

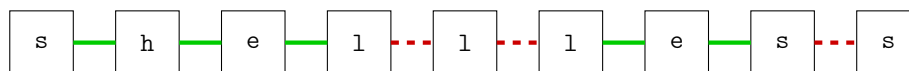
Zmiennoliterowe słowo (rozwiązanie)

Autor zadania: **Karol Pokorski**
Opracowanie: **Bartosz Kostka, Mateusz Lewko**
Opis rozwiązania: **Bartosz Kostka**



Przypomnijmy, że słowo jest zmiennoliterowe, jeśli każde jego dwie sąsiednie litery są różne. Powiemy, że para sąsiednich liter jest *zła*, jeżeli obie litery są równe. Analogicznie, para sąsiednich liter będzie *dobra*, jeżeli są różne. Naszym zadaniem jest sprawić, poprzez zakrywanie liter, aby słowo z wejścia miało zero złych par (jako że będzie wtedy zmiennoliterowe).

Przypatrzmy się słowu z pierwszego testu przykładowego.



Mamy tutaj trzy pary sąsiednich liter, które są złe (są połączone przerywaną czerwoną kreską na rysunku poniżej). Pozostałe sąsiadujące litery, które są dobre, zostały połączone kolorem zielonym.

Zastanówmy się teraz, co się dzieje w przypadku zakrywania liter. Gdy zakrywamy jedną z liter, wtedy tak naprawdę usuwamy ją ze słowa, jednocześnie usuwając ewentualne parowanie z sąsiadami na lewo i prawo, a litery na lewo i prawo (o ile istnieją) dostają nowego sąsiada, który może stworzyć dobrą albo złą parę.

Zauważmy, że nie możemy nigdy poprzez zakrycie litery usunąć więcej niż jedną złą parę. Jeżeli usunęlibyśmy dwie złe pary to oznaczałoby, że trzy kolejne litery są równe. Po zakryciu środkowej litery z tych trzech, zawsze otrzymamy znowu jedną złą parę, zatem liczba wszystkich złych par w słowie zmniejszy się o jeden.



Zatem jeżeli wiemy, że przez zakrycie jednej literki możemy usunąć co najwyżej jedną złą parę, to bardzo prostą strategią jest przejście słowa od lewej do prawej i usuwanie np. zawsze lewej literki ze złej pary. Taki manewr powoduje, że sumaryczna liczba złych par w słowie zawsze zmniejsza się o jeden.



Podsumowując, widzimy że z każdą zakrytą literką jesteśmy w stanie usunąć **dokładnie** jedną złą parę. Stąd minimalna liczba zakrytych literek to liczba złych par, czyli liczba par sąsiednich liter, które są równe. Zliczyć takie pary możemy w czasie $O(n)$, gdzie n to długość słowa danego na wejściu, przechodząc w pętli po tym słowie i sprawdzając czy dwie sąsiednie literki są równe.

zmi.py

```
1 def main():
2     # Wczytujemy słowo z wejścia.
3     s = input()
4
5     # Tworzymy zmienną na wynik.
6     zakryte = 0
7
8     # Przechodzimy całe słowo, patrząc na sąsiednie litery.
9     # Będziemy sprawdzać litery na pozycjach i-1 oraz i, dlatego zaczynamy pętlę od 1.
10    for i in range(1, len(s)):
11        # Jeżeli litery są równe, musimy którąś zakryć.
12        if s[i-1] == s[i]: zakryte += 1
13
14    # Wypisujemy wynik.
15    print(zakryte)
16
17
18 main()
```



```
1 #include "bits/stdc++.h"
2
3 using namespace std;
4
5 int main() {
6     // Deklarujemy zmienną i wczytujemy do niej słowo z wejścia.
7     string s;
8     cin >> s;
9
10    // Deklarujemy zmienną na wynik.
11    int zakryte = 0;
12
13    // Przechodzimy całe słowo, patrząc na sąsiednie litery.
14    // Będziemy sprawdzać litery na pozycjach i-1 oraz i, dlatego zaczynamy pętlę od 1.
15    for (int i=1; i<(int)s.size(); i++) {
16        // Jeżeli litery są równe, musimy którąś zakryć.
17        if (s[i-1] == s[i]) zakryte++;
18    }
19
20    // Wypisujemy wynik.
21    cout << zakryte << "\n";
22 }
```

