-------------------------------------------------------------

O(n^3)

#include<bits/stdc++.h>

using namespace std;

int solve(string s){

int n = s.size();

vector<vector<bool> > pall(n,vector<bool>(n,false));

// for storing palindrome

for(int g=0;g<n;g++){

for(int i=0,j=g;j<n;i++,j++){

if(g==0){

pall[i][j] = true;

}

else if(g==1){

if(s[i]==s[j]){

pall[i][j] = true;

}

}

else{

if(s[i]==s[j]){

pall[i][j] = pall[i+1][j-1];

}

}

}

}

// actual dp to store min cut

vector<vector<int> > dp(n,vector<int>(n));

for(int g=0;g<n;g++){

for(int i=0,j=g;j<n;i++,j++){

if(g==0){

dp[i][j] = 0;

}

else if(g==1){

if(s[i]==s[j]){

dp[i][j] = 0;

}

else{

dp[i][j] = 1;

}

}

else{

if(pall[i][j]){

dp[i][j] = 0;

}

else{

int min\_ = INT\_MAX;

for(int k=i;k<j;k++){

min\_ = min(min\_, dp[i][k]+dp[k+1][j]+1);

}

dp[i][j] = min\_;

}

}

}

}

return dp[0][n-1];

}

int main(){

string s;

cin>>s;

cout<<solve(s);

}

---------------------------------------------------------------

O(n^2)

#include<bits/stdc++.h>

using namespace std;

int solve(string s){

int n = s.size();

vector<vector<bool> > pall(n,vector<bool>(n,false));

// for storing pallindrome

for(int g=0;g<n;g++){

for(int i=0,j=g;j<n;i++,j++){

if(g==0){

pall[i][j] = true;

}

else if(g==1){

if(s[i]==s[j]){

pall[i][j] = true;

}

}

else{

if(s[i]==s[j]){

pall[i][j] = pall[i+1][j-1];

}

}

}

}

// actual dp to store min cut

vector<int> dp(n);

dp[0] = 0;

for(int j=1;j<n;j++){

if(pall[0][j]){

dp[j] = 0;

}

else{

int min\_ = INT\_MAX;

for(int i=j;i>=1;i--){

if(pall[i][j]){

min\_ = min(min\_,dp[i-1]);

}

}

dp[j] = min\_ + 1;

}

}

return dp[n-1];

}

int main(){

string s;

cin>>s;

cout<<solve(s);

}