Palindrome

A palindrome is a word, sentence or sequence of characters which reads the same backward as forward. Consider a language of $\{A,B,C,D\}$. ABBCDAA is string of this language and the **subsequences** of the string are A, B, ABC, ABBA, ACDAA, ACA, BD, BDA, etc. Thus, A, B, ABBA, ACA are the Palindrome as they read the same in both ways.

As you have already noticed, there some subsequences of a string from a language of {A, B, C, D}. Write an efficient algorithm that finds the longest subsequence that is a palindrome.

Again, you are not allowed to use optimization options at compilation and the runtime of your program processing the 10 cases should not exceed 10 second.

[Input]
An input file "input.txt" will be given with 10 test cases. The first line of a test case contains a single positive integer n ($10 \le n \le 5,000$), and the second line is a string formed with the set of letters {A, B, C, D}.

[Output]
For each case, you should print the case number as #x where x is the index of the case. Then, print the length of the longest subsequence of the string which is a palindrome followed by space.

You must produce your results as "output.txt"

[Example]
Input (input.txt)

```
10                                                              ← First Case
ABBCDAAAAA
15                                                              ← Second Case
ABCDDDDCCDDAABB
...
```

Output (output.txt)

```
#1  6
#2  10
...
```