Started on	Friday, 14 March 2025, 1:13 PM
State	Finished
Completed on	Friday, 14 March 2025, 2:02 PM
Time taken	48 mins 50 secs
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 printSubStr(ss, low, high):
    for i in range(low, high + 1):
        print(s[i], end = "")
def findLongestPalindromicSubstring(s):
    n = len(s)
    maxLength = 1
    start = 0
    for i in range(n):
        for j in range(i, n):
            flag = 1
            for k in range(0, ((j - i) // 2) + 1):
                if (s[i + k] != s[j - k]):
                    flag = 0
            if (flag != 0 and (j - i + 1) > maxLength):
                start = i
                maxLength = j - i + 1
    printSubStr(s, start, start + maxLength - 1)
s = input()
```

	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 length of longest common subsequence using naive recursive method

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()
y=input()
print(f"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	✔
✔	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 **4**

Correct

Mark 20.00 out of 20.00

LONGEST COMMON SUBSTRING PROBLEM

The longest common substring problem is the problem of finding the longest string (or strings) that is a substring (or are substrings) of two strings.

Answer: (penalty regime: 0 %)

```
def LCS(X, Y, m, n):
    maxLength = 0
    endingIndex = m
    lookup = [[0 for x in range(n + 1)] for y in range(m + 1)]
    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if X[i - 1] == Y[j - 1]:
                lookup[i][j] = lookup[i - 1][j - 1] + 1
                if lookup[i][j] > maxLength:
                    maxLength = lookup[i][j]
                    endingIndex = i
    return X[endingIndex - maxLength: endingIndex]

X = input()
Y = input()
m = len(X)
n = len(Y)
sub=LCS(X, Y, m, n)
print("The longest common substring is",sub)

```

	Input	Expected	Got	
✔	ABC BABA	The longest common substring is AB	The longest common substring is AB	✔
✔	abcdxyz xyzabcd	The longest common substring is abcd	The longest common substring is abcd	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Create a Naive recursive python program to find the minimum number of operations to convert str1 to str2

For example:

Input	Result
Python Peithen	Edit Distance 3

Answer: (penalty regime: 0 %)

Reset answer

```
def LD(s, t):
    if s=="":
        return len(t)
    if t=="":
        return len(s)
    if s[-1]==t[-1]:
        cost=0
    else:
        cost=1
    res=min([LD(s[:-1],t)+1,LD(s,t[:-1])+1,LD(s[:-1],t[:-1])+cost])
    return res

str1=input()
str2=input()
print('Edit Distance',LD(str1,str2))


```

	Input	Expected	Got	
✔	Python Peithen	Edit Distance 3	Edit Distance 3	✔
✔	food money	Edit Distance 4	Edit Distance 4	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.