

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

```

1 def lcw(X,Y):
2     m = len(X)
3     n = len(Y)
4     maxLength = 0
5     endingIndex = m
6     lookup = [[0 for x in range(n + 1)] for y in range(m + 1)]
7     for i in range(1, m + 1):
8         for j in range(1, n + 1):
9             if X[i - 1] == Y[j - 1]:
10                lookup[i][j] = lookup[i - 1][j - 1] + 1
11                if lookup[i][j] > maxLength:
12                    maxLength = lookup[i][j]
13                    endingIndex = i
14     return X[endingIndex - maxLength: endingIndex]
15
16 u = input()
17 v = input()
18 print("Longest Common Subword:", lcw(u,v))
19
20

```

	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

Flag question

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

```

1 def printSubStr(str, low, high):
2
3     for i in range(low, high + 1):
4         print(str[i], end = "")
5
6 def longestPalindrome(str):
7     ##### Add your code here #####
8     n = len(str)
9     maxLength = 1
10    start = 0
11    for i in range(n):
12        for j in range(i, n):

```

```

13         flag = 1
14         for k in range(0, ((j - i) // 2) + 1):
15             if (str[i + k] != str[j - k]):
16                 flag = 0
17         if (flag != 0 and (j - i + 1) > maxLength):
18             start = i
19             maxLength = j - i + 1
20         printSubStr(str, start, start + maxLength - 1)
21
22 if __name__ == '__main__':

```

	Input	Expected	Got	
	mojologiccigolmojo	logiccigol	logiccigol	
	sampleelpams	pleelp	pleelp	

Passed all tests!

**Correct**

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

🚩 Flag question

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 %)

```

1 def LD(s, t):
2     if s == "":
3         return len(t)
4     if t == "":
5         return len(s)
6     if s[-1] == t[-1]:
7         cost = 0
8     else:
9         cost = 1
10    res = min([LD(s[:-1], t)+1,
11              LD(s, t[:-1])+1,
12              LD(s[:-1], t[:-1]) + cost])
13    return res
14
15 str1=input()
16 str2=input()
17 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**

Incorrect

Mark 0.00 out of 20.00

🚩 Flag question

Write a python to implement Quick sort using the first element as pivot value

**For example:**

Input	Result
5 61 24	Pivot: 61 Pivot: 8

Input	Result
3 50 8	Pivot: 24 Sorted array: [3, 8, 24, 50, 61]
6 2 3 54 10 28 94	Pivot: 2 Pivot: 3 Pivot: 54 Pivot: 28 Sorted array: [2, 3, 10, 28, 54, 94]

**Answer:** (penalty regime: 0 %)

```

1 def quick_sort(arr):
2     if len(arr) <= 1:
3         return arr
4     pivot = arr[0]
5     less = [x for x in arr[1:] if x <= pivot]
6     greater = [x for x in arr[1:] if x > pivot]
7     print("Pivot: ", pivot)
8     return quick_sort(less) + [pivot] + quick_sort(greater)
9
10 n = int(input())
11 arr = [int(input()) for _ in range(n)]
12 sorted_arr = quick_sort(arr)
13 print("Sorted array:", sorted_arr)

```

	Input	Expected	Got	
	5 61 24 3 50 8	Pivot: 61 Pivot: 8 Pivot: 24 Sorted array: [3, 8, 24, 50, 61]	Pivot: 61 Pivot: 24 Pivot: 3 Sorted array: [3, 8, 24, 50, 61]	

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

**Incorrect**

Marks for this submission: 0.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Flag question

Write a Python Program to find longest common subsequence using Dynamic Programming

**Answer:** (penalty regime: 0 %)

```

1 def lcs(str1, str2):
2     m = len(str1)
3     n = len(str2)
4     matrix = [[0]*(n+1) for i in range(m+1)]
5     for i in range(m+1):
6         for j in range(n+1):
7             if i==0 or j==0:
8                 matrix[i][j] = 0
9             elif str1[i-1] == str2[j-1]:
10                matrix[i][j] = 1 + matrix[i-1][j-1]
11            else:
12                matrix[i][j] = max(matrix[i-1][j], matrix[i][j-1])
13    return matrix[-1][-1]
14 str1 = input()
15 str2 = input()
16 lcs_length = lcs(str1, str2)
17 print("Length of LCS is : {}".format(lcs_length))

```

	Input	Expected	Got	
	abcbdbab bdcaba	Length of LCS is : 4	Length of LCS is : 4	
	treehouse elephant	Length of LCS is : 3	Length of LCS is : 3	
	AGGTAB GXTXAYB	Length of LCS is : 4	Length of LCS is : 4	

Passed all tests!

**Correct**

Marks for this submission: 20.00/20.00.

//