

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4
5  Manacher's Algorithm for string:
6  struct Manacher{
7      int n;
8      string s;
9      vector<int> d1, d2;
10
11      Manacher(string s) : s(s)
12      {
13          n = s.size();
14          d1.resize(n);
15          d2.resize(n);
16          // d1
17          for (int i = 0, l = 0, r = -1; i < n; i++)
18          {
19              int k = (i > r) ? 1 : min(d1[l + r - i], r - i + 1);
20              while (0 ≤ i - k && i + k < n && s[i - k] == s[i + k])
21              {
22                  k++;
23              }
24              d1[i] = k--;
25              if (i + k > r)
26              {
27                  l = i - k;
28                  r = i + k;
29              }
30          }
31          // d2
32          for (int i = 0, l = 0, r = -1; i < n; i++)
33          {
34              int k = (i > r) ? 0 : min(d2[l + r - i + 1], r - i + 1);
35              while (0 ≤ i - k - 1 && i + k < n && s[i - k - 1] == s[i + k])
36              {
37                  k++;
38              }
39              d2[i] = k--;
40              if (i + k > r)
41              {
42                  l = i - k - 1;
43                  r = i + k;
44              }
45          }
46      }
47
48      //check if subString is palindrome O(1)
49      bool isPal(int l, int r)
50      {
51          int len = r - l + 1;
52          int i = l + r >> 1;
53          if (len % 2)
54              return d1[i] > len / 2;
55          else
56              return d2[i + 1] ≥ len / 2;
57      }
58
59      //get the number of palindrome subString in string O(n)
60      ll numberOfPal(){
61          ll even = accumulate(d1.begin(), d1.end(), 0LL);
62          ll odd = accumulate(d2.begin(), d2.end(), 0LL);

```