**QUESTION:**

You have an interesting string S of length N. It is interesting because you can rearrange the characters of this string in any order. You want to cut this string into some contiguous pieces such that after cutting, all the pieces are equal to one another. You can't rearrange the characters in the cut pieces or join the pieces together. You want to make the number of pieces as large as possible. What is the maximum number of pieces you can get?

Note: You can observe that you may not want to cut the string at all, therefore the number of pieces is 1. Hence, the answer always exists.

**Parameters:**

S :: STRING The first line contains a string, S, denoting the string. length(S) :: 1 -> 2 \* 10^5

**Test Cases**

Case#: 1

Input: zzzzz

Output: 5

You can cut it into 5 pieces "z" + "z" + "z" + "z" + "z"

Case#: 2

Input: ababcc

Output: 2

Rearrange the string as abcabc. you can cut it into "abc" + "abc", hence the answer is 2

Case#: 3

Input: abccdcabacda

Output: 2

Rearrange the string as "dcbaca" + "dcbaca", the answer is 2.

**ANSWER:**

import java.util.\*;

public class infyqns5

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int c=1,b=1,min=0;

String s=sc.next();

char ch[]=s.toCharArray();

Arrays.sort(ch);

System.out.println(ch);

int a[]=new int[ch.length];

for(int i=0;i<ch.length-1;i=i+b)

{

for(int j=i+1;j<ch.length;j++)

{

if(ch[i]==ch[j])

{

b=c++;

}

}

a[i]=b;

c=1;

}

if(a.length>1)

{

min=a[0];

for(int i=1;i<a.length;i++)

{

if(a[i]!=0)

{

if(a[i]<min)

{

min=a[i];

}

}

}

System.out.println(min-1);

}

if(a.length==1)

{

min=a[0]+1;

System.out.println(min+1);

}

}

}