21BRS1296 – Anika Kamath

Design and Analysis of Algorithms (Lab)

L37+L38

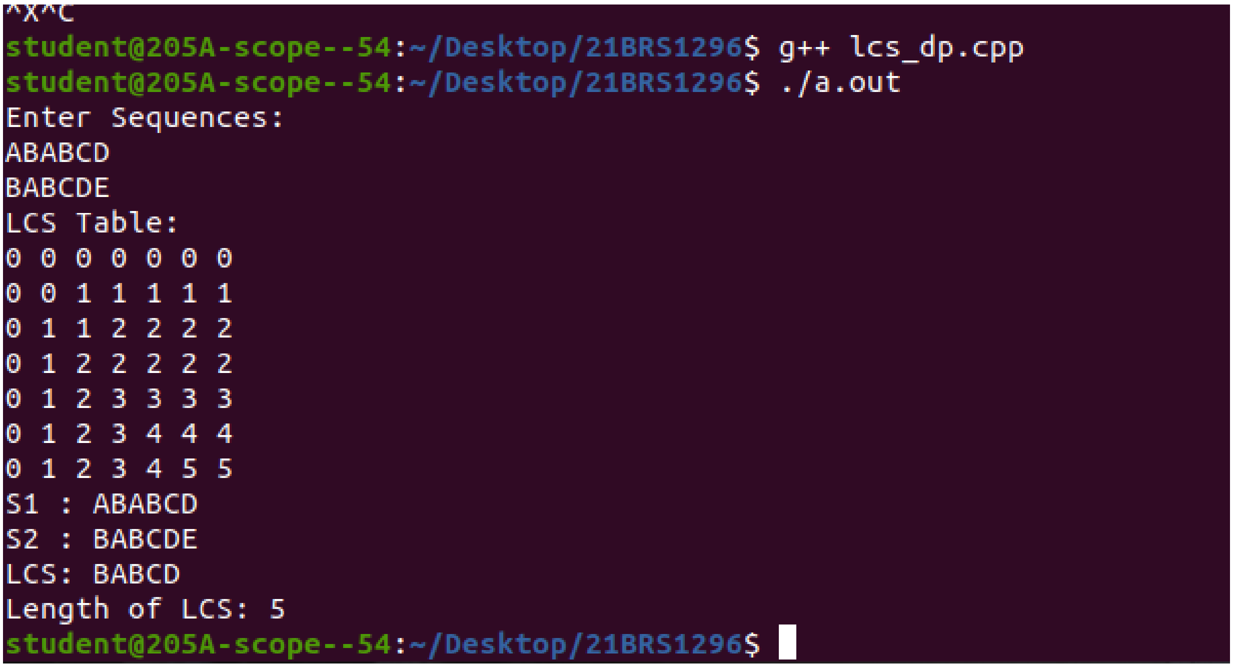
Experiment No.: 6

Q. Longest Common Subarray Sequence – Using Dynamic Programming

Code:

```
#include <iostream>
#include <cstring>

using namespace std;

void longestCommonSubsequence(char *string1, char *string2, int
length1, int length2) {
   int lcsTable[length1 + 1][length2 + 1];

  // Building the matrix in bottom-up way
   for (int i = 0; i <= length1; i++) {
     for (int j = 0; j <= length2; j++) {
       if (i == 0 || j == 0)
         lcsTable[i][j] = 0;
       else if (string1[i - 1] == string2[j - 1])
         lcsTable[i][j] = lcsTable[i - 1][j - 1] + 1;
       else
         lcsTable[i][j] = max(lcsTable[i - 1][j], lcsTable[i][j -
1]);
     }
   }

  int lcsLength = lcsTable[length1][length2];
   char lcs[lcsLength + 1];
   lcs[lcsLength] = '\0';
  int i = length1, j = length2;
   while (i > 0 && j > 0) {
     if (string1[i - 1] == string2[j - 1]) {
       lcs[lcsLength - 1] = string1[i - 1];
       i--;
       j--;
       lcsLength--;
     }

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

  // Printing the LCS table
   cout << "LCS Table: ";
   for (int i = 0; i <= length1; i++) {
     cout << endl;
     for (int j = 0; j <= length2; j++) {
       cout << lcsTable[i][j] << " ";
     }
   }

  // Printing the sub sequences and length of LCS
   cout << "\nString 1: " << string1 << "\nString 2: " << string2 <<
"\nLongest Common Subsequence: " << lcs << "\nLength of LCS: " <<
strlen(lcs) << "\n";
 }

int main() {
   char string1[50], string2[50];
  cout << "Enter Strings: " << endl;
  cin >> string1;
  cin >> string2;
  int length1 = strlen(string1);
  int length2 = strlen(string2);

 longestCommonSubsequence(string1, string2, length1, length2);

  return 0;
}
```

Output:



```
^X^C
student@205A-scope--54:~/Desktop/21BRS1296$ g++ lcs_dp.cpp
student@205A-scope--54:~/Desktop/21BRS1296$ ./a.out
Enter Sequences:
ABABCD
BABCDE
LCS Table:
0 0 0 0 0 0 0
0 0 1 1 1 1 1
0 1 1 2 2 2 2
0 1 2 2 2 2 2
0 1 2 3 3 3 3
0 1 2 3 4 4 4
0 1 2 3 4 5 5
S1 : ABABCD
S2 : BABCDE
LCS: BABCD
Length of LCS: 5
student@205A-scope--54:~/Desktop/21BRS1296$
```