**STRINGS ASSIGNMENT**

**PROBLEM 01:**

**class Solution {**

**public boolean backspaceCompare(String s, String t) {**

**return buildString(s).equals(buildString(t));**

**}**

**private String buildString(String str) {**

**Stack<Character> stack = new Stack<>();**

**for (char c : str.toCharArray()) {**

**if (c != '#') {**

**stack.push(c);**

**} else if (!stack.isEmpty()) {**

**stack.pop();**

**}**

**}**

**StringBuilder sb = new StringBuilder();**

**while (!stack.isEmpty()) {**

**sb.append(stack.pop());**

**}**

**return sb.toString();**

**}**

**}**

**PROBLEM 02:**

**class Solution {**

**public int minFlips(String s) {**

**StringBuilder sb = new StringBuilder(s).append(s);**

**StringBuilder alt1 = new StringBuilder();**

**StringBuilder alt2 = new StringBuilder();**

**for (int i = 0; i < sb.length(); i++) {**

**if (i % 2 == 0) {**

**alt1.append(0);**

**alt2.append(1);**

**}**

**else {**

**alt1.append(1);**

**alt2.append(0);**

**}**

**}**

**int diff1 = 0, diff2 = 0;**

**int l = 0;**

**int res = sb.length();**

**for (int r = 0; r < sb.length(); r++) {**

**if (alt1.charAt(r) != sb.charAt(r)) diff1++;**

**if (alt2.charAt(r) != sb.charAt(r)) diff2++;**

**if (r - l + 1 > s.length()) {**

**if (alt1.charAt(l) != sb.charAt(l)) diff1--;**

**if (alt2.charAt(l) != sb.charAt(l)) diff2--;**

**l++;**

**}**

**if (r - l + 1 == s.length()) {**

**res = Math.min(res, Math.min(diff1, diff2));**

**}**

**}**

**return res;**

**}**

**}**

**PROBLEM 05:**

**class Solution {**

**public int minInsertions(String s) {**

**Stack<Character> stack=new Stack<>();**

**int count=0;**

**for(int i=0;i<s.length();i++){**

**if(s.charAt(i)=='('){**

**stack.push(s.charAt(i));**

**}else{**

**if(i+1<s.length()&& s.charAt(i+1)==')'){**

**i++;**

**}else count++;**

**if(!stack.isEmpty()){**

**stack.pop();**

**}else count++ ;**

**}**

**}**

**count+=2\*stack.size();**

**return count;**

**}**

**}**

**PROBLEM 08:**

**class Solution {**

**public int countSubstrings(String s) {**

**int count=0;**

**String str="";**

**for(int i=0; i<s.length();i++){**

**str="";**

**for(int j=i; j<s.length();j++){**

**Character ch=s.charAt(j);**

**str+=ch;**

**boolean bool=isPalindrome(str);**

**if(bool) count++;**

**}**

**}**

**return count;**

**}**

**public boolean isPalindrome(String str){**

**int i=0,j=str.length()-1;**

**while(i<=j){**

**char a=str.charAt(i);**

**char b=str.charAt(j);**

**if(a!=b) return false;**

**i++;**

**j--;**

**}**

**return true;**

**}**

**}**

**PROBLEM 09:**

**import java.io.\*;**

**import java.util.\*;**

**public class Main {**

**public static String compression1(String s) {**

**String ans = "";**

**for (int i = 0 ; i < s.length(); i++) {**

**while (i < s.length() - 1 && s.charAt(i) == s.charAt(i + 1)) {**

**i++;**

**}**

**ans += s.charAt(i);**

**}**

**return ans;**

**}**

**public static String compression2(String s) {**

**String ans = "";**

**for (int i = 0 ; i < s.length(); i++) {**

**int count = 1;**

**while (i < s.length() - 1**

**&& s.charAt(i) == s.charAt(i + 1)) {**

**count++;**

**i++;**

**}**

**ans += s.charAt(i);**

**if (count > 1) {**

**ans += count;**

**}**

**}**

**return ans;**

**}**

**public static void main(String[] args) {**

**Scanner scn = new Scanner(System.in);**

**String s = scn.next();**

**System.out.println(compression1(s));**

**System.out.println(compression2(s));**

**}**

**}**