

Day 73 coding Statement :

A string is called *boring* if all the characters of the string are **same**.

You are given a string S of length N , consisting of lowercase english alphabets. Find the length of the longest *boring* substring of S which occurs **more than once**.

Note that if there is no *boring* substring which occurs more than once in S , the answer will be 00.

A substring is obtained by deleting some (possibly zero) elements from the beginning of the string and some (possibly zero) elements from the end of the string.

Input Format

- The first line of input will contain a single integer T , denoting the number of test cases.
- Each test case consists of two lines of input.
 - The first line of each test case contains an integer N , denoting the length of string S .
 - The next contains string S .

Output Format

For each test case, output on a new line, the length of the longest *boring* substring of S which occurs **more than once**.

Sample Input

4

3

aaa

3

abc

5

bcaca

6

caabaa

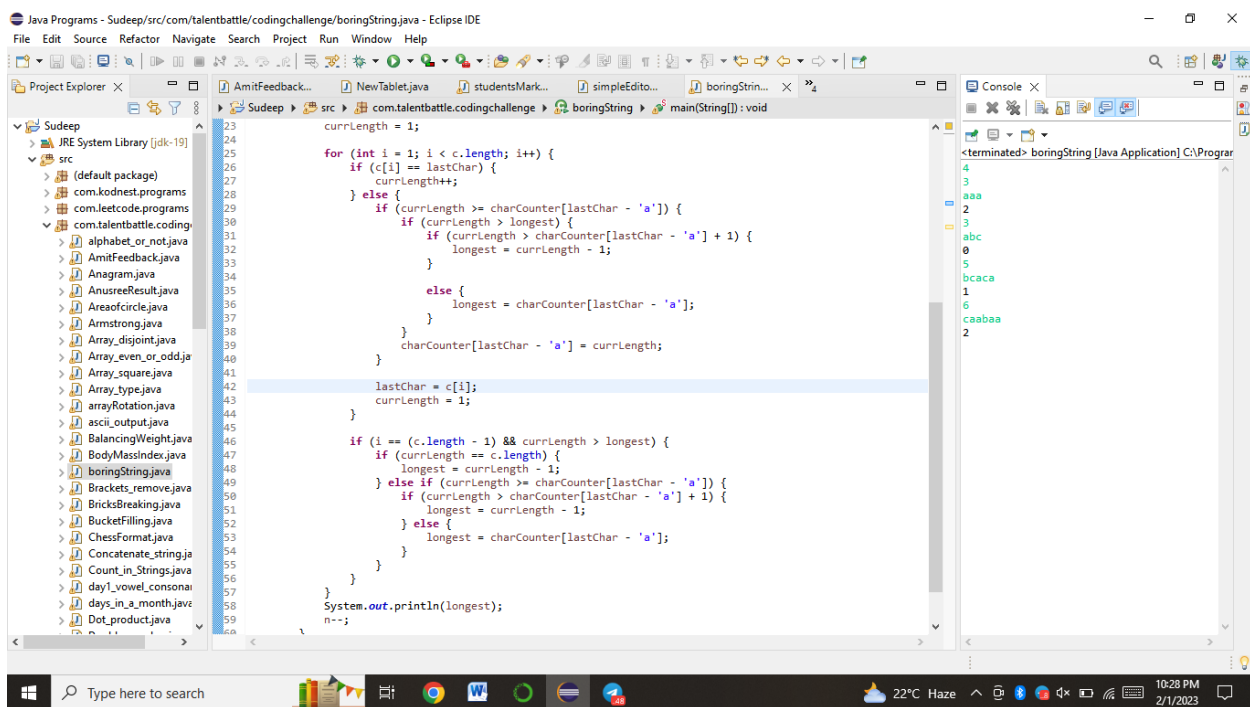
Sample Output

2

0

1

2



The screenshot shows the Eclipse IDE interface. The main editor displays a Java file named `boringString.java` with the following code:

```
23 currLength = 1;
24
25 for (int i = 1; i < c.length; i++) {
26     if (c[i] == lastChar) {
27         currLength++;
28     } else {
29         if (currLength >= charCounter[lastChar - 'a']) {
30             if (currLength > longest) {
31                 if (currLength > charCounter[lastChar - 'a'] + 1) {
32                     longest = currLength - 1;
33                 }
34             }
35             else {
36                 longest = charCounter[lastChar - 'a'];
37             }
38             charCounter[lastChar - 'a'] = currLength;
39         }
40     }
41     lastChar = c[i];
42     currLength = 1;
43 }
44
45 if (i == (c.length - 1) && currLength > longest) {
46     if (currLength == c.length) {
47         longest = currLength - 1;
48     } else if (currLength >= charCounter[lastChar - 'a']) {
49         if (currLength > charCounter[lastChar - 'a'] + 1) {
50             longest = currLength - 1;
51         } else {
52             longest = charCounter[lastChar - 'a'];
53         }
54     }
55 }
56 System.out.println(longest);
57 n--;
58 }
59 }
```

The Project Explorer on the left shows the file structure, including `boringString.java`. The Console on the right displays the output of the program:

```
<terminated> boringString [Java Application] C:\Program
4
3
2
3
abc
0
5
bcaca
1
6
caabaa
2
```

The Windows taskbar at the bottom shows the system clock as 10:28 PM on 2/1/2023, with a temperature of 22°C and weather conditions of Haze.