**Manacher, O(N)**

**Tested on None (from Geeksforgeeks.org)**

char text[100];

void findLongestPalindromicString()

{

int N = strlen(text);

if(N == 0) return;

N = 2\*N + 1; //Position count

int L[N], C=1, R=2, i=0, iMirror, maxLPSLength=0, maxLPSCenterPosition=0, start=-1, end=-1, diff=-1 ;

L[0] = 0, L[1] = 1;

for (i = 2; i < N; i++)

{

iMirror = 2\*C-i, L[i] = 0, diff = R - i;

if(diff > 0) L[i] = min(L[iMirror], diff); //If currentRightPosition i is within centerRightPosition R

while ( ((i+L[i])<N && (i-L[i])>0) && ( ((i+L[i]+1)%2==0) || (text[(i+L[i]+1)/2]==text[(i-L[i]-1)/2])))

L[i]++;

if(L[i] > maxLPSLength) maxLPSLength = L[i], maxLPSCenterPosition = i;

if (i + L[i] > R) C = i, R = i + L[i];

//printf("%d ", L[i]);

}

start = (maxLPSCenterPosition - maxLPSLength)/2;

end = start + maxLPSLength - 1;

printf("LPS of string %s is: ", text);

for(i=start; i<=end; i++) printf("%c", text[i]);

printf("\n");

}