**DAA Lab**

**Practical 5**

Name: Madhura Mahatme

Section: A1-B3

Roll no.: 36

**Aim:**

Implement Longest Common Subsequence (LCS) algorithm to find the length and LCS for DNA sequences.

**Problem Statement:**

DNA sequences can be viewed as strings of A, C, G, and T characters, which represent

nucleotides. Finding the similarities between two DNA sequences are an important

computation performed in bioinformatics.

[Note that a subsequence might not include consecutive elements of the original sequence.]

**TASK-1:**

Find the similarity between the given X and Y sequence.

X=AGCCCTAAGGGCTACCTAGCTT

Y= GACAGCCTACAAGCGTTAGCTTG

**Code:**

#include <stdio.h>

#include <string.h>

#define MAX 100

void lcs(char a[], char b[], int c[MAX][MAX], char dir[MAX][MAX], int m, int n) {

    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 (a[i-1] == b[j-1]) {

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

                dir[i][j] = 'D';

            }

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

                c[i][j] = c[i-1][j];

                dir[i][j] = 'U';

            }

            else{

                c[i][j] = c[i][j-1];

                dir[i][j] = 'L';

            }

        }

    }

}

void print\_matrix(int c[MAX][MAX], char dir[MAX][MAX], int m, int n, char a[], char b[]) {

    printf("\nCost Matrix with Directions:\n\n   ");

    for(int j = 0; j < n; j++){

        printf("   %c", b[j]);

    }

    printf("\n");

    for (int i = 0; i <= m; i++){

        if (i==0) printf("  ");

        else printf("%c ", a[i-1]);

        for (int j = 0; j <= n; j++){

            if (i==0 || j==0)

                printf("  %2d ", c[i][j]);

            else

                printf(" %2d%c ", c[i][j], dir[i][j]);

        }

        printf("\n");

    }

}

void print\_lcs(char a[], char b[], char dir[MAX][MAX], int i, int j){

    if(i==0 || j==0)

        return;

    if(dir[i][j] == 'D'){

        print\_lcs(a, b, dir, i-1, j-1);

        printf("%c", a[i-1]);

    } else if (dir[i][j] == 'U'){

        print\_lcs(a, b, dir, i-1, j);

    } else {

        print\_lcs(a, b, dir, i, j-1);

    }

}

int main(){

    char X[MAX], Y[MAX];

    int c[MAX][MAX];

    char dir[MAX][MAX];

    printf("Enter first sequence (X): ");

    scanf("%s", X);

    printf("Enter second sequence (Y): ");

    scanf("%s", Y);

    int m = strlen(X);

    int n = strlen(Y);

    lcs(X, Y, c, dir, m, n);

    print\_matrix(c, dir, m, n, X, Y);

    printf("\nFinal cost (Length of LCS): %d\n", c[m][n]);

    printf("LCS: ");

    print\_lcs(X, Y, dir, m, n);

    printf("\n");

    return 0;

}

A screen shot of a computer program

AI-generated content may be incorrect.

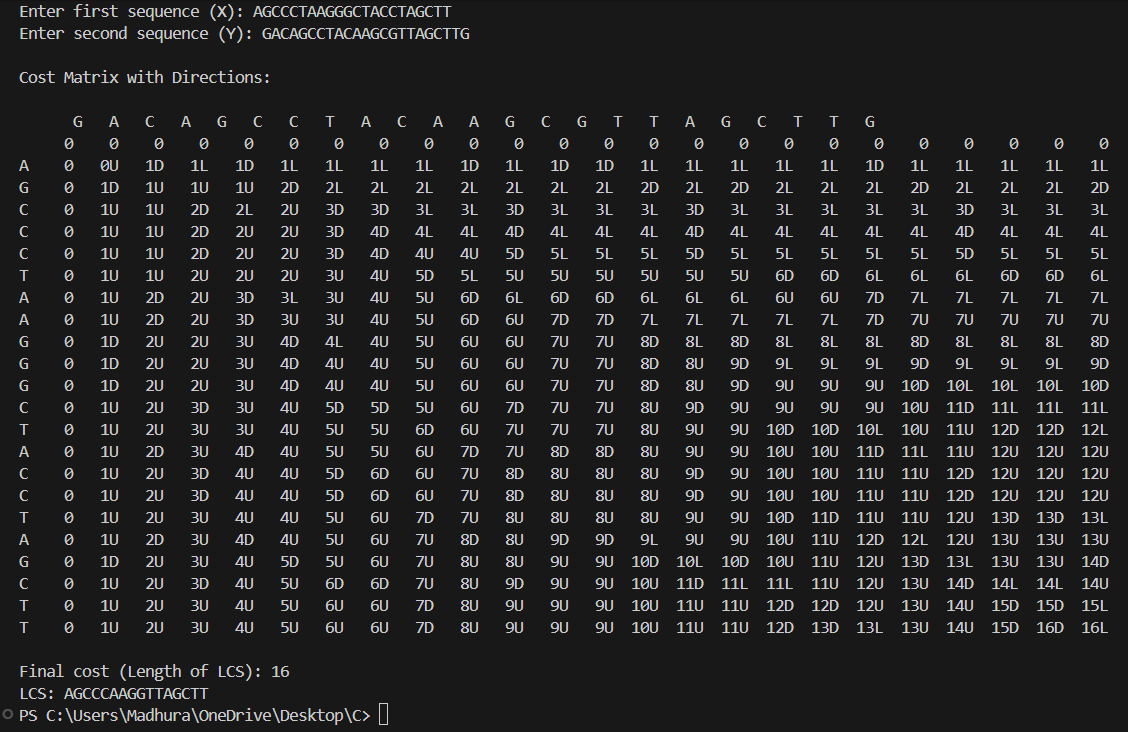
A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a black screen with white text

AI-generated content may be incorrect.

**Output:**



**TASK-2:**

Find the longest repeating subsequence (LRS). Consider it as a variation of the

longest common subsequence (LCS) problem.

Let the given string be S. You need to find the LRS within S. To use the LCS framework, you effectively compare S with itself. So, consider string1 = S and string2 = S.

Example:

AABCBDC

LRS= ABC or ABD

**Code:**

#include <stdio.h>

#include <string.h>

#define MAX 100

void lrs(char s[], int c[MAX][MAX], int m){

    for (int i=0; i <= m; i++) {

        c[i][0] = 0;

        c[0][i] = 0;

    }

    for (int i = 1; i <= m; i++){

        for (int j = 1; j <= m; j++){

            if (s[i-1]==s[j-1] && i != j){

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

            }

            else{

                c[i][j] = (c[i-1][j] > c[i][j-1]) ? c[i-1][j] : c[i][j-1];

            }

        }

    }

}

void print\_lrs(char s[], int c[MAX][MAX], int i, int j){

    if (i==0 || j==0) return;

    if (s[i-1] == s[j-1] && i != j){

        print\_lrs(s, c, i-1, j-1);

        printf("%c", s[i-1]);

    }

    else if(c[i-1][j] > c[i][j-1]){

        print\_lrs(s, c, i-1, j);

    }

    else{

        print\_lrs(s, c, i, j-1);

    }

}

int main(){

    char S[MAX];

    int c[MAX][MAX];

    printf("Enter the string: ");

    scanf("%s", S);

    int m = strlen(S);

    lrs(S, c, m);

    printf("\nLength of Longest Repeating Subsequence: %d\n", c[m][m]);

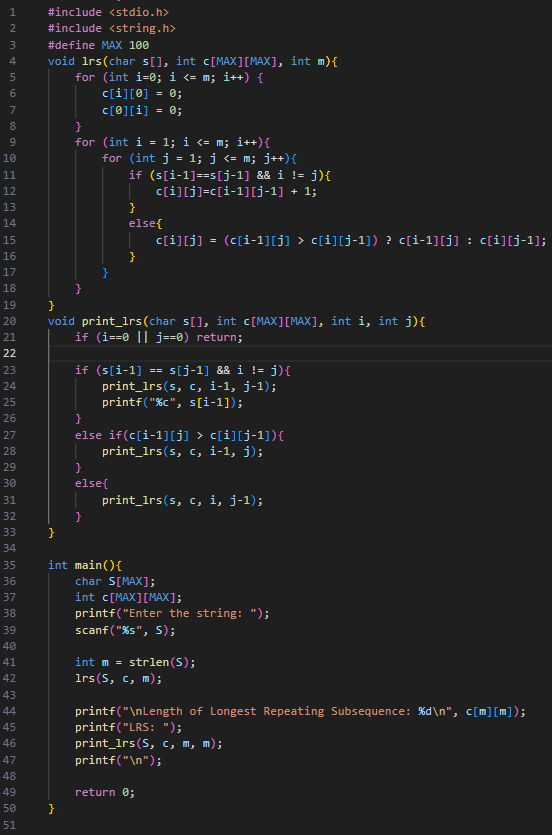
    printf("LRS: ");

    print\_lrs(S, c, m, m);

    printf("\n");

    return 0;

}

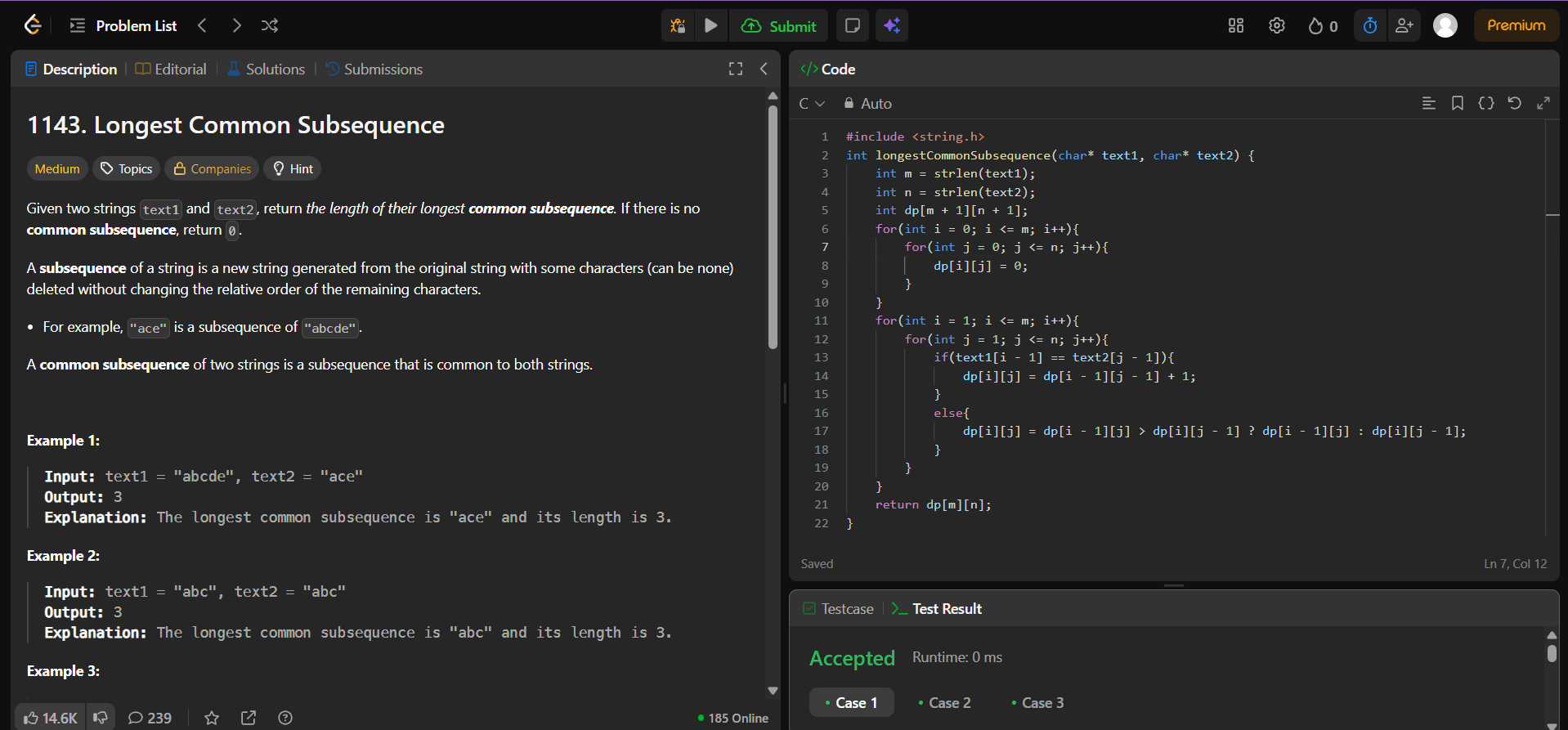


**Output:**

A screen shot of a computer

AI-generated content may be incorrect.

**LeetCode Assesment:**



A screenshot of a computer

AI-generated content may be incorrect.