class Solution {

public:

std::string longestPalindrome(std::string s) {

if (s.length() <= 1) {

return s;

}

int max\_len = 1;

int start = 0;

int end = 0;

std::vector<std::vector<bool>> dp(s.length(), std::vector<bool>(s.length(), false));

for (int i = 0; i < s.length(); ++i) {

dp[i][i] = true;

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

if (s[j] == s[i] && (i - j <= 2 || dp[j + 1][i - 1])) {

dp[j][i] = true;

if (i - j + 1 > max\_len) {

max\_len = i - j + 1;

start = j;

end = i;

}

}

}

}

return s.substr(start, end - start + 1);

}

};