public class PalindromePossiblity {

    public static boolean isPalindromePossible(String input)

    {

        char[] charStr = input.toCharArray();

        int len = input.length(), i;

        int diffCount = 0;

        char[][] diff = new char[2][2];

        for (i = 0; i < len / 2; i++) {

            if (charStr[i] != charStr[len - i - 1]) {

                // 3rd differences encountered and its no longer

                // possible to make is palindrome by one swap

                if (diffCount == 2)

                    return false;

                // record the different character

                diff[diffCount][0] = charStr[i];

                // store the different characters

                diff[diffCount++][1] = charStr[len - i - 1];

            }

        }

        switch (diffCount) {

        case 0:

            return true;

        case 1:

            char midChar = charStr[i];

            if (len % 2 != 0 && (diff[0][0] == midChar

                                 || diff[0][1] == midChar))

                return true;

        case 2:

            if ((diff[0][0] == diff[1][0] && diff[0][1] == diff[1][1])

                || (diff[0][0] == diff[1][1] && diff[0][1] == diff[1][0]))

                return true;

        }

        return false;

    }

    public static void main(String[] args)

    {

        System.out.println(isPalindromePossible("bbg"));

        System.out.println(isPalindromePossible("bdababd"));

        System.out.println(isPalindromePossible("gcagac"));

    }

}