

```

#include <iostream>
#include <string>
using namespace std;

int findLCS(string first, string second, string &lcsString) {
    int m = first.length();
    int n = second.length();

    int** c = new int*[m+1];
    for (int i = 0; i <= m; i++) {
        c[i] = new int[n+1];
    }

    for (int i = 0; i <= m; i++) {
        c[i][0] = 0;
    }
    for (int j = 0; j <= n; j++) {
        c[0][j] = 0;
    }

    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (first[i-1] == second[j-1]) {
                c[i][j] = c[i-1][j-1] + 1;
            } else {
                c[i][j] = max(c[i-1][j], c[i][j-1]);
            }
        }
    }
    int lcsLength = c[m][n];
    int i = m, j = n;
    while (i > 0 && j > 0) {
        if (first[i-1] == second[j-1]) {
            lcsString = first[i-1] + lcsString;
            i--;
            j--;
        } else if (c[i-1][j] > c[i][j-1]) {
            i--;
        } else {
            j--;
        }
    }
    for (int i = 0; i <= m; i++) {
        delete[] c[i];
    }
    delete[] c;
}

```

```

        return lcsLength;
    }

int main() {
    string first, second;
    cout << "Type first string: ";
    cin >> first;
    cout << "Type second string: ";
    cin >> second;

    cout << "First string is: " << first << "\n";
    cout << "Second string is: " << second << "\n";

    string lcsString = "";
    int lcsLength = findLCS(first, second, lcsString);

    cout << "Length of Longest Common Subsequence: " << lcsLength <<
endl;
    cout << "Longest Common Subsequence: " << lcsString << endl;

    return 0;
}

```

```

Type first string: wordless
Type second string: annual
First string is: wordless
Second string is: annual
Length of Longest Common Subsequence: 1
Longest Common Subsequence: l

```