

DAA Lab Pract5

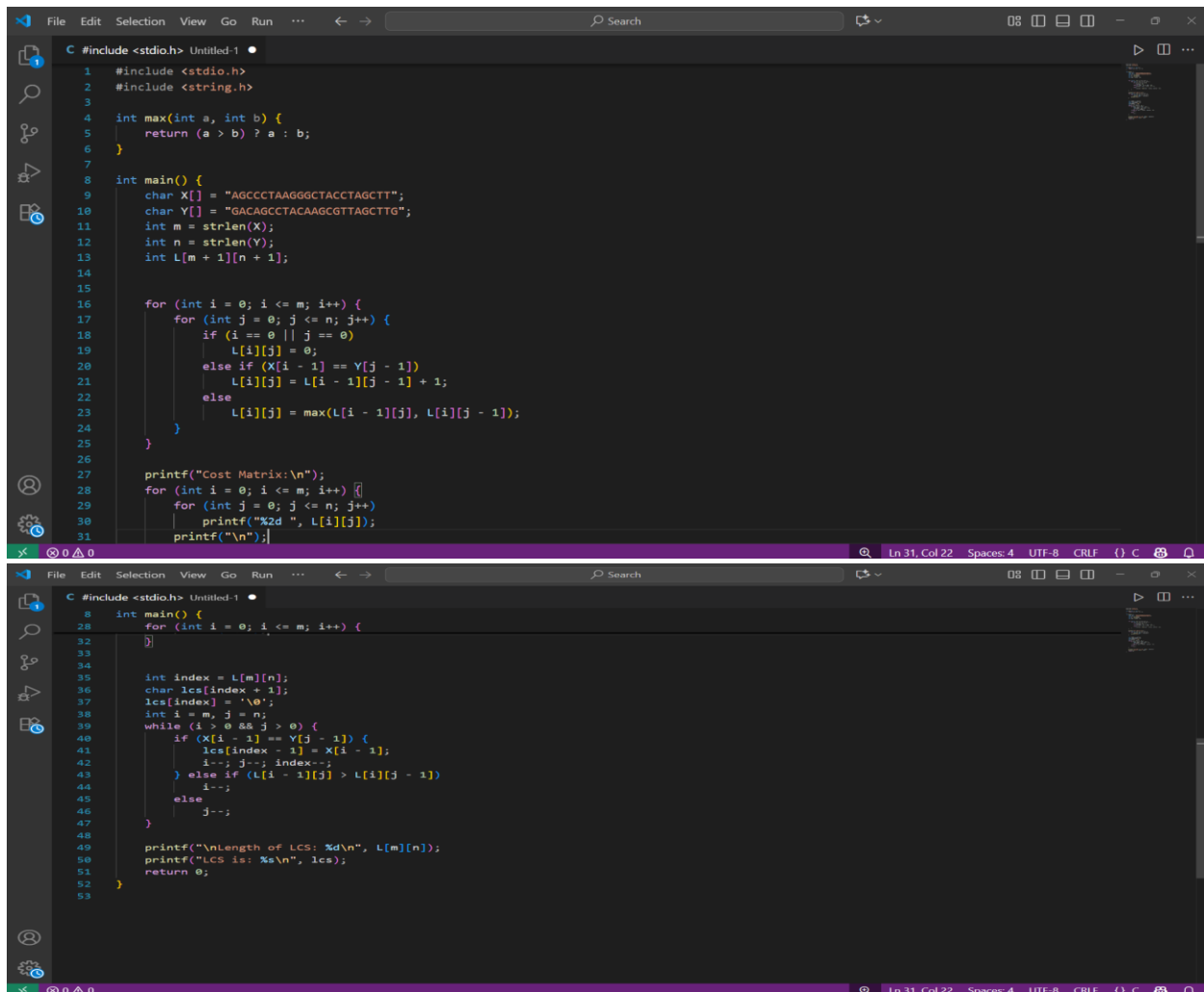
Name: Anshika Tiwari

Roll No : A7_B1_14

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

Task 1 :

Code:



```
#include <stdio.h>
#include <string.h>

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

int main() {
    char X[] = "AGCCCTAAGGGCTACCTAGCTT";
    char Y[] = "GACAGCCTACAAGCGTTAGCTTG";
    int m = strlen(X);
    int n = strlen(Y);
    int L[m + 1][n + 1];

    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0)
                L[i][j] = 0;
            else if (X[i - 1] == Y[j - 1])
                L[i][j] = L[i - 1][j - 1] + 1;
            else
                L[i][j] = max(L[i - 1][j], L[i][j - 1]);
        }
    }

    printf("Cost Matrix:\n");
    for (int i = 0; i <= m; i++)
        for (int j = 0; j <= n; j++)
            printf("%2d ", L[i][j]);
    printf("\n");
}

int main() {
    for (int i = 0; i <= m; i++) {
        // ... (rest of the code is the same as the top screenshot)
    }

    int index = L[m][n];
    char lcs[index + 1];
    lcs[index] = '\0';
    int i = m, j = n;
    while (i > 0 && j > 0) {
        if (X[i - 1] == Y[j - 1]) {
            lcs[index - 1] = X[i - 1];
            i--; j--; index--;
        } else if (L[i - 1][j] > L[i][j - 1])
            i--;
        else
            j--;
    }

    printf("\nLength of LCS: %d\n", L[m][n]);
    printf("LCS is: %s\n", lcs);
    return 0;
}
```

Output:

```
input
Cost Matrix:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
0 1 1 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
0 1 1 2 2 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
0 1 1 2 2 2 3 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5
0 1 1 2 2 2 3 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6
0 1 2 2 3 3 3 4 5 6 6 6 6 6 6 6 6 7 7 7 7 7
0 1 2 2 3 3 3 4 5 6 6 7 7 7 7 7 7 7 7 7 7 7
0 1 2 2 3 4 4 4 5 6 6 7 7 8 8 8 8 8 8 8 8 8
0 1 2 2 3 4 4 4 5 6 6 7 7 8 8 9 9 9 9 9 9 9
0 1 2 2 3 4 4 4 5 6 6 7 7 8 8 9 9 9 10 10 10 10
0 1 2 3 3 4 5 5 6 7 7 7 8 8 9 9 9 10 11 11 11 11
0 1 2 3 3 4 5 5 6 6 7 7 8 8 9 9 10 10 10 11 12 12 12
0 1 2 3 4 4 5 5 6 7 7 8 8 8 9 9 10 10 11 11 12 12 12
0 1 2 3 4 4 5 6 6 7 8 8 8 8 9 9 10 10 11 11 12 12 12
0 1 2 3 4 4 5 6 6 7 8 8 8 8 9 9 10 10 11 11 12 12 12
0 1 2 3 4 4 5 6 7 7 8 8 8 8 9 9 10 11 11 11 12 13 13 13
0 1 2 3 4 5 6 7 8 8 9 9 9 9 9 10 11 12 12 12 13 13 13
0 1 2 3 4 5 6 7 8 8 9 9 10 10 10 10 11 12 13 13 13 14
0 1 2 3 4 5 6 7 8 9 9 9 10 11 11 11 11 12 13 14 14 14 14
0 1 2 3 4 5 6 7 8 9 9 9 10 11 12 12 12 13 14 15 15 15
0 1 2 3 4 5 6 7 8 9 9 9 10 11 12 13 13 13 14 15 16 16

Length of LCS: 16
LCS is: GCCCTAAGCTTAGCTT

...Program finished with exit code 0
Press ENTER to exit console.
```

Task :2

Code :

```
File Edit Selection View Go Run Terminal Help
C #include <stdio.h>
1 #include <string.h>
2 #include <string.h>
3 int max(int a, int b) {
4     return (a > b) ? a : b;
5 }
6 int main() {
7     char S[] = "AACBCDC";
8     int n = strlen(S);
9     int l[n+1][n+1];
10    for (int i = 0; i <= n; i++) {
11        for (int j = 0; j <= n; j++) {
12            if (i == 0 || j == 0)
13                l[i][j] = 0;
14            else if (S[i-1] == S[j-1] && i != j)
15                l[i][j] = 1 + l[i-1][j-1];
16            else
17                l[i][j] = max(l[i-1][j], l[i][j-1]);
18        }
19    }
20    printf("Cost Matrix:\n");
21    for (int i = 0; i <= n; i++) {
22        for (int j = 0; j <= n; j++)
23            printf("%2d ", l[i][j]);
24        printf("\n");
25    }
26    int index = l[n][n];
27    char lrs[index+1];
28    lrs[index] = '\0';
29    int i = n, j = n;
30    while (i > 0 && j > 0) {
31        if (l[i][j] == l[i-1][j-1] + 1 && S[i-1] == S[j-1] && i != j) {
32            lrs[index--] = S[i-1];
33            i--; j--;
34        } else if (l[i-1][j] > l[i][j-1])
35            i--;
36        else
37            j--;
38    }
39    printf("\nLongest Repeating Subsequence: %s\n", lrs);
40    return 0;
41 }
42
```

Code :

```
Cost Matrix:
0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 1
0 1 1 1 1 2 2 2
0 1 1 1 1 2 2 3
0 1 1 2 2 2 2 3
0 1 1 2 2 2 2 3
0 1 1 2 3 3 3 3

Longest Repeating Subsequence: ABC

...Program finished with exit code 0
Press ENTER to exit console.
```

Leet code submission:

The screenshot displays a LeetCode submission for the "Longest Common Subsequence" problem. The submission is marked as "Accepted" with 47/47 test cases passed. The user, Aman_nagmote, submitted it on Oct 17, 2025, at 23:55. The performance metrics show a runtime of 25 ms (beating 45.66% of submissions) and a memory usage of 12.16 MB (beating 77.78% of submissions). A bar chart illustrates the distribution of runtimes for other submissions. The code is written in C and implements a dynamic programming solution. The test result section shows the submission was accepted with a runtime of 0 ms for the provided test case.

```
int longestCommonSubsequence(char* text1, char* text2) {
    int m = strlen(text1);
    int n = strlen(text2);
    int dp[m + 1][n + 1];
    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            dp[i][j] = 0;
        }
    }
    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (text1[i - 1] == text2[j - 1]) {
                dp[i][j] = 1 + dp[i - 1][j - 1];
            } else {
                dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];
            }
        }
    }
    return dp[m][n];
}
```

My git hub link for pract 5 repo : <https://github.com/24tiwaria2-code/DAA->