Name:-Devansh Koyani

Er no:-22162171007

Batch:-54

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:-

```
from flask import Flask, render_template, request

app = Flask(__name__)

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



    dp = [[0] * (n + 1) for _ in range(m + 1)]
     direction = [[""] * (n + 1) for _ in range(m + 1)]  # To store the
direction for backtracking



    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
```

Name:-Devansh Koyani

Er no:-22162171007

Batch:-54

```
                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


    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()

    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")
```

Name:-Devansh Koyani

Er no:-22162171007

Batch:-54

```
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>LCS with Arrows</title>
    <style>
        table {
            border-collapse: collapse;
            margin: 20px auto;
            width: 80%;
        }
        table, th, td {
            border: 1px solid black;
        }
        td {
            width: 50px;
            height: 50px;
            text-align: center;
            font-size: 20px;
        }
        .arrow {
            font-size: 24px;
        }
        .center {
            text-align: center;
        }
    </style>
</head>
<body>

<h1 class="center">Longest Common Subsequence</h1>
```

Name:-Devansh Koyani

Er no:-22162171007

Batch:-54

```
<form method="POST" class="center">
    <label for="seq1">Sequence P (comma-separated):</label>
    <input type="text" id="seq1" name="seq1" required><br><br>

    <label for="seq2">Sequence Q (comma-separated):</label>
    <input type="text" id="seq2" name="seq2" required><br><br>

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

{% if dp %}
    <h2 class="center">LCS Table</h2>
    <table>
        <thead>
            <tr>
                <th></th>
                {% for q in Q %}
                    <th>{{ q }}</th>
                {% endfor %}
            </tr>
        </thead>
        <tbody>
            {% for i in range(1, dp|length) %}
                <tr>
                    <td>{{ P[i-1] }}</td>
                    {% for j in range(1, dp[i]|length) %}
                                <td>{{ dp[i][j] }} <span class="arrow">{{
direction[i][j] }}</span></td>
                    {% endfor %}
                </tr>
            {% endfor %}
        </tbody>
    </table>

    <h2 class="center">Longest Common Subsequence: {{ lcs_sequence|join(',
') }}</h2>
    <h3 class="center">Length: {{ lcs_length }}</h3>
```

Name:-Devansh Koyani

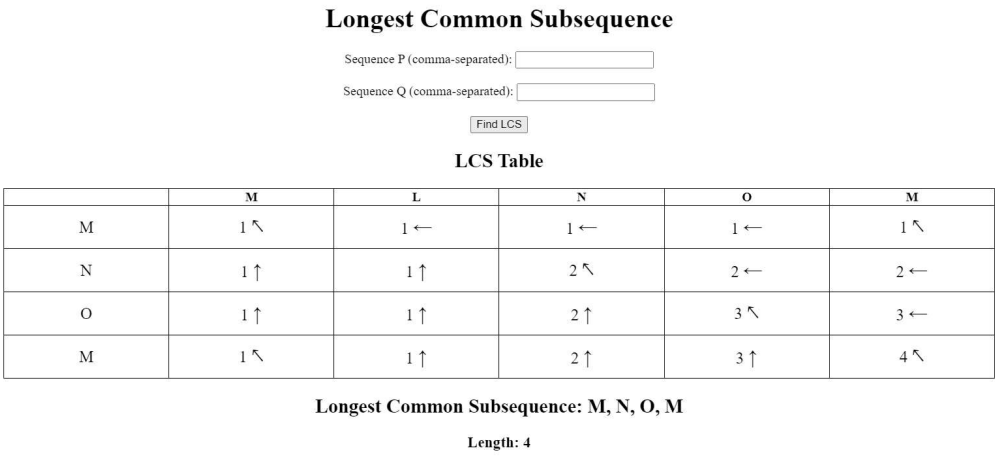
Er no:-22162171007

Batch:-54

```
{% endif %}


</body>
</html>
```

Output:-



Longest Common Subsequence

Sequence P (comma-separated):

Sequence Q (comma-separated):

Find LCS

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