

Started on Tuesday, 6 May 2025, 9:23 AM**State** Finished**Completed on** Tuesday, 6 May 2025, 8:44 PM**Time taken** 11 hours 21 mins**Overdue** 9 hours 21 mins**Grade** 80.00 out of 100.00Question **1**

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 GTXAYB	Length of LCS is 4

Answer: (penalty regime: 0 %)

```

1 def lcs(x,y,m,n):
2     if m==0 or n==0:
3         return 0
4     elif x[m-1]==y[n-1]:
5         return 1+lcs(x,y,m-1,n-1)
6     else:
7         return max(lcs(x,y,m,n-1),lcs(x,y,m-1,n))
8 X = input()
9 Y = input()
10 print ("Length of LCS is ", lcs(X , Y, len(X), len(Y)) )

```

	Input	Expected	Got	
✓	AGGTAB GTXAYB	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 **2**

Correct

Mark 20.00 out of 20.00

LONGEST COMMON SUBSTRING PROBLEM

Given two strings 'X' and 'Y', find the length of the longest common substring.

Answer: (penalty regime: 0 %)

```

1 def LongComSubS(st1, st2):
2     ans = 0;
3     for a in range(len(st1)):
4         for b in range(len(st2)):
5             k = 0;
6             while ((a + k) < len(st1) and (b + k) < len(st2)
7                 and st1[a + k] == st2[b + k]):
8                 k = k + 1;
9             ans = max(ans, k);
10    return ans;
11
12 if __name__ == '__main__':
13
14     A = input()
15     B = input()
16     i = len(A)
17     j = len(B)
18     print('Length of Longest Common Substring is', LongComSubS(A, B))

```

	Input	Expected	Got	
✓	ABC BABA	Length of Longest Common Substring is 2	Length of Longest Common Substring is 2	✓
✓	abcdxyz xyzabcd	Length of Longest Common Substring is 4	Length of Longest Common Substring is 4	✓

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 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     n=len(str)
8     max_len=0
9     start=0
10    for i in range(n):
11        for j in range(1,n):
12            s=str[i:j+1]
13            if s==s[::-1]:
14                cur=j-i+1
15                if cur>max_len:
16                    max_len=cur
17                    start=i
18    printSubStr(str, start, start + max_len - 1)
19
20 if __name__ == '__main__':
21
22     str = input()

```

	Input	Expected	Got	
✓	mojologiccigolmojo	logiccigol	logiccigol	✓
✓	sampleelpams	pleelp	pleelp	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 4

Correct

Mark 20.00 out of 20.00

Write a python program to check whether Hamiltonian path exists in the given graph.

For example:

Test	Result
Hamiltonian_path(adj, N)	YES

Answer: (penalty regime: 0 %)

Reset answer

```

1 def is_valid(v,pos,path,adj,N):
2     if adj[path[pos-1]][v]==0:
3         return False
4     if v in path:
5         return False
6     return True
7 def hamUtil(adj,path,pos,N):
8     if pos==N:
9         return True
10    for v in range(N):
11        if is_valid(v,pos,path,adj,N):
12            path[pos]=v
13            if hamUtil(adj,path,pos+1,N):
14                return True
15            path[pos]=-1
16    return True
17 def Hamiltonian_path(adj,N):
18     path=[-1]*N
19     path[0]=0
20
21     if hamUtil(adj,path,1,N) == False:
22         print ("Solution does not exist\n")

```

	Test	Expected	Got	
✓	Hamiltonian_path(adj, N)	YES	YES	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Incorrect

Mark 0.00 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

```
1 | def edit_distance(str1, str2, a, b):  
2 |     ##### Add your code here #####  
3 | if __name__ == '__main__':  
4 |     str1 = input()  
5 |     str2 = input()  
6 |     print('No. of Operations required :',edit_distance(str1, str2, len(str1), len(str2)))
```

Syntax Error(s)

Sorry: IndentationError: expected an indented block (__tester__.python3, line 3)

Incorrect

Marks for this submission: 0.00/20.00.