Practical :-5

Name: Ujjwal Tiwari

Section:A2

Batch :B4

Roll : 51

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

**TASK-1: Longest Common Subsequence (LCS) for DNA Sequences**

#include <stdio.h>

#include <string.h>

int max(int a, int b) { return (a > b) ? a : b; }

void findLCS(char X[], char Y[]) {

int n = strlen(X);

int m = strlen(Y);

int dp[n + 1][m + 1];

int i, j;

for (i = 0; i <= n; i++) {

for (j = 0; j <= m; j++) {

if (i == 0 || j == 0)

dp[i][j] = 0;

else if (X[i - 1] == Y[j - 1])

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

else

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

}

}

printf("\nCOST MATRIX:\n");

for (i = 0; i <= n; i++) {

for (j = 0; j <= m; j++) {

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

}

printf("\n");

}

int index = dp[n][m];

char lcs[index + 1];

lcs[index] = '\0';

i = n;

j = m;

while (i > 0 && j > 0) {

if (X[i - 1] == Y[j - 1]) {

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

i--;

j--;

index--;

} else if (dp[i - 1][j] > dp[i][j - 1])

i--;

else

j--;

}

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

printf("LCS: %s\n", lcs);

}

int main() {

char X[] = "AGCCCTAAGGGCTACCTAGCTT";

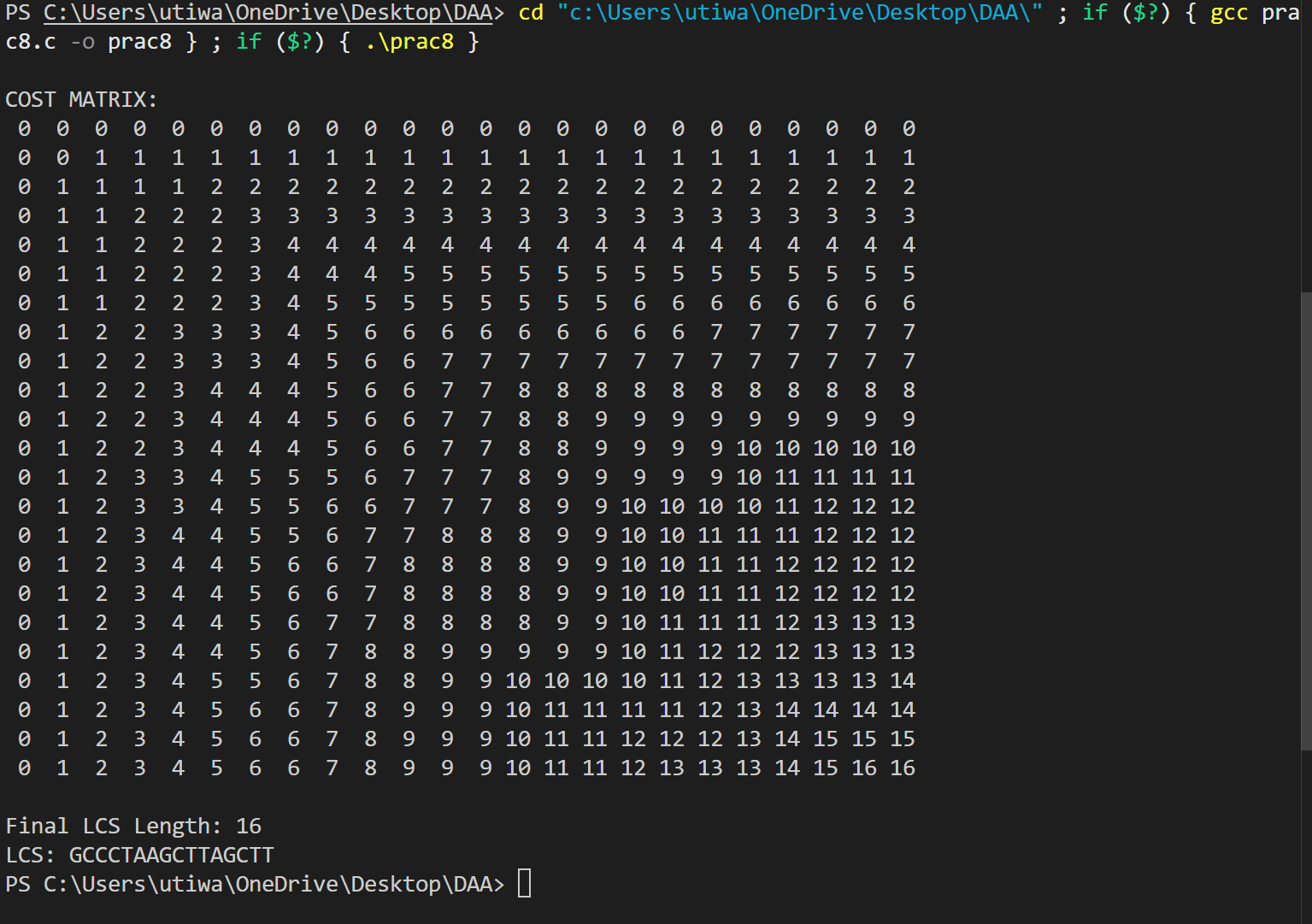
char Y[] = "GACAGCCTACAAGCGTTAGCTTG";

findLCS(X, Y);

return 0;

}

Output



**TASK-2: Longest Repeating Subsequence (LRS)**

#include <stdio.h>

#include <string.h>

int max(int a, int b) { return (a > b) ? a : b; }

void findLRS(char str[]) {

int n = strlen(str);

int dp[n + 1][n + 1];

int i, j;

for (i = 0; i <= n; i++) {

for (j = 0; j <= n; j++) {

if (i == 0 || j == 0)

dp[i][j] = 0;

else if (str[i - 1] == str[j - 1] && i != j)

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

else

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

}

}

int index = dp[n][n];

char lrs[index + 1];

lrs[index] = '\0';

i = n;

j = n;

while (i > 0 && j > 0) {

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

lrs[index - 1] = str[i - 1];

i--;

j--;

index--;

} else if (dp[i - 1][j] > dp[i][j - 1])

i--;

else

j--;

}

printf("Longest Repeating Subsequence: %s\n", lrs);

printf("Length: %d\n", dp[n][n]);

}

int main() {

char str[] = "AABCBDC";

findLRS(str);

return 0;

}

Output:-

