Started on	Thursday, 30 January 2025, 3:18 PM
State	Finished
Completed on	Thursday, 30 January 2025, 4:02 PM
Time taken	44 mins 13 secs
Grade	**80.00** out of 100.00

Question **1**

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 top-down approach or memoization.

Problem Description

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)	potato tomato	Longest Common Subword: ato

Answer: (penalty regime: 0 %)

Reset answer

```
def lcw(u, v):
    c = [[-1]*(len(v) + 1) for _ in range(len(u) + 1)]
    lcw_i = lcw_j = -1
    length_lcw = 0
    for i in range(len(u)):
        for j in range(len(v)):
            temp = lcw_starting_at(u, v, c, i, j)
            if length_lcw < temp:
                length_lcw = temp
                lcw_i = i
                lcw_j = j
    return length_lcw, lcw_i, lcw_j
def lcw_starting_at(u, v, c, i, j):
    if c[i][j] >= 0:
        return c[i][j]
    if i == len(u) or j == len(v):
        q = 0
    elif u[i] != v[j]:
        q = 0
    else:
        q = 1 + lcw_starting_at(u, v, c, i + 1, j + 1)
    c[i][j] = q
```

	Test	Input	Expected	Got	
✔	lcw(u, v)	potato tomato	Longest Common Subword: ato	Longest Common Subword: ato	✔
✔	lcw(u, v)	snakegourd bottlegourd	Longest Common Subword: egourd	Longest Common Subword: egourd	✔

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 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 ''
    max_str = ''
    max_length = 0
    for i in range(len(s)):
        curr_str = expand(s, i, i)
        curr_length = len(curr_str)

        if curr_length > max_length:
            max_length = curr_length
            max_str = curr_str
        curr_str = expand(s, i, i + 1)
        curr_length = len(curr_str)

        if curr_length > max_length:
```

	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 **3**

Correct

Mark 20.00 out of 20.00

Create a python program to find the longest common subsequence using Memoization Implementation.

For example:

Input	Result
AGGTAB GXTXAYB	Length of LCS is 4

Answer: (penalty regime: 0 %)

```
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()#"AGGTAB"
Y = input()#"GXTXAYB"
print ("Length of LCS is", lcs(X , Y, len(X), len(Y)) )
```

	Input	Expected	Got	
✔	AGGTAB GXTXAYB	Length of LCS is 4	Length of LCS is 4	✔
✔	SAMPLE SAEMSUNG	Length of LCS is 3	Length of LCS is 3	✔
✔	saveetha sabeetha	Length of LCS is 7	Length of LCS is 7	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Create a python program to find the Edit distance between two strings using dynamic programming.

For example:

Input	Result
Cats Rats	No. of Operations required : 1

Answer: (penalty regime: 0 %)

Reset answer

```
def edit_distance(str1, str2, a, b):
    string_matrix = [[0 for i in range(b+1)] for i in range(a+1)]
    for i in range(a+1):
        for j in range(b+1):
            if i == 0:
                string_matrix[i][j] = j
            elif j == 0:
                string_matrix[i][j] = i
            elif str1[i-1] == str2[j-1]:
                string_matrix[i][j] = string_matrix[i-1][j-1]
            else:
                string_matrix[i][j] = 1 + min(string_matrix[i][j-1],string_matrix[i-1][j],string_matr
    return string_matrix[a][b]
if __name__ == '__main__':
    str1 = input()
    str2 = input()
    print('No. of Operations required :',edit_distance(str1, str2, len(str1), len(str2)))
```

	Input	Expected	Got	
✔	Cats Rats	No. of Operations required : 1	No. of Operations required : 1	✔
✔	Saturday Sunday	No. of Operations required : 3	No. of Operations required : 3	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Not answered

Mark 0.00 out of 20.00

Write a Python Program to print factorial of a number recursively.

For example:

Input	Result
5	Factorial of number 5 = 120
6	Factorial of number 6 = 720

Answer: (penalty regime: 0 %)

```
1
```

Syntax Error(s)

```
  File "__tester__.python3", line 4
    print("Factorial of number= " fact);
                                  ^

SyntaxError: invalid syntax
```

Incorrect

Marks for this submission: 0.00/20.00.