**Assignment – 7**

**Q1. Implementation to find the Longest Common Subsequence (LCS) of two given sequences and display the intermediate tables.**

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
int lcs\_length(char[], char[], int, int, int\*\*, char\*\*);  
void print\_lcs(char\*\*, char[], int, int);  
  
int main() {  
 char X[20], Y[20], \*\*b;  
 printf("Enter the first sequence: ");  
 scanf("%s", X);  
 printf("Enter the second sequence: ");  
 scanf("%s", Y);  
 int m = strlen(X), n = strlen(Y), \*\*c = (int\*\*)malloc((m + 1) \* sizeof(int\*)), i, j;  
 b = (char\*\*)malloc(m \* sizeof(char\*));  
  
 if (!b || !c) {  
 printf("Memory was not allocated");  
 exit(0);  
 }  
 for (i = 0; i < m; i++) {  
 c[i] = (int\*)malloc((n + 1) \* sizeof(int));  
 b[i] = (char\*)malloc(n \* sizeof(char));  
  
 if (!b[i] || !c[i]) {  
 printf("Memory was not allocated");  
 exit(0);  
 }  
 }  
 c[m] = (int\*)malloc((n + 1) \* sizeof(int));  
  
 if (!c[m]) {  
 printf("Memory was not allocated");  
 exit(0);  
 }  
 printf("The length of LCS is: %d\n\n", lcs\_length(X, Y, m, n, c, b));  
 printf("Cost Matrix after LCS-LENGTH:\n");

for (i = 0; i <= m; i++) {  
 for (j = 0; j <= n; j++) {  
 printf("%d ", c[i][j]);  
 }  
 printf("\n");  
 }  
 printf("\n\nDirection Matrix after LCS-LENGTH:\n");  
  
 for (i = 0; i < m; i++) {  
 for (j = 0; j < n; j++) {  
 printf("%c ", b[i][j]);  
 }  
 printf("\n");  
 }  
 printf("\nThe Longest Common Subsequence of %s and %s is: ", X, Y);  
 print\_lcs(b, X, m - 1, n - 1);  
 printf("\n");  
 return 0;  
}  
  
int lcs\_length(char X[], char Y[], int m, int n, int\*\* c, char\*\* b) {  
 int i, j;  
  
 for (i = 0; i <= m; i++) {  
 c[i][0] = 0;  
 }  
 for (j = 1; j <= n; j++) {  
 c[0][j] = 0;  
 }  
 for (i = 1; i <= m; i++) {  
 for (j = 1; j <= n; j++) {  
 if (X[i - 1] == Y[j - 1]) {  
 c[i][j] = c[i - 1][j - 1] + 1;  
 b[i - 1][j - 1] = 'D';  
 } else if (c[i - 1][j] >= c[i][j - 1]) {  
 c[i][j] = c[i - 1][j];  
 b[i - 1][j - 1] = 'U';  
 } else {  
 c[i][j] = c[i][j - 1];  
 b[i - 1][j - 1] = 'L';  
 }  
 }  
 }  
 return c[m][n];  
}  
void print\_lcs(char\*\* b, char X[], int i, int j) {  
 if (i >= 0 && j >= 0) {  
 switch (b[i][j]) {  
 case 'D':  
 print\_lcs(b, X, i - 1, j - 1);  
 printf("%c", X[i]);  
 break;  
  
 case 'U':  
 print\_lcs(b, X, i - 1, j);  
 break;  
  
 case 'L':  
 print\_lcs(b, X, i, j - 1);  
 break;  
  
 default:  
 printf("Invalid direction component\n");  
 exit(0);  
 }  
 }  
}

**Output**:

Enter the first sequence: 10010101  
Enter the second sequence: 010110110  
The length of LCS is: 6  
  
Cost Matrix after LCS-LENGTH:  
0 0 0 0 0 0 0 0 0 0  
0 0 1 1 1 1 1 1 1 1  
0 1 1 2 2 2 2 2 2 2  
0 1 1 2 2 2 3 3 3 3  
0 1 2 2 3 3 3 4 4 4  
0 1 2 3 3 3 4 4 4 5  
0 1 2 3 4 4 4 5 5 5  
0 1 2 3 4 4 5 5 5 6  
0 1 2 3 4 5 5 6 6 6

Direction Matrix after LCS-LENGTH:  
U D L D D L D D L  
D U D L L D L L D  
D U D U U D L L D  
U D U D D U D D L  
D U D U U D U U D  
U D U D D U D D U  
D U D U U D U U D  
U D U D D U D D U  
  
The Longest Common Subsequence of 10010101 and 010110110 is: 100110