DAA EXERCISE 6

Q1

* RECURSIVE METHOD

CODE

int longestCommonSubsequence(string a, string b, int m, int n)

{

    if (n == 0 || m == 0)

        return 0;

    if (a[m - 1] == b[n - 1])

    {

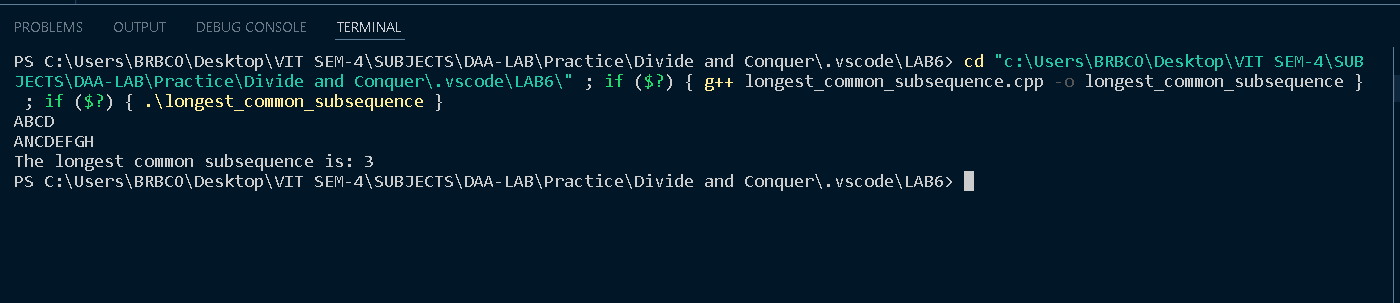
        return longestCommonSubsequence(a, b, m - 1, n - 1) + 1;

    }

    return max(longestCommonSubsequence(a, b, m - 1, n), longestCommonSubsequence(a, b, m, n - 1));

}

OUTPUT



* USING DYNAMIC PROGRAMMING

CODE

int longestSubsequence(string a, string b, int n, int m)

{

    int dp[n + 1][m + 1];

    for (int i = 0; i <= n; i++)

    {

        for (int j = 0; j <= m; j++)

        {

            if (i == 0 || j == 0)

                dp[i][j] = 0;

            else if (a[i - 1] == b[j - 1])

                dp[i][j] = 1 + dp[i - 1][j - 1];

            else

            {

                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);

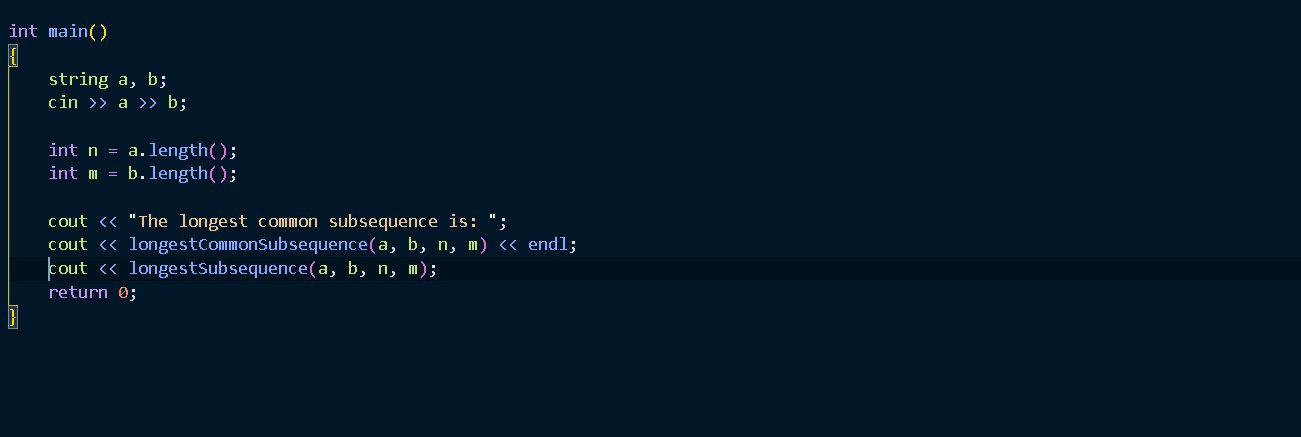
            }

        }

    }

    return dp[n][m];

}



OUTPUT

