**Palindrome Partitioning**

#include <bits/stdc++.h>

bool isPalindrome(string &str, int start, int end) {

while (start <= end) {

if (str[start++] != str[end--])

return false;

}

return true;

}

void partitionHelper(int index, vector<vector<string>> &result, vector<string> &currentPartition, string &str) {

if (index == str.size()) {

result.push\_back(currentPartition);

return;

}

for (int i = index; i < str.size(); i++) {

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

currentPartition.push\_back(str.substr(index, i - index + 1));

partitionHelper(i + 1, result, currentPartition, str);

currentPartition.pop\_back();

}

}

}

vector<vector<string>> partition(string &str) {

vector<vector<string>> result;

vector<string> currentPartition;

int index = 0;

partitionHelper(index, result, currentPartition, str);

return result;

}