Name: Kaustubh Dhongade

Reg no.: 231070018

Batch: A

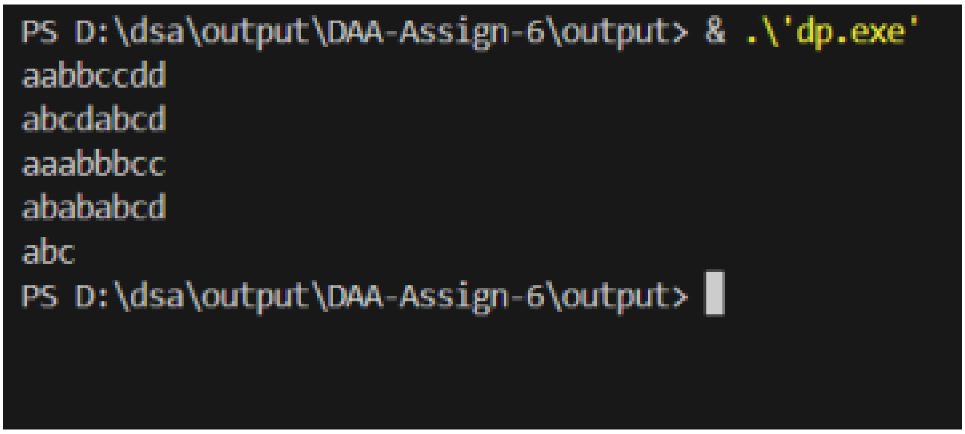
Lab-6

Aim:

Task -1 : Consider grades received by 20 students, like AA, AB, BB, ..., FF of each student. Computer the Longest common sequence of grades among students.

```
#include <bits/stdc++.h>
using namespace std;

string LCS(string X, string Y) {
    int m = X.size();
    int n = Y.size();
    vector<vector<int>> dp(m + 1, vector<int>(n + 1, 0));

    X = '0' + X;
    Y = '0' + Y;

    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (X[i] == Y[j]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }

    string ans = "";
    int i = m, j = n;
    while (i > 0 && j > 0) {
        if (X[i] == Y[j]) {
            ans = string(1, X[i]) + ans;
            i--;
            j--;
        } else if (dp[i - 1][j] > dp[i][j - 1]) {
            i--;
        } else {
            j--;
        }
    }

    return ans;
}

int main() {
    vector<string> v;

    for (int i = 0; i < 4; i++) {
        string s;
        cin >> s;
        v.push_back(s);
    }

    string ans = v[0];
    for (int i = 1; i < 4; i++) {
        ans = LCS(ans, v[i]);
    }

    cout << ans << endl;

    return 0;
}
```

Output



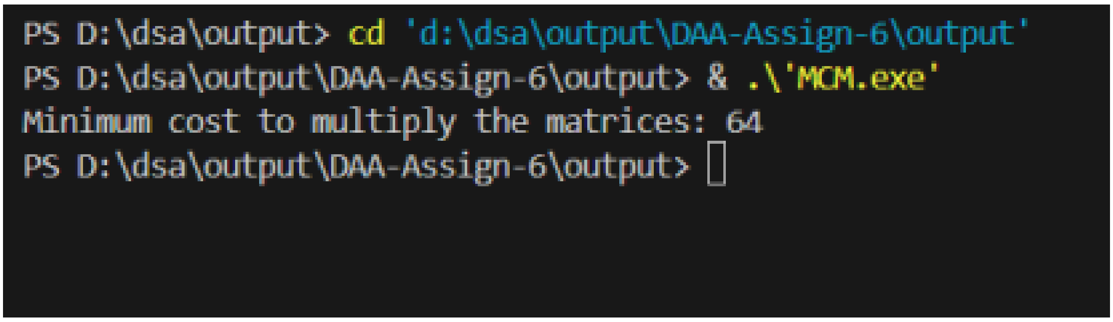
```
PS D:\dsa\output\DAA-Assign-6\output> & .\'dp.exe'
aabbccdd
abcdabcd
aaabbbcc
abababcd
abc
PS D:\dsa\output\DAA-Assign-6\output> 
```

Aim: Task 2: Consider meteorological data like temperature, dew point, wind direction, wind speed, cloud cover, cloud layer(s) for each city. This data is available in two dimensional array for a week. Assuming all tables are compatible for multiplication. You have to implement the matrix chain multiplication algorithm to find fastest way to complete the matrices multiplication to achieve timely predication.

Code:

```
#include<iostream>
#include<vector>
#include <climits>

using namespace std;

int matrixMultiplication(const vector<int>& dimensions)
{
    int n = dimensions.size();
    vector<vector<int>> dp(n, vector<int>(n, 0));

    for (int s = n - 2; s >= 0; --s)
    { // Start from n-2 as we need at least 2 matrices
        for (int e = s + 1; e < n; ++e)
        { // e must be greater than s
            int minCost = INT_MAX;
            for (int k = s + 1; k < e; ++k)
            {
                int cost = dp[s][k] + dp[k][e] + dimensions[s] *
dimensions[k] * dimensions[e];
                minCost = min(minCost, cost);
            }
            dp[s][e] = (minCost == INT_MAX ? 0 : minCost); // Set to 0 if no
valid split
        }
    }

    return dp[0][n - 1];
}

int main() {
    vector<int> dimensions = {1, 2, 3 ,4};
    cout << "Minimum cost to multiply the matrices: " <<
matrixMultiplication(dimensions) << endl;
    return 0;
}
```

Output



```
PS D:\dsa\output> cd 'd:\dsa\output\DAA-Assign-6\output'
PS D:\dsa\output\DAA-Assign-6\output> & .\'MCM.exe'
Minimum cost to multiply the matrices: 64
PS D:\dsa\output\DAA-Assign-6\output> 
```

Conclusion: Using the **Matrix Chain Multiplication (MCM)** algorithm for meteorological data ensures efficient processing of complex datasets like temperature, wind, and cloud cover across multiple cities. It minimizes computational costs, saves resources, and enables faster weather predictions, which are crucial for timely decision-making and scalability.

The **Longest Common Subsequence (LCS)** algorithm helps identify the longest sequence of similar grade patterns among students, even if the grades are not consecutive. This analysis can highlight common performance trends, support targeted interventions, and improve teaching strategies.