

INSTITUTE OF COMPUTER TECHNOLOGY  
B-TECH COMPUTER SCIENCE ENGINEERING 2025-26  
SUBJECT:-Algorithm Analysis and Design

NAME: Rahul Prajapati

ENRLL NO: 23162171020

BRANCH: CYBER SECURITY

BATCH: 52

PRACTICAL\_7

**Aim:** A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. Longest common subsequence (LCS) of 2 sequences is a subsequence, with maximal length, which is common to both the sequences.

Given two sequences of integers,  $P = \langle M, N, O, M \rangle$  and  $Q = \langle M, L, N, O, M \rangle$ , find any one longest common subsequence.

In case multiple solutions exist, print any of them. It is guaranteed that at least one non-empty common subsequence will exist.

## CODE:

```
server_socket.py Practical7_1.py X
Practical7_1.py > lcs
1  def lcs(P, Q):
2      m, n = len(P), len(Q)
3      dp = [[0] * (n + 1) for _ in range(m + 1)]
4
5      for i in range(1, m + 1):
6          for j in range(1, n + 1):
7              if P[i-1] == Q[j-1]:
8                  dp[i][j] = dp[i-1][j-1] + 1
9              else:
10                 dp[i][j] = max(dp[i-1][j], dp[i][j-1])
11
12     lcs_seq = []
13     i, j = m, n
14
15     while i > 0 and j > 0:
16         if P[i-1] == Q[j-1]:
17             lcs_seq.append(P[i-1])
18             i -= 1
19             j -= 1
20         elif dp[i-1][j] > dp[i][j-1]:
21             i -= 1
22         else:
23             j -= 1
24
25     lcs_seq.reverse()
26     return lcs_seq
27
28     P = ['M', 'N', 'o', 'M']
29     Q = ['M', 'L', 'N', 'o', 'M']
30
31     result = lcs(P, Q)
32
33     print("Longest Common Subsequence:", result)
```

## OUTPUT:

```
PS C:\Users\Hp\OneDrive\Desktop\SEM_05\Algorithm Analysis & Design\SOURCE_CODES> python .\Practical7_1.py
Longest Common Subsequence: ['M', 'N', 'M']
PS C:\Users\Hp\OneDrive\Desktop\SEM_05\Algorithm Analysis & Design\SOURCE_CODES>
```