Problem: Alternating Characters

You are given a string containing characters  and  only, your task is to change it into a string such that every two consecutive characters are different. To do this, you are allowed to delete one or more characters in the string.

Your task is to find the minimum number of required deletions.

For example, string  should be changed to  by deleting one character .

**Input Format**

The first line contains an integer , i.e. the number of test cases.   
The next  lines contain a string .

**Constraints**

**Output Format**

For each test case, print the minimum number of deletions required in a new line.

**Sample Input**

5

AAAA

BBBBB

ABABABAB

BABABA

AAABBB

**Sample Output**

3

4

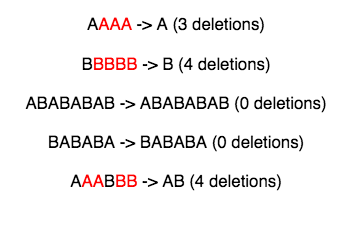
0

0

4

**Explanation**

The characters marked red are the ones that need to be deleted so that the string doesn't have two same consecutive characters.



Solution

int main() {

int cases;

cin>>cases;

string str[cases];

for(int i=0; i<cases; i++)

{

cin>>str[i];

string temp = str[i];

int length = temp.length();

int previous = 0; int next = 1; int del = 0;

while(previous <= length-1 && next <= length)

{ if(temp[next] == temp[previous])

{

del+=1; next+=1;

}

else

{

previous=next; next+=1;

}

}

cout<<del<<endl;

}

return 0;

}

Elegant Solution

int main()

{

int cases;

cin>>cases;

for(int i=0; i<cases; i++)

{

int count = 0; string temp;

cin>>temp;

for(int j=0; j < temp.length()-1; j++)

{

if( temp[j] == temp[j+1] )

{ count++; }

}

cout<<count<<endl;

}

return 0;

}

`’Anshul AgGarwal