**Longest Palindromic Substring**

string expandAroundCenter(string str, int left, int right)

{

int length = str.length();

while (left >= 0 && right < length && str[left] == str[right]) {

left--;

right++;

}

return str.substr(left + 1, right - left - 1);

}

string longestPalinSubstring(string str)

{

int length = str.length();

if (length < 2)

return str;

string longestPalindrome = "";

string currentPalindrome;

for (int i = 0; i < length - 1; i++) {

currentPalindrome = expandAroundCenter(str, i, i);

if (currentPalindrome.length() > longestPalindrome.length())

longestPalindrome = currentPalindrome;

currentPalindrome = expandAroundCenter(str, i, i + 1);

if (currentPalindrome.length() > longestPalindrome.length())

longestPalindrome = currentPalindrome;

}

return longestPalindrome;

}