Algorithm: Longest Palindrome Substring iman.irajian@gmail.com

Input: A string, say S

Output: Longest Palindrome Substring, say `result`

Let $P(i,j) = \begin{cases} \text{true}, & \text{if the substring } S_i \dots S_j \text{ is a palindrome} \\ \text{false}, & \text{otherwise} \end{cases}$

We have $\boldsymbol{P(i,j) = \left(P(i+1, j-1) \text{ and } S_i == S_j\right)}$

Base case 1: $P(i,i) = \text{true}$

Base case 2: $P(i,i+1) = (S_i == S_{i+1})$

```
1   n ← len(S)                                  Length of input string
2   table ← 0_{n×n}                             Boolean table for dynamic programming
3   start ← 0                                   Result starting index
4
5   maxLen ← 1                                  First consider substrings of length 1
6   for i ← 0 to n − 1
7      table[i][i] ← 1
8
9   for i ← 0 to n − 2                          Check for substrings of length 2
10     if (S[i]==S[i + 1])
11        table[i][i + 1] ← 1
12        start ← i
13        maxLen ← 2
14
15  for k ← 3 to n                              Check for substrings of length k > 2
16     for i ← 0 to n − k + 1                   (i) For starting index
17        j ← i + k − 1                         (j) For ending index
18        if (table[i + 1][j − 1] and S[i]==S[j])
19           table[i][j] ← 1
20           if (k > maxLen)
21              start ← i
22              maxLen ← k
23
24  result ← S[start … start+maxLen-1]
```

Complexity	
Time	**Space**
$O(n^2)$	$O(n^2)$

■