**Program :**

#include <bits/stdc++.h>

using namespace std;

int L[10][10];

void display(string X, string Y, int m, int n){

    cout << "    ";

    for (int i = 0; i < m; i++)

        cout << X[i] << " ";

    cout << endl;

    for (int i = 0; i <= n; i++)

    {

        if (i > 0)

            cout << Y[i - 1] << " ";

        else

            cout << "  ";

        for (int j = 0; j <= m; j++)cout << L[j][i] << " ";

        cout << endl;

    }

}

int lcs(string X, string Y, int m, int n){

    int i, j;

    vector<char> lcs;

    for (i = 0; i <= m; i++){

        for (j = 0; j <= n; j++){

            if (i == 0 || j == 0) //initializing top left edgr with 0

                L[i][j] = 0;

            else if (X[i - 1] == Y[j - 1])//if equal add 1 to diag elem

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

            else                          // if unequal take max of upper or

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

        }

    }

    display(X,Y,m,n);

    i = m;

    j = n;

    cout << "LCS : ";

    while (i > 0 && j > 0)

    {

        if (L[i][j] == L[i][j - 1])

        {

            j = j - 1;

        }

        else if (L[i][j] == L[i - 1][j])

        {

            i = i - 1;

        }

        else if (L[i][j] == L[i - 1][j - 1] + 1)

        {

            lcs.push\_back(X[i - 1]);

            i = i - 1;

            j = j - 1;

        }

    }

    for (auto x = lcs.rbegin(); x != lcs.rend(); ++x)

        cout << \*x;

    return L[m][n];

}

int main()

{

    string str1, str2;

    cout << "Enter string 1 :";

    cin >> str1;

    cout << "Enter string 2 :";

    cin >> str2;

    int m = str1.length();

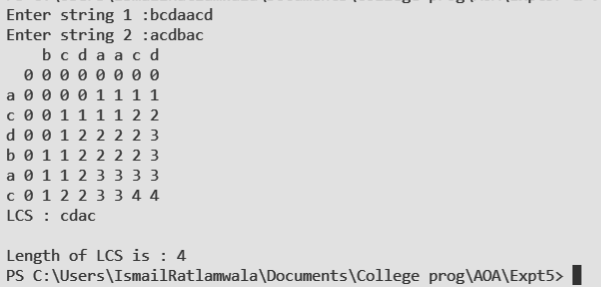
    int n = str2.length();

    printf("\n\nLength of LCS is : %d\n", lcs(str1, str2, m, n));

    return 0;

}

**Output :**

****

**Program :**

import java.util.Arrays;

import java.util.Scanner;

public class PrimsAlgorithm {

  public static void Prim(int G[][], int V) {

    int INF = 9999999;

    int no\_edge;

    boolean[] selected = new boolean[V];

    Arrays.fill(selected, false);

    no\_edge = 0;

    selected[0] = true;

    System.out.println("Edge : Weight");

    while (no\_edge < V - 1) {

      int min = INF;

      int x = 0; // row number

      int y = 0; // col number

      for (int i = 0; i < V; i++) {

        if (selected[i] == true) {

          for (int j = 0; j < V; j++) {

            if (!selected[j] && G[i][j] != 0) {

              if (min > G[i][j]) {

                min = G[i][j];

                x = i;

                y = j;

              }

            }

          }

        }

      }

      System.out.println(x + " - " + y + " :  " + G[x][y]);

      selected[y] = true;

      no\_edge++;

    }

  }

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of nodes : ");

    int n = sc.nextInt();

    int[][] G = new int[n][n];

    for(int i=0;i<n;i++)

        for(int j=0;j<n;j++) G[i][j] = sc.nextInt();

    Prim(G, n);

  }

}

**Output :**

