```
/* SPOJ - NUMOFPAL - Number of Palindromes
        Given a string S. Find total Number of palindromic substring of S.
        The length of S, |S| will not more than 10000. */
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 10005;
struct Node{
    int nxt[26];
    int length, suffixLink;
    int startPos, endPos;
    int val;
};
struct PalTree{
    Node tree[MAXN];
    Node root1, root2;
    int ptr, curNode;
    char s[MAXN];

    void init(){
        root1.length = -1, root1.suffixLink = 1;
        root2.length =  0, root2.suffixLink = 1;
        tree[1] = root1, tree[2] = root2;
        ptr = curNode = 2;
    }

    void addLetter(int pos){
        int ch = s[pos]-'a';
        int cur = curNode;

        while(true){
            int curLength = tree[cur].length;
            if(pos-1-curLength >= 0 && s[pos-1-curLength] == s[pos])break;
            cur = tree[cur].suffixLink;
        }

        if(tree[cur].nxt[ch] != 0){
            curNode = tree[cur].nxt[ch];
            tree[curNode].val++;
            return;
        }

        ptr++;
        curNode = ptr;
        tree[cur].nxt[ch] = curNode;
        tree[curNode].length = tree[cur].length + 2;
        tree[curNode].startPos = pos - tree[curNode].length + 1;
        tree[curNode].endPos = pos;

        if(tree[curNode].length == 1){
            tree[curNode].suffixLink = 2;
            tree[curNode].val = 1;
            return;
        }
```

```
54.
55.         while(true){
56.             cur = tree[cur].suffixLink;
57.             int curLength = tree[cur].length;
58.             if(pos-1-curLength >= 0 && s[pos-1-curLength] == s[pos]){
59.                 tree[curNode].suffixLink = tree[cur].nxt[ch];
60.                 break;
61.             }
62.         }
63.
64.         tree[curNode].suffixLink = tree[cur].nxt[ch];
65.         tree[curNode].val = 1;
66.         return;
67.     }
68.
69.     int getResult(){
70.         int ans = 0;
71.         for(int i=ptr; i>=3; i--){
72.             ans += tree[i].val;
73.             tree[tree[i].suffixLink].val += tree[i].val;
74.         }
75.         return ans;
76.     }
77.
78.     void Clear(){
79.         for(int i=0; i<=ptr; i++){
80.             memset(tree[i].nxt, 0, sizeof(tree[i].nxt));
81.         }
82.     }
83. };
84. PalTree Pt;
85. int main(){
86.     scanf("%s",&Pt.s);
87.     int n = strlen(Pt.s);
88.     Pt.init();
89.     for(int i=0; i<n; i++) Pt.addLetter(i);
90.     int ans = Pt.getResult();
91.     printf("%d\n",ans);
92.     return 0;
93. }
94. Input                   Output
95. aaaaa                   15
96. malayalam               15
97. ababab                  12
```