ASSIGNMENT: LCS ALGORITHM

Implementation of the LCS algorithm:

CODE -

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

#define MAX 100

int c[MAX][MAX];
char b[MAX][MAX];

void LCS(char *X, char *Y, 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 (X[i - 1] == Y[j - 1]) {
                c[i][j] = c[i - 1][j - 1] + 1;
                b[i][j] = '\\';
            } else if (c[i - 1][j] >= c[i][j - 1]) {
                c[i][j] = c[i - 1][j];
                b[i][j] = '^';
            } else {
                c[i][j] = c[i][j - 1];
                b[i][j] = '<';
            }
        }
    }
}
//recursive approach to print the LCS
void printLCS(char *X, int i, int j) {
    if (i == 0 || j == 0) return;
    if (b[i][j] == '\\') {
        printLCS(X, i - 1, j - 1);
        printf("%c", X[i - 1]);
    }
```

```
      else if (b[i][j] == '^') {
        printLCS(X, i - 1, j);
      }
      else {
        printLCS(X, i, j - 1);
      }
}

int main() {
    int test;
    printf("Enter number of test cases: ");
    scanf("%d", &test);

    while (test--) {
        char X[MAX], Y[MAX];
        printf("\nEnter first string: ");
        scanf("%s", X);
        printf("Enter second string: ");
        scanf("%s", Y);

        int m = strlen(X);
        int n = strlen(Y);

        LCS(X, Y, m, n);

        printf("Longest Common Subsequence: ");
        printLCS(X, m, n);
        printf("\n");
    }

    return 0;
}
```

OUTPUT -

Enter number of test cases: 6

// general case

Enter first string: BADCXYDA

Enter second string: BXDCADA

Longest Common Subsequence: BDCDA

//same string (LCS -> whole string)

Enter first string: BADCXY

Enter second string: BADCXY

Longest Common Subsequence: BADCXY

// reverse string with no repetitions (LCS -> one character)

Enter first string: BXCDYA

Enter second string: AYDCXB

Longest Common Subsequence: B

//no match (LCS -> not found)

Enter first string: AXC

Enter second string: BYD

Longest Common Subsequence:

// single character no match (LCS -> not found)

Enter first string: A

Enter second string: C

Longest Common Subsequence:

//single character match (LCS -> the character)

Enter first string: C

Enter second string: C

Longest Common Subsequence: C