|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Odd-Even Steven**    Steven is trying a new operation over a string. The operation is that he will merge all the same consecutive characters in a string e.g. if a string is aabb, he will convert it to ab. Now, after doing this for a lot of strings, Steven observed an interesting thing that some of the strings became a palindrome (strings which are same when reading backward and forward). He gave these palindromic strings and the original strings to his son. His son then asked him a difficult question, that how many palindromic sub-strings of even length and odd length could be formed from the original string before and after merging operation. Steven was troubled with this question and asked for your help.    **Input Specification:**   |  | | --- | | The string. |   **Note:** **The input string is made up of characters 'a' and 'b' only.**    **Output Specification:**   |  | | --- | | A 2 element array with the first element as the number of even length sub-strings and the second element as the number of odd length sub-strings. |     **Example 1:**   |  | | --- | | **input1:**  aabb |      |  | | --- | | **Output:** {2,4} |     **Explanation:**  Here, the even length sub-strings which can be formed are {aa}, {bb} which can be converted to {a},{b} and they will be palindromes.  Odd length substrings can be {a,a,b,b} which can be converted to palindromes after merging. Note that substrings {abb,aab} will not be counted because after merging they will give {ab} and {ab} which are not palindromes.    **Example 2:**   |  | | --- | | **input1:** baa |      |  | | --- | | **Output:** {1,3} |     **Explanation:**  Here, the even length substrings which can be formed are {aa} which can be converted to {a}. Odd length sub-strings can be {b,a,a} which are palindromes. Again note that the sub-string {baa} cannot be used because it will give {ba} after merging which is not a palindrome.      // |
|  | Q.2 | **You Know Nothin' Jon !**    Jon has recently started taking programming classes. His friends say that "Jon knows nothing!". But Jon wants to prove them wrong by completing the given task.    He was given a string A and three substrings D, E & F. He has to find sub-strings D or E or F in A and then either remove them or leave them as it is.    If D or E or F is removed from the given string, then he gets a new string A1.This process is repeated on A1to get A2 and so on till the process is not possible.  Let the final string be Af.    Help Jon in determining minimum possible length of final string.    **Input Specification:**   |  | | --- | | **input1**: String A. **input2**: Sub String D. **input2**: Sub String E. **input2**: Sub String F. |     **Output Specification:**   |  | | --- | | Return length of the final String |     **Example 1:**   |  | | --- | | **input1**: aaabccd **input2**: abc  **input3**: ac  **input4**: aaa |      |  | | --- | | **Output**: 2 |     **Explanation:**  Remove abc from aaabccd, we get aacd.  Remove ac from aacd, we get ad.  Further the process cannot be repeated. So the length of final string ad is 2.    **Example 2:**   |  | | --- | | **input1**: abcdefabcd **input2**: ab **input3**: cd **input4**: e |      |  | | --- | | **Output**: 1 |     **Explanation:**  Remove ab from abcdefabcd we get cdefabcd.  Remove cd from cdefabcd we get efabcd.  Remove e from efabcd, we get fabcd.  On repeating the process we get the final string as f which cannot be further processed.  So the length of final ​string, is 1.    // |
|  | Q.3 | **Happy Strings**  Allie is good friends with Samuel. It is Samuel's birthday and Allie brought a gift for him. The gift is a game called happy strings. For playing the game, Allie picks up a string and chooses a number A. Samuel has to find the maximum distance (distance is given by number of characters in between) between any 2 subsequent same characters in the string S given that a character X is said to be present at index i if it is present in any of the index from |i - A| to |i + A|.    **Input Specification:**   |  | | --- | | **input1:** S, denoting the string in the game containing only characters 'A' - 'Z'.  **input2:** A, the number chosen by Allie.  **input3:** Length of input1 |   **Output Specification:**   |  | | --- | | Your function should return the maximum distance between any 2 same characters satisfying the given constraint. |     **Example 1:**   |  | | --- | | **input1:** AABAABAA  **input2:**1  **input3:** 8 |      |  | | --- | | **Output:** 0 |     **Explanation:**   * Considering character A - Maximum distance = 0. * Considering character B - Maximum distance = 0.     **Example 2:**   |  | | --- | | **input1:**AABAAABAA  **input2:**1  **input3:** 9 |      |  | | --- | | **Output:** 1 |     **Explanation:**   * Considering character A - Maximum distance = 0. * Considering character B - Maximum distance = 1. |