NAME: Dhairya Arora

ROLL NO.- 01616401522

BTECH IT 4TH SEM 22-26

ASSIGNMENT 14

AIM: LCS Code for implementation

CODE:

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <ctime>
#include <chrono>
using namespace std;
using namespace std::chrono;

std::string lcs(const std::string &X, const std::string &Y)
{
    int m = X.length();
    int n = Y.length();
    // Create a 2D vector to store the lengths of LCS for subproblems
    std::vector<std::vector<int>> L(m + 1, std::vector<int>(n + 1, 0));

    // Building the LCS matrix in a bottom-up manner
    for (int i = 0; i <= m; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            if (i == 0 || j == 0)
            {
```

```cpp
            L[i][j] = 0;
        }
        else if (X[i - 1] == Y[j - 1])
        {
            L[i][j] = L[i - 1][j - 1] + 1;
        }
        else
        {
            L[i][j] = std::max(L[i - 1][j], L[i][j - 1]);
        }
    }
}

// Length of LCS is the value stored at the bottom-right corner of the matrix
int length = L[m][n];

// Backtrack to find the LCS itself
std::string lcsString;
int i = m, j = n;
while (i > 0 && j > 0)
{
    if (X[i - 1] == Y[j - 1])
    {lcsString = X[i - 1] + lcsString;
        i--;
        j--;
    }
    else if (L[i - 1][j] > L[i][j - 1]) {
        i--;
    }
```

```cpp
        else {
            j--;
        }
    }
    return lcsString;
}


int main() {
    auto start_time = high_resolution_clock::now();
    string X = "AGGTAB";
    string Y = "GXTXAYB";
    string longest_common_subsequence = lcs(X, Y);
    cout << "Length of LCS is " << longest_common_subsequence.length() << std::endl;
    cout << "LCS: " << longest_common_subsequence << std::endl;
    auto end_time = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(end_time - start_time);
    cout << "\nTime taken by LCS - " << duration.count() << " Microseconds" << endl;
    return 0;
}
```
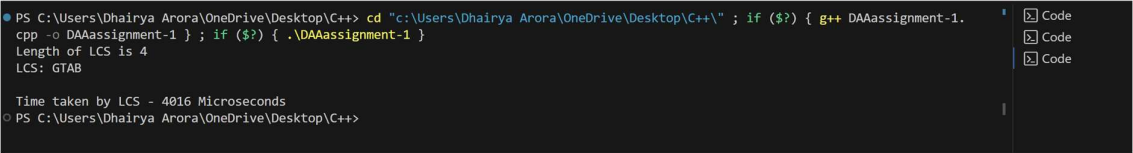
OUTPUT: