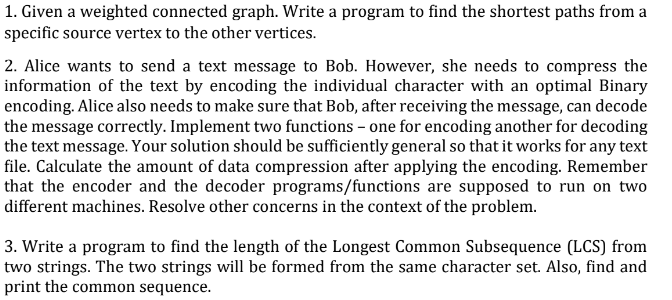
**Graded Programming Assignments-3**

**Submitted by: D Veera Harsha Vardhan Reddy**



CODE:

#include <iostream>

using namespace std;

#include <limits.h>

#define V 5

int minDistance(int dist[], bool sptSet[])

{

    int min = INT\_MAX, min\_index;

    for (int v = 0; v < V; v++)

        if (sptSet[v] == false && dist[v] <= min)

            min = dist[v], min\_index = v;

    return min\_index;

}

void printSolution(int dist[])

{

    cout << "Vertex \t Distance from Source" << endl;

    for (int i = 0; i < V; i++)

        cout << i << " \t\t\t\t" << dist[i] << endl;

}

void dijkstra(int graph[V][V], int src)

{

    int dist[V];

    bool sptSet[V];

    for (int i = 0; i < V; i++)

        dist[i] = INT\_MAX, sptSet[i] = false;

    dist[src] = 0;

    for (int count = 0; count < V - 1; count++)

    {

        int u = minDistance(dist, sptSet);

        sptSet[u] = true;

        for (int v = 0; v < V; v++)

            if (!sptSet[v] && graph[u][v] && dist[u] != INT\_MAX && dist[u] + graph[u][v] < dist[v])

                dist[v] = dist[u] + graph[u][v];

    }

    printSolution(dist);

}

int main()

{

    int graph[V][V] = {{0,5,0,7,0},

                       {5,0,10,4,0},

                       {0,10,0,6,9},

                       {7,4,6,0,8},

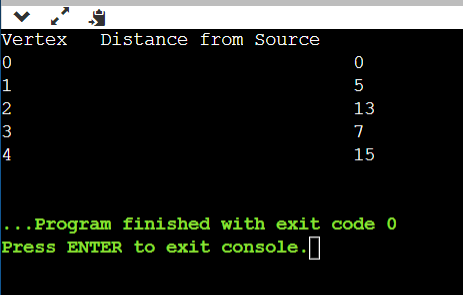
                       {0,0,9,8,0}};

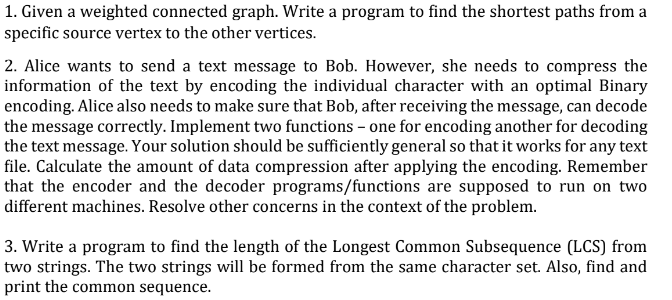
    dijkstra(graph, 0);

    return 0;

}

OUTPUT:





CODE:

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

void lcsAlgo(char X[],char Y[])

{

    int i, j, m, n, table[20][20];

    m = strlen(X);

    n = strlen(Y);

    for (i = 0; i <= m; i++)

        table[i][0] = 0;

    for (i = 0; i <= n; i++)

        table[0][i] = 0;

    for (i = 1; i <= m; i++)

        for (j = 1; j <= n; j++)

        {

            if (X[i - 1] == Y[j - 1])

            {

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

            }

            else if (table[i - 1][j] >= table[i][j - 1])

            {

                table[i][j] = table[i - 1][j];

            }

            else

            {

                table[i][j] = table[i][j - 1];

            }

        }

    int index = table[m][n];

    char lcsAlgo[index + 1];

    lcsAlgo[index] = '\0';

    i = m;

    j = n;

    while (i > 0 && j > 0)

    {

        if (X[i - 1] == Y[j - 1])

        {

            lcsAlgo[index - 1] = X[i - 1];

            i--;

            j--;

            index--;

        }

        else if (table[i - 1][j] > table[i][j - 1])

            i--;

        else

            j--;

    }

    cout<<endl;

    cout<<"First Sequence : "<<X<<endl;

    cout<<endl;

    cout<<"Second Sequence : "<<Y<<endl;

    cout<<endl;

    cout<<"LCS: "<<lcsAlgo<<endl;

    cout<<endl;

    cout<<"Count of LCS is: "<<strlen(lcsAlgo);

    cout<<endl;

}

int main()

{

    char X[20] = "zyzzyzz", Y[20] = "zyzzzy";

    lcsAlgo(X,Y);

}

OUTPUT:

