Started on	Friday, 16 May 2025, 8:43 AM
State	Finished
Completed on	Friday, 16 May 2025, 9:45 AM
Time taken	1 hour 1 min
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

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

```
def expand(s, low, high):
    length = len(s)

    while low >= 0 and high < length and s[low] == s[high]:
        low = low - 1
        high = high + 1

    return s[low + 1:high]


def findLongestPalindromicSubstring(s):

    if not s or not len(s):
        return ''

    longest_palindrome = ""

    # Iterate through the string
    for i in range(len(s)):
        odd_palindrome = expand(s, i, i)
        even_palindrome = expand(s, i, i + 1)

```

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

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create a python program to find the length of longest common subsequence using naive recursive method

For example:

Input	Result
AGGTAB GXTXAYB	Length of LCS is 4

Answer: (penalty regime: 0 %)

```python
def lcs(X, Y, m, n):
    if m == 0 or n == 0:
        return 0
    elif X[m - 1] == Y[n - 1]:
        return 1 + lcs(X, Y, m - 1, n - 1)
    else:
        return max(lcs(X, Y, m, n - 1), lcs(X, Y, m - 1, n))

X = input()
Y = input()

result = lcs(X, Y, len(X), len(Y))

print("Length of LCS is ", result)

```

	Input	Expected	Got	
✓	AGGTAB GXTXAYB	Length of LCS is 4	Length of LCS is 4	✓
✓	saveetha engineering	Length of LCS is 2	Length of LCS is 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Create a python program to compute the edit distance between two given strings using iterative method.

For example:

Input	Result
kitten sitting	3

Answer: (penalty regime: 0 %)

```python
def LD(s, t):
    if len(s)==0:
        return len(t)
    if len(t)==0:
        return len(s)
    if s[-1]==t[-1]:
        return LD(s[:-1],t[:-1])
    else:
        insert=LD(s,t[:-1])
        delete=LD(s[:-1],t)
        replace=LD(s[:-1],t[:-1])

        return 1+min(insert,delete,replace)

str1=input()
str2=input()
print(LD(str1,str2))

```

	Input	Expected	Got	
✓	kitten sitting	3	3	✓
✓	medium median	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Not answered

Mark 0.00 out of 20.00

Write a Python Program Using a recursive function to calculate the sum of a sequence
For example:

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

Question **5**

Correct

Mark 20.00 out of 20.00

Create a Python program to find longest common substring or subword (LCW) of two strings using dynamic programming with bottom-up approach.

A string r is a substring or subword of a string s if r is contained within s. A string r is a common substring of s and t if r is a substring of both s and t. A string r is a longest common substring or subword (LCW) of s and t if there is no string that is longer than r and is a common substring of s and t. The problem is to find an LCW of two given strings.

For example:

Test	Input	Result
lcw(u, v)	bisect trisect	Longest Common Subword: isect

Answer: (penalty regime: 0 %)

Reset answer

```python
def lcw(u, v):
    m = len(u)
    n = len(v)
    dp = [[0] * (n + 1) for _ in range(m + 1)]

    max_len = 0
    end_index_u = 0

    # Fill the table
    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if u[i - 1] == v[j - 1]:
                dp[i][j] = dp[i - 1][j - 1] + 1
                if dp[i][j] > max_len:
                    max_len = dp[i][j]
                    end_index_u = i - max_len
            else:
                dp[i][j] = 0
    return max_len, end_index_u, 0


u = input()
```

	Test	Input	Expected	Got	
✓	lcw(u, v)	bisect trisect	Longest Common Subword: isect	Longest Common Subword: isect	✓
✓	lcw(u, v)	director conductor	Longest Common Subword: ctor	Longest Common Subword: ctor	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.