Name: Devanshi Jain
Er-No:22162101006
Batch: 51
Branch: CBA

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 8

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.

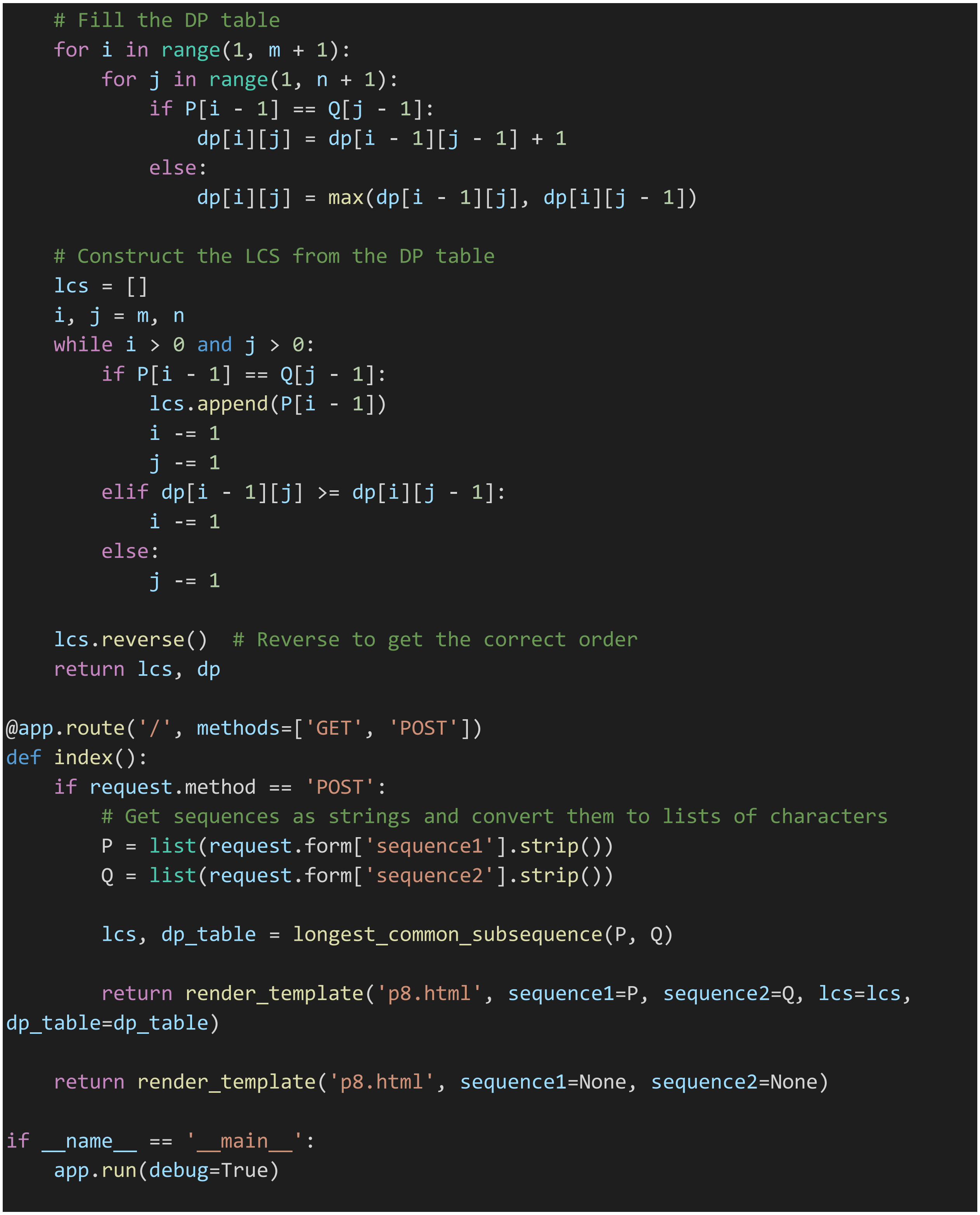
Given two sequences of integers, P = <M, N, O, M> and Q = <M, L, N, O, M>, find any one longest common subsequence.

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

CODE:

.py:

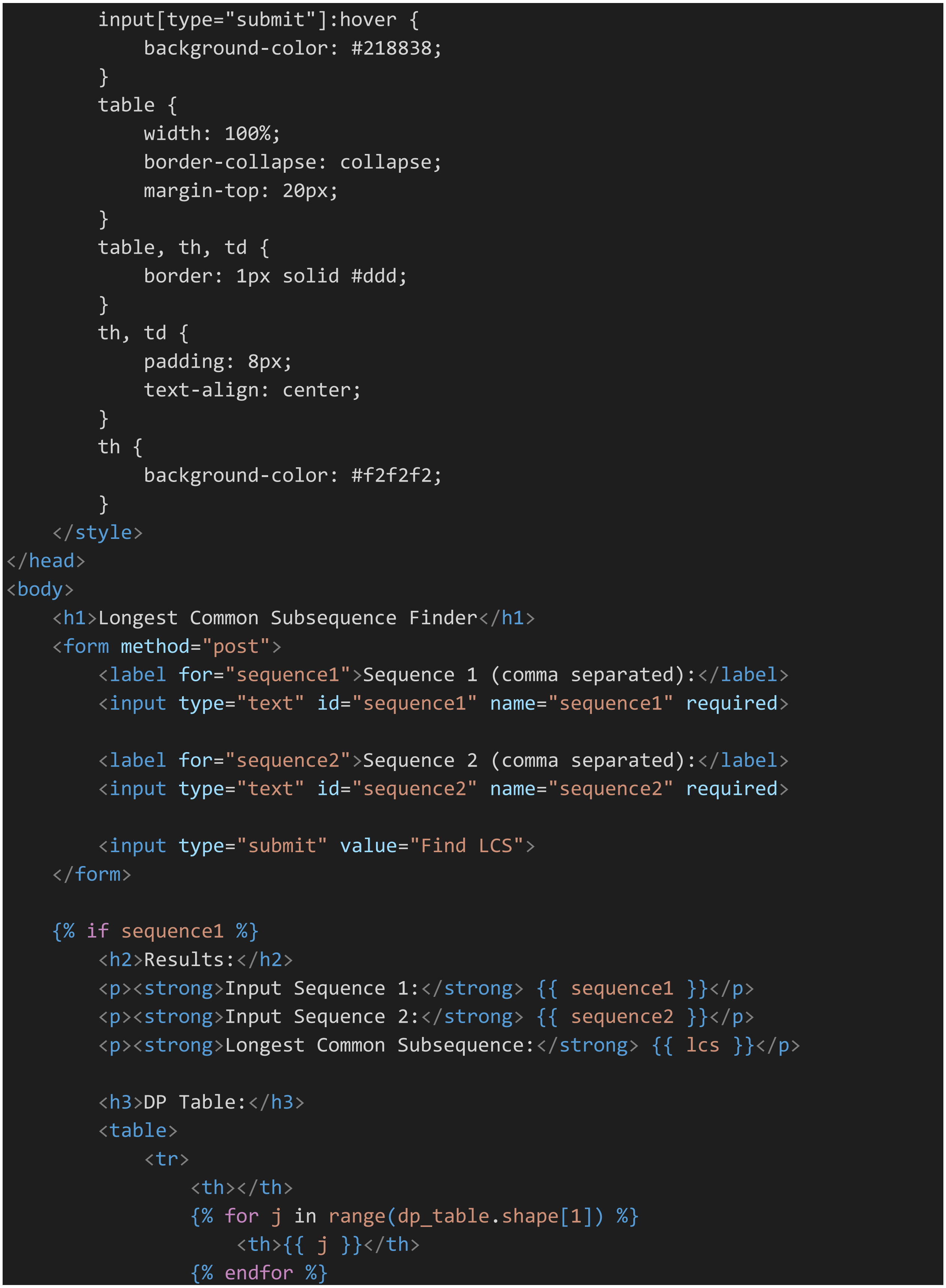
```python
from flask import Flask, render_template, request
import numpy as np

app = Flask(__name__)

def longest_common_subsequence(P, Q):
    m = len(P)
    n = len(Q)

    # Create a DP table
    dp = np.zeros((m + 1, n + 1), dtype=int)
```



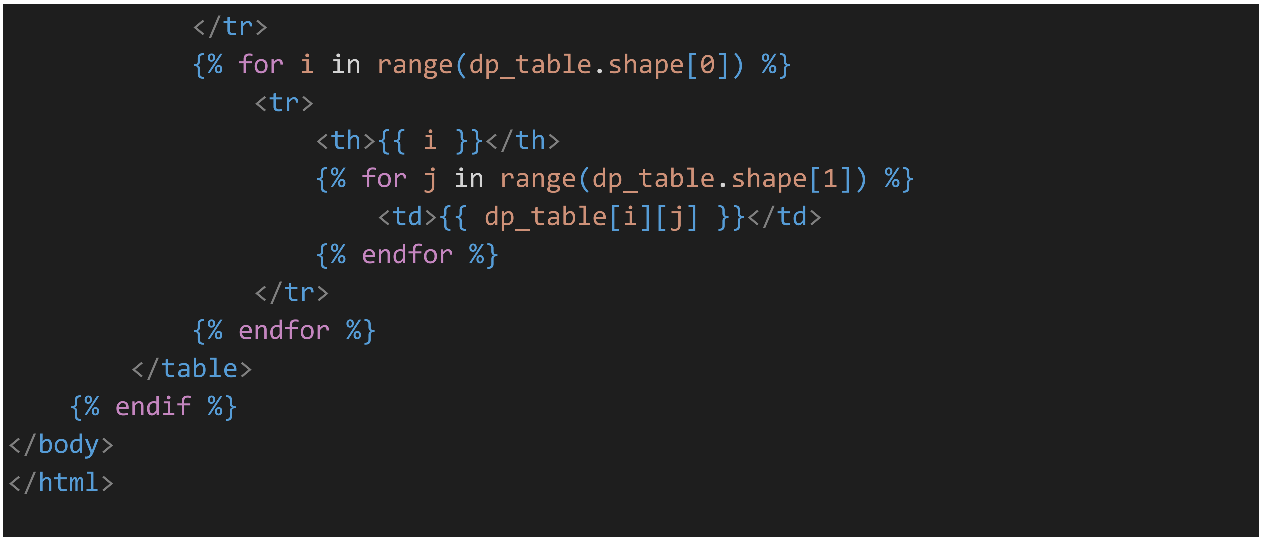
```python
    # Fill the DP table
    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if P[i - 1] == Q[j - 1]:
                dp[i][j] = dp[i - 1][j - 1] + 1
            else:
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])

    # Construct the LCS from the DP table
    lcs = []
    i, j = m, n
    while i > 0 and j > 0:
        if P[i - 1] == Q[j - 1]:
            lcs.append(P[i - 1])
            i -= 1
            j -= 1
        elif dp[i - 1][j] >= dp[i][j - 1]:
            i -= 1
        else:
            j -= 1

    lcs.reverse()  # Reverse to get the correct order
    return lcs, dp

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Get sequences as strings and convert them to lists of characters
        P = list(request.form['sequence1'].strip())
        Q = list(request.form['sequence2'].strip())

        lcs, dp_table = longest_common_subsequence(P, Q)

        return render_template('p8.html', sequence1=P, sequence2=Q, lcs=lcs,
dp_table=dp_table)

    return render_template('p8.html', sequence1=None, sequence2=None)

if __name__ == '__main__':
    app.run(debug=True)
```

.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Longest Common Subsequence</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
        }
        h1 {
            color: #333;
        }
        form {
            background: #fff;
            padding: 20px;
            border-radius: 5px;
            box-shadow: 0 2px 10px rgba(0,0,0,0.1);
            margin-bottom: 20px;
        }
        label {
            display: block;
            margin-bottom: 5px;
        }
        input[type="text"] {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        input[type="submit"] {
            background-color: #28a745;
            color: white;
            padding: 10px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 100%;
        }
```

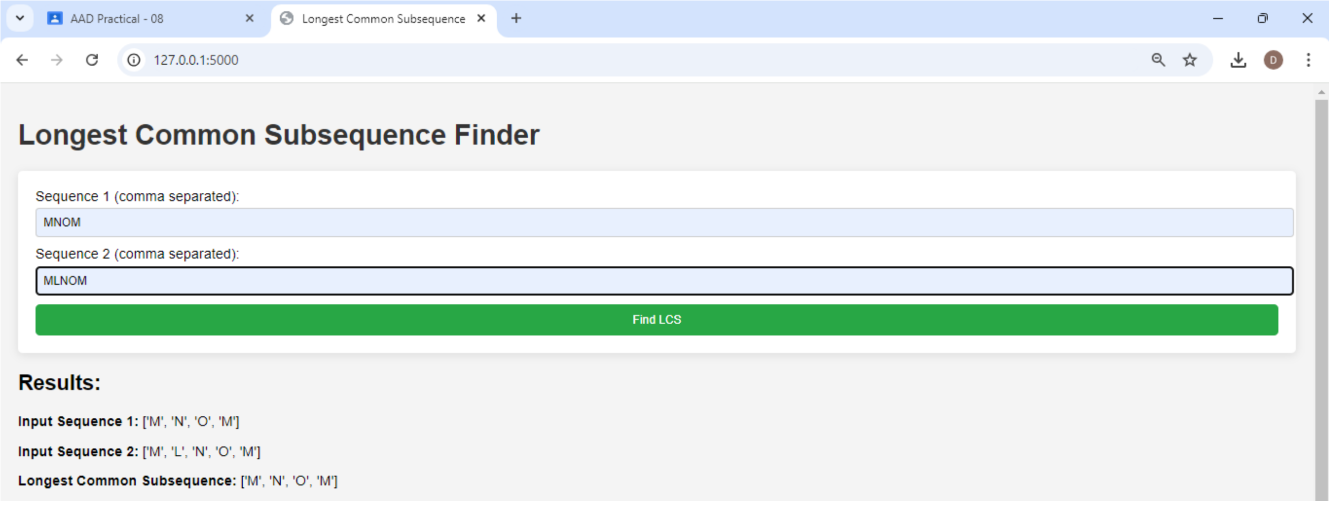


```
        input[type="submit"]:hover {
            background-color: #218838;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        table, th, td {
            border: 1px solid #ddd;
        }
        th, td {
            padding: 8px;
            text-align: center;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>
    <h1>Longest Common Subsequence Finder</h1>
    <form method="post">
        <label for="sequence1">Sequence 1 (comma separated):</label>
        <input type="text" id="sequence1" name="sequence1" required>

        <label for="sequence2">Sequence 2 (comma separated):</label>
        <input type="text" id="sequence2" name="sequence2" required>

        <input type="submit" value="Find LCS">
    </form>

    {% if sequence1 %}
        <h2>Results:</h2>
        <p><strong>Input Sequence 1:</strong> {{ sequence1 }}</p>
        <p><strong>Input Sequence 2:</strong> {{ sequence2 }}</p>
        <p><strong>Longest Common Subsequence:</strong> {{ lcs }}</p>

        <h3>DP Table:</h3>
        <table>
            <tr>
                <th></th>
                {% for j in range(dp_table.shape[1]) %}
                    <th>{{ j }}</th>
                {% endfor %}
```



```
            </tr>
            {% for i in range(dp_table.shape[0]) %}
                <tr>
                    <th>{{ i }}</th>
                    {% for j in range(dp_table.shape[1]) %}
                        <td>{{ dp_table[i][j] }}</td>
                    {% endfor %}
                </tr>
            {% endfor %}
        </table>
    {% endif %}
</body>
</html>
```

OUTPUT:



Longest Common Subsequence Finder

Sequence 1 (comma separated):

MNOM

Sequence 2 (comma separated):

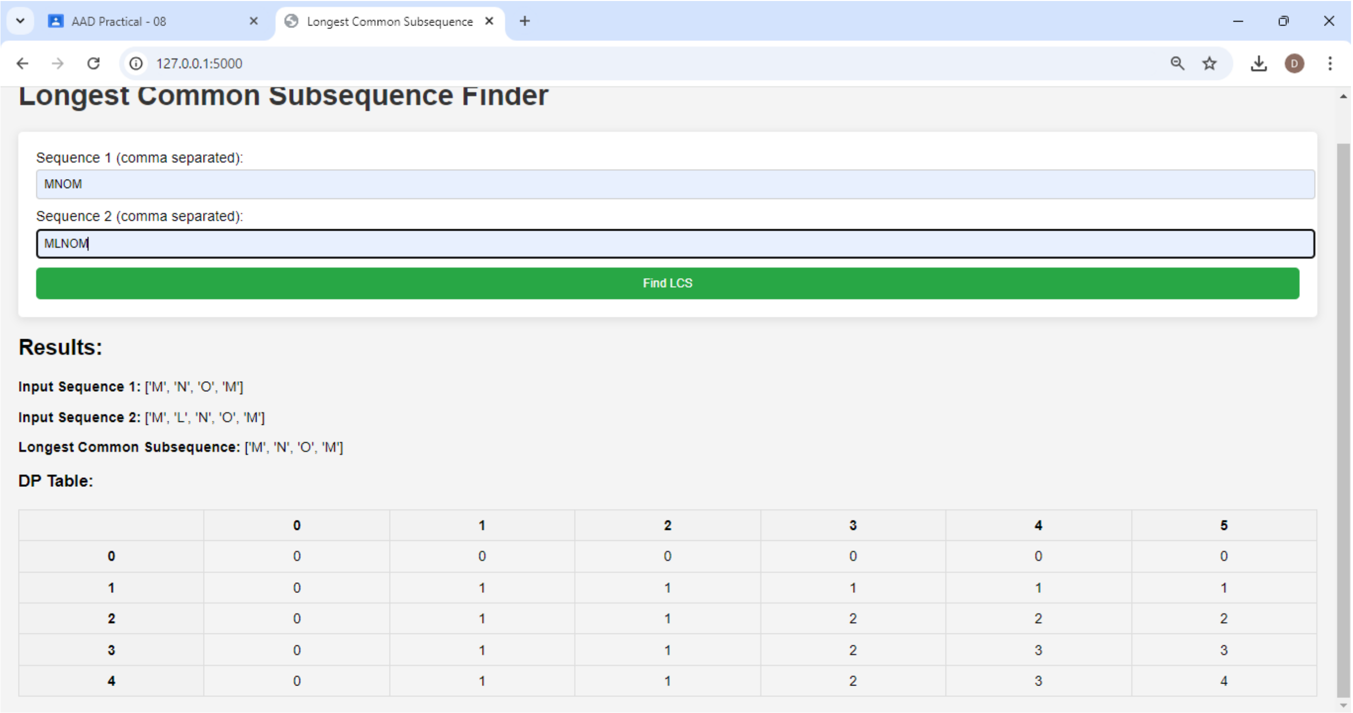
MLNOM

Results:

Input Sequence 1: ['M', 'N', 'O', 'M']

Input Sequence 2: ['M', 'L', 'N', 'O', 'M']

Longest Common Subsequence: ['M', 'N', 'O', 'M']



Longest Common Subsequence Finder

Sequence 1 (comma separated):

MNOM

Sequence 2 (comma separated):

MLNOM

Results:

Input Sequence 1: ['M', 'N', 'O', 'M']

Input Sequence 2: ['M', 'L', 'N', 'O', 'M']

Longest Common Subsequence: ['M', 'N', 'O', 'M']

DP Table:

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	1	1	1	1
2	0	1	1	2	2	2
3	0	1	1	2	3	3
4	0	1	1	2	3	4