**Palindrome Partitioning ll**

#include <bits/stdc++.h>

using namespace std;

bool isPalindrome(int i, int j, string &s) {

while (i < j) {

if (s[i] != s[j]) return false;

i++;

j--;

}

return true;

}

int f(int i, int n, string &str, vector<int> &dp) {

//Base case:

if (i == n) return 0;

if (dp[i] != -1) return dp[i];

int minCost = INT\_MAX;

//string[i...j]

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

if (isPalindrome(i, j, str)) {

int cost = 1 + f(j + 1, n, str, dp);

minCost = min(minCost, cost);

}

}

return dp[i] = minCost;

}

int palindromePartitioning(string str) {

// Write your code here

int n = str.size();

vector<int> dp(n, -1);

return f(0, n, str, dp) - 1;

}