------

One of the way to divide the word is "m","i","s","si","ssi","p","pi".

NOTE: Subsequences are not allowed.

aababb 4

'''

s=input()

l=[]

i=0

ss=""

flag=False

while(i<len(s)):

if(s[i] not in l ):

# ss=""

l.append(s[i])

i+=1

continue

ss+=s[i]

# flag=True

if(ss not in l):

# flag=False

l.append(ss)

ss=""

i+=1

print(len(l))

# print(l)

In a joint family of N members, every person assigned with an ID, an integer value.

and the entire family is arranged in the from of binary tree.

You will be given the preOrder data and inOrder data of the tree.

Your task is to build the original fammily tree using the given data.

and return the root of the tree.

Implement the class Solution:

- Node buildBinaryTree(int[] preOrder, int[] inOrder):

return the root node of the tree.

Input Format:

-------------

Line-1: An integer N, number of persons in the family.

Line-2: N space separated integers, preOrder[] data.

Line-3: N space separated integers, inOrder[] data.

Output Format:

--------------

Return the root node of the original tree.

Sample Input-1:

---------------

7

1 2 4 5 3 6 7

4 2 5 1 6 3 7

Sample Output-1:

----------------

1 2 3 4 5 6 7

Sample Input-2:

---------------

8

1 2 8 3 4 6 5 7

2 8 1 6 4 3 5 7

Sample Output-2:

----------------

1 2 3 8 4 5 6 7

import java.util.\*;

/\*

Reference of Node

class Node {

public int data;

public Node left;

public Node right;

public Node(int value) {

data = value;

left = null;

right = null;

}

}

\*/

class Solution{

Node Helper(int pst,int inst,int inend,int[] preOrder,int[] inOrder){

if (pst>=preOrder.length || inst>inend ||inend>=inOrder.length){

return null;

}

Node t=new Node(preOrder[pst]);

int ind=0;

for(int i=inst;i<=inend;i++){

if (inOrder[i]==t.data){

ind=i;

break;

}

}

t.left=Helper(pst+1,inst,ind-1,preOrder,inOrder);

t.right=Helper(pst+(ind-inst)+1,ind+1,inend,preOrder,inOrder);

return t;

}

Node buildBinaryTree(int[] preOrder, int[] inOrder){

// Implement your code here.

return Helper(0,0,inOrder.length-1,preOrder,inOrder);

}

}

For X-Mas, santa claus is preparing a X-Mas Tree with set of Bulbs.

The bulbs are of different voltages, and preparation of tree as follows:

- The bulbs are arranged in level-wise, levels are numbered from 0,1,2,3..

so on.

- At level-0: There will be only one bulb as root bulb.,

- From next level onwards, we have to attach atmost two bulbs, one is to

left side, and the other is for right side of every bulb in previous level.

- A bulb B is said to be BRIGHT, if all the bulbs in the path root bulb to

bulb B, there are no bulbs with a voltage greater than bulb B.

You will be given the root of the X-Mas Tree,

Your task is to return the number of BRIGHT bulbs in the X-Mastree.

Implement the class Solution:

1.public int brighterBulbs(Node root): returns an integer output.

NOTE:

-----

In the input -1 indicates no bulb (null node).

Input Format:

-------------

An integer, number of bulbs.

A single line of space separated integers, voltages of the set of N bulbs.

Output Format:

--------------

Print the number of BRIGHT bulbs in the tree.

Sample Input-1:

---------------

7

5 6 3 8 2 6 9

Sample Output-1:

----------------

5

Sample Input-2:

---------------

13

1 2 3 4 5 6 7 8 9 10 -1 -1 11

Sample Output-2:

----------------

11