Given a string, remove characters until the string is made up of any two alternating characters. When you choose a character to remove, all instances of that character must be removed. Determine the longest string possible that contains just two alternating letters.

**Example**

Delete a, to leave bcdbd. Now, remove the character c to leave the valid string bdbd with a length of 4. Removing either b or d at any point would not result in a valid string. Return .

Given a string , convert it to the longest possible string  made up only of alternating characters. Return the length of string . If no string  can be formed, return .

**Function Description**

Complete the *alternate* function in the editor below.

alternate has the following parameter(s):

* *string s:* a string

**Returns**.

* *int:* the length of the longest valid string, or  if there are none

**Input Format**

The first line contains a single integer that denotes the length of .  
The second line contains string .

**Constraints**

**Sample Input**

STDIN Function

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

10 length of s = 10

beabeefeab s = 'beabeefeab'

**Sample Output**

5

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.regex.\*;

class Result {

    /\*

     \* Complete the 'alternate' function below.

     \*

     \* The function is expected to return an INTEGER.

     \* The function accepts STRING s as parameter.

     \*/

    public static int alternate(String s)

     {

    // Write your code here

String res="";

for(int i=0;i<26;i++)

{

   for(int j=i+1; j<26; j++){

        char a=(char)('a'+i);

        char b=(char)('a'+j);

        String cur="";

        for(int k=0; k<s.length(); k++)

        {

            if(s.charAt(k)==a||s.charAt(k)==b)

            {

                cur+=s.charAt(k);

            }

        }

        if (cur.length()<res.length())

        continue;

        if(isGood(cur)) res=cur;

   }

}

        System.out.println(res.length());

        return res.length();

        }

public static boolean isGood(String s)

{

    if (s.length()==1)

    return false;

    for(int i=1; i<s.length(); i++)

    {

        if(s.charAt(i)==s.charAt(i-1))

        return false;

    }

    return true;

}

}

    public class Solution {

    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

        int l = Integer.parseInt(bufferedReader.readLine().trim());

        String s = bufferedReader.readLine();

        int result = Result.alternate(s);

        bufferedWriter.write(String.valueOf(result));

        bufferedWriter.newLine();

        bufferedReader.close();

        bufferedWriter.close();

    }

}



