

Two Characters

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

$s = \text{'abaacdabd'}$

Delete **a**, to leave **bcdabd**. 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 **4**.

Given a string s , convert it to the longest possible string t made up only of alternating characters. Return the length of string t . If no string t can be formed, return **0**.

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 **0** if there are none

Input Format

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

Constraints

- $1 \leq \text{length of } s \leq 1000$
- $s[i] \in \text{ascii[a-z]}$

Sample Input

STDIN	Function
-----	-----
10	length of s = 10
beabeefeab	s = 'beabeefeab'

Sample Output

5

Explanation

The characters present in s are `a`, `b`, `e`, and `f`. This means that t must consist of *two* of those characters and we must delete *two* others. Our choices for characters to leave are `[a,b]`, `[a,e]`, `[a,f]`, `[b,e]`, `[b,f]` and `[e,f]`.

If we delete `e` and `f`, the resulting string is `babab`. This is a valid t as there are only two distinct characters (`a` and `b`), and they are alternating within the string.

If we delete `a` and `f`, the resulting string is `bebeeeeb`. This is not a valid string t because there are consecutive `e`'s present. Removing them would leave consecutive `b`'s, so this fails to produce a valid string t .

Other cases are solved similarly.

`babab` is the longest string we can create.