Name : Harpalsinh Bhati

Enr. No.: 22162171010

Sub: Algorithm Analysis & Design

Branch: CS

Batch: 54

Sem: 5-B

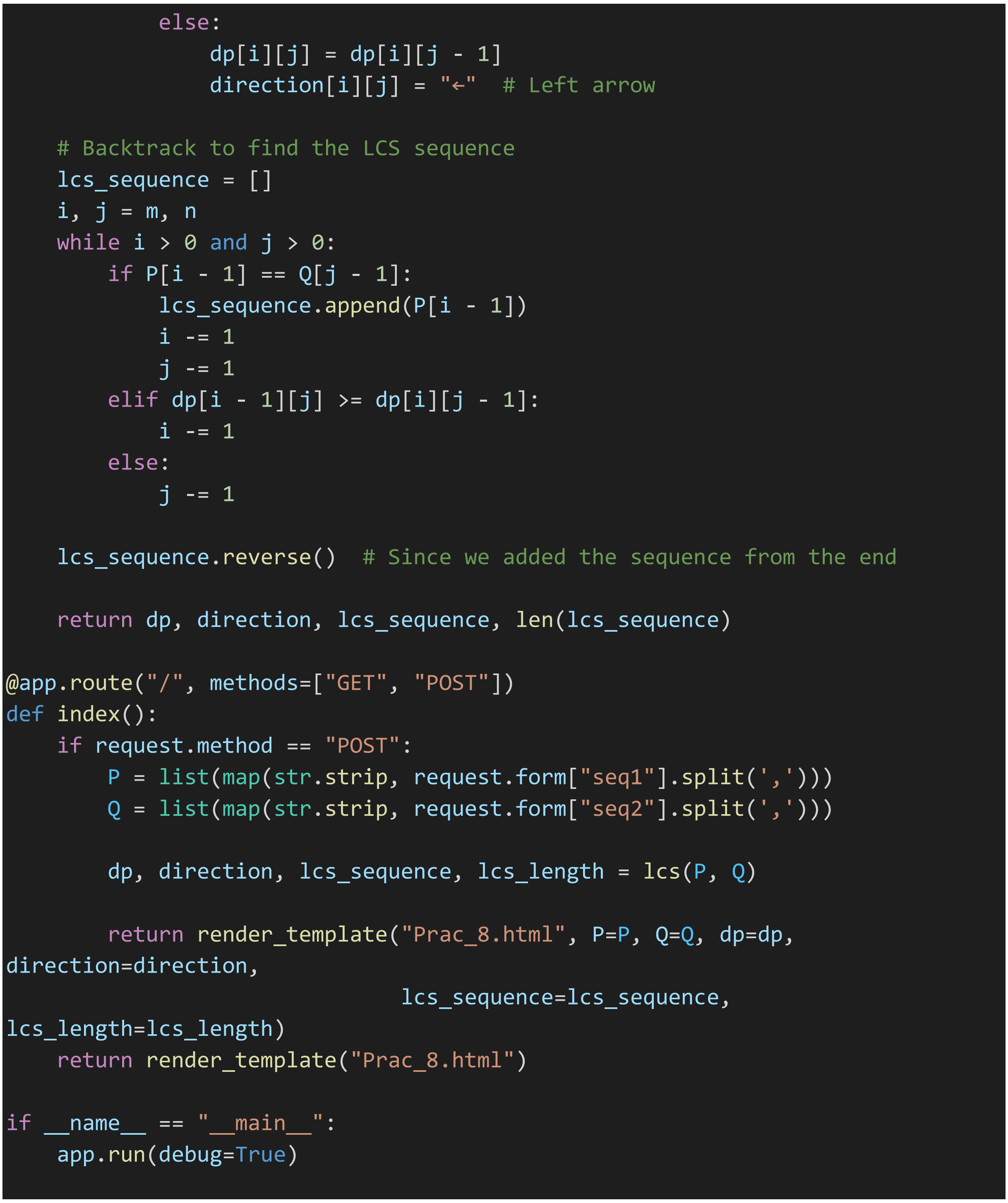
Practical-8

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 = &lt;M, N, O, M&gt; and Q = &lt;M, L, N, O, M&gt;, find any one longest common subsequence. In case multiple olutions exist, print any of them. It is guaranteed that at least one non-empty common subsequence will exist.

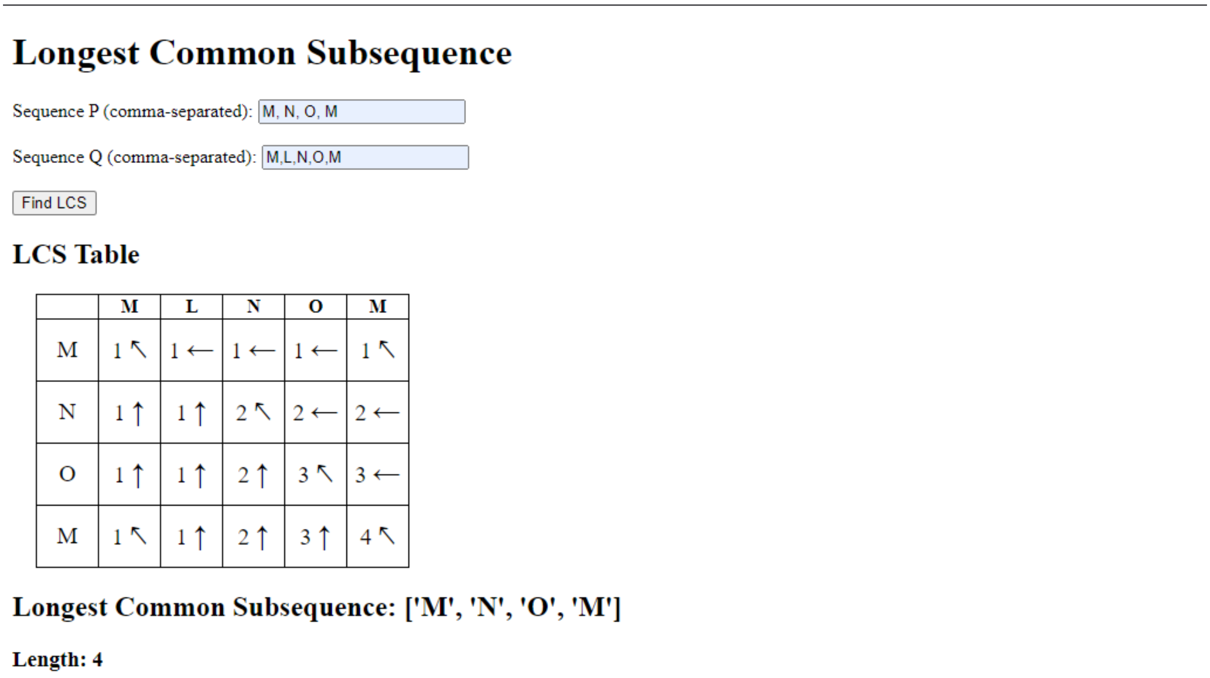
Code:

```
from flask import Flask, render_template, request

app = Flask(__name__)

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

    # Create LCS table with additional row and column for base case (0s)
    dp = [[0] * (n + 1) for _ in range(m + 1)]
    direction = [[""] * (n + 1) for _ in range(m + 1)]  # To store the
direction for backtracking

    # Fill the LCS 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
                direction[i][j] = "↖️"  # Diagonal arrow (match)
            elif dp[i - 1][j] >= dp[i][j - 1]:
                dp[i][j] = dp[i - 1][j]
                direction[i][j] = "↑"  # Up arrow
```



```
            else:
                dp[i][j] = dp[i][j - 1]
                direction[i][j] = "←"  # Left arrow

    # Backtrack to find the LCS sequence
    lcs_sequence = []
    i, j = m, n
    while i > 0 and j > 0:
        if P[i - 1] == Q[j - 1]:
            lcs_sequence.append(P[i - 1])
            i -= 1
            j -= 1
        elif dp[i - 1][j] >= dp[i][j - 1]:
            i -= 1
        else:
            j -= 1

    lcs_sequence.reverse()  # Since we added the sequence from the end

    return dp, direction, lcs_sequence, len(lcs_sequence)

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":
        P = list(map(str.strip, request.form["seq1"].split(',')))
        Q = list(map(str.strip, request.form["seq2"].split(',')))

        dp, direction, lcs_sequence, lcs_length = lcs(P, Q)

        return render_template("Prac_8.html", P=P, Q=Q, dp=dp,
direction=direction,
                               lcs_sequence=lcs_sequence,
lcs_length=lcs_length)
    return render_template("Prac_8.html")

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

Output:



Longest Common Subsequence

Sequence P (comma-separated): M, N, O, M

Sequence Q (comma-separated): M,L,N,O,M

LCS Table

	M	L	N	O	M
M	1 ↖	1 ←	1 ←	1 ←	1 ↖
N	1 ↑	1 ↑	2 ↖	2 ←	2 ←
O	1 ↑	1 ↑	2 ↑	3 ↖	3 ←
M	1 ↖	1 ↑	2 ↑	3 ↑	4 ↖

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

Length: 4