TT DS PYTHON MODULE-22

Started on	Saturday, 3 May 2025, 1:46 PM
State	Finished
Completed on	Saturday, 3 May 2025, 1:49 PM
Time taken	3 mins 1 sec
Grade	**10.00** out of 10.00 (**100**%)

LONGEST PALINDROMIC SUBSEQUENCE

Given a sequence, find the length of the longest palindromic subsequence in it.

For example:

Input	Result
ABBDCACB	The length of the LPS is 5

Answer: (penalty regime: 0 %)

```python
dp = [[-1 for i in range(1001)]for j in range(1001)]
def lps(s1, s2, n1, n2):
    if (n1 == 0 or n2 == 0):
        return 0
    if (dp[n1][n2] != -1):
        return dp[n1][n2]
    if (s1[n1 - 1] == s2[n2 - 1]):
        dp[n1][n2] = 1 + lps(s1, s2, n1 - 1, n2 - 1)
        return dp[n1][n2]
    else:
        dp[n1][n2] = max(lps(s1, s2, n1 - 1, n2), lps(s1, s2, n1, n2 - 1))
        return dp[n1][n2]
seq = input()
n = len(seq)
s2 = seq
s2 = s2[::-1]
print(f"The length of the LPS is",lps(s2, seq, n, n))
```

	Input	Expected	Got	
	ABBDCACB	The length of the LPS is 5	The length of the LPS is 5	
	BBABCBCAB	The length of the LPS is 7	The length of the LPS is 7	
	cbbd	The length of the LPS is 2	The length of the LPS is 2	
	abbab	The length of the LPS is 4	The length of the LPS is 4	

Passed all tests!

Correct

Marks for this submission: 2.50/2.50.

Given a string s, return *the longest palindromic substring* in s.

Example 1:

Input: s = "babad"
Output: "bab"

Explanation: "aba" is also a valid answer.

Example 2:

```
Input: s = "cbbd"
Output: "bb"
```

For example:

Test	Input	Result
ob1.longestPalindrome(str1)	ABCBCB	BCBCB

Answer: (penalty regime: 0 %)

Reset answer

```python
1  class Solution(object):
2      def longestPalindrome(self, s):
3          dp = [[False for i in range(len(s))] for i in range(len(s))]
4          for i in range(len(s)):
5              dp[i][i] = True
6          max_length = 1
7          start = 0
8          for l in range(2,len(s)+1):
9              for i in range(len(s)-l+1):
10                 end = i+l
11                 if l==2:
12                     if s[i] == s[end-1]:
13                         dp[i][end-1]=True
14                         max_length = l
15                         start = i
16                 else:
17                     if s[i] == s[end-1] and dp[i+1][end-2]:
18                         dp[i][end-1]=True
19                         max_length = l
20                         start = i
21          return s[start:start+max_length]
22
```

	Test	Input	Expected	Got	
	ob1.longestPalindrome(str1)	ABCBCB	BCBCB	BCBCB	
	ob1.longestPalindrome(str1)	BABAD	ABA	ABA	

Passed all tests!

Correct

Marks for this submission: 2.50/2.50.

Create a python program to find the longest palindromic substring using Brute force method in a given string.

For example:

Input	Result
mojologiccigolmojo	logiccigol

Answer: (penalty regime: 0 %)

Reset answer

```python
1  def printSubStr(str, low, high):
2      for i in range(low, high + 1):
3          print(str[i], end = "")
4  def longestPalindrome(str):
5      n = len(str)
6      maxLength = 1
```

```
 7        start = 0
 8        for i in range(n):
 9            for j in range(i, n):
10                flag = 1
11                for k in range(0, ((j - i) // 2) + 1):
12                    if (str[i + k] != str[j - k]):
13                        flag = 0
14                if (flag != 0 and (j - i + 1) > maxLength):
15                    start = i
16                    maxLength = j - i + 1
17        printSubStr(str, start, start + maxLength - 1)
18 str = input() #"mojologiccigolmojo"
19 longestPalindrome(str)
20
```

	Input	Expected	Got	
	mojologiccigolmojo	logiccigol	logiccigol	
	sampleelpams	pleelp	pleelp	

Passed all tests!

Marks for this submission: 2.50/2.50.

Create a python program to find the longest palindromic substring using optimal algorithm Expand around center.

For example:

Test	Input	Result
findLongestPalindromicSubstring(s)	samsunggnusgnusam	sunggnus

Answer: (penalty regime: 0 %)

Reset answer

```
 1 def printSubStr(ss, low, high):
 2     for i in range(low, high + 1):
 3         print(s[i], end = "")
 4 def findLongestPalindromicSubstring(s):
 5     n = len(s)
 6     maxLength = 1
 7     start = 0
 8     for i in range(n):
 9         for j in range(i, n):
10             flag = 1
11             for k in range(0, ((j - i) // 2) + 1):
12                 if (s[i + k] != s[j - k]):
13                     flag = 0
14             if (flag != 0 and (j - i + 1) > maxLength):
15                 start = i
16                 maxLength = j - i + 1
17     printSubStr(s, start, start + maxLength - 1)
18 s = input()
19
20
```

	Test	Input	Expected	Got	
	findLongestPalindromicSubstring(s)	samsunggnusgnusam	sunggnus	sunggnus	
	findLongestPalindromicSubstring(s)	welcomeindiaaidni	indiaaidni	indiaaidni	

Passed all tests!

Correct

Marks for this submission: 2.50/2.50.

Quiz navigation

1 ✓ 2 ✓ 3 ✓ 4 ✓