

National Institute Of Technology Surathkal Mangalore Karnataka-575025

Department Of Information Technology



Lab Assignment :- 04

Name:- Chikkeri Chinmaya

Roll Number:- 2111T017

Branch:- Information Technology (B.Tech)

Section :- S13

Course :- Data Structure And Algorithm (IT251)

Submitted To:-

HariDas Pai Sir

1st Question :- Levenshtein Distance

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int levenshteinDistance(const string& word1, const string& word2) {
    // Initialize a 2D vector to store the dynamic programming table
    vector<vector<int>> dpTable(word1.length() + 1, vector<int>(word2.length()
+ 1, 0));

    // Fill the first row and column of the table with the corresponding
indices
    for (int i = 0; i <= word1.length(); i++) {
        dpTable[i][0] = i;
    }
    for (int j = 0; j <= word2.length(); j++) {
        dpTable[0][j] = j;
    }

    // Iterate through the rest of the table and fill it using the DP formula
    for (int i = 1; i <= word1.length(); i++) {
        for (int j = 1; j <= word2.length(); j++) {
            if (word1[i - 1] == word2[j - 1]) {
                dpTable[i][j] = dpTable[i - 1][j - 1];
            }
            else {
                dpTable[i][j] = 1 + min({ dpTable[i - 1][j], dpTable[i][j -
1], dpTable[i - 1][j - 1] });
            }
        }
    }

    // Return the bottom-right element of the table
    return dpTable[word1.length()][word2.length()];
}

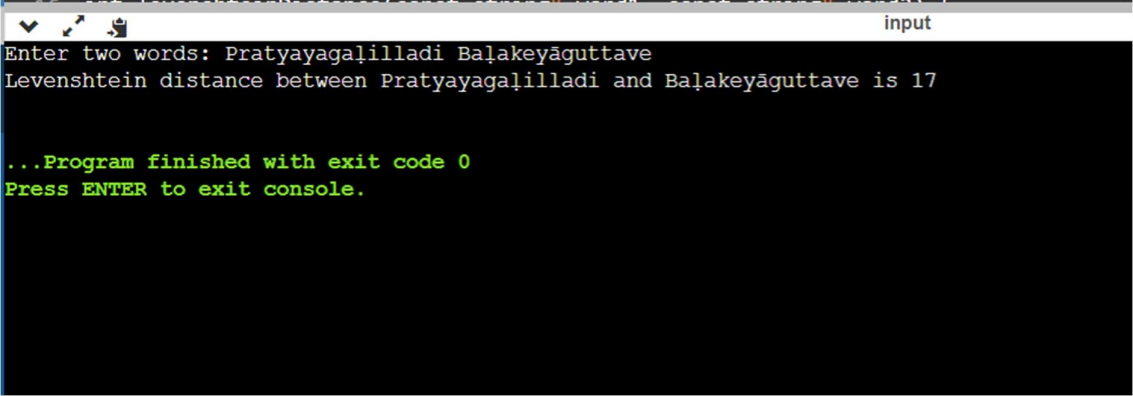
int main() {
    string word1, word2;
    cout << "Enter two words: ";
    cin >> word1 >> word2;

    int distance = levenshteinDistance(word1, word2);
```

```
    cout << "Levenshtein distance between " << word1 << " and " << word2 << "  
is " << distance << endl;  
  
    return 0;  
}
```

To calculate the Levenshtein distance between two strings, which is the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other, the function "levenshteinDistance" which takes two constant references to strings as input and returns an integer computes the Levenshtein distance between the two strings using a dynamic programming approach. 2D vector of integers called "dpTable" with dimensions $(\text{word1.length() + 1}) \times (\text{word2.length() + 1})$. The main function prompts the user to enter two words and reads them into the variables "word1" and "word2". It then calls the "levenshteinDistance" function and stores the result in the variable "distance".

OutPut:-



```
input  
Enter two words: Pratyayagalilladi Balakeyaguttave  
Levenshtein distance between Pratyayagalilladi and Balakeyaguttave is 17  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```