Keegan Vaz
SE CMPNB
42

AOA EXPERIMENT 8

Aim: Write a program to find **Longest Common Subsequence** for two sequences using **DynamicProgramming**

Problem statement:
Write a program to take input two string X and Y and find its LCS.

Input:
Give the below two strings as input:

X="HYDERABAD" ,
Y="AHMEDNAGAR"

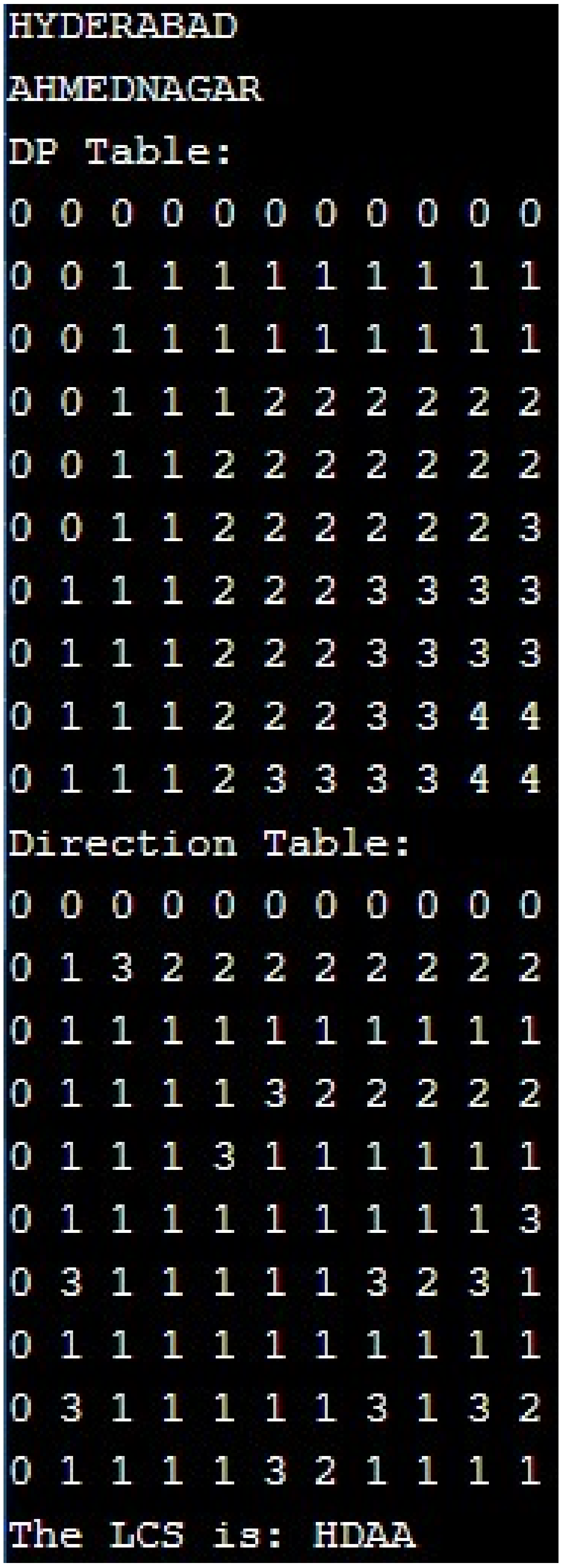
Output: Display the matrix generated for LCS and also the matrix that stores directions. Display the final LCS.
(Paste your code and output below)

Code:

```
#include <bits/stdc++.h>
using namespace std;

int dp[3003][3003],b[3003][3003];

void print_lcs(int b[3003][3003],string s,int i,int j){
    if (i==0 || j==0) return;

    if (b[i][j]==3){
        print_lcs(b,s,i-1,j-1);
        cout<<s[i-1];
    }else if (b[i][j]==1){
        print_lcs(b,s,i-1,j);
    }else{
        print_lcs(b,s,i,j-1);
    }
}

void lcs(string s,string t,int n,int m){
    int i,j;
    for(i=0;i<m+1;i++){
        dp[0][i]=0;
        b[0][i]=0;
    }
    for(i=0;i<n+1;i++){
        dp[i][0]=0;
        b[i][0]=0;
    }
    for(i=1;i<=n;i++){
        for(j=1;j<=m;j++){
            if (s[i-1]==t[j-1]){
                dp[i][j]=1+dp[i-1][j-1];
                b[i][j]=3;
            }else if (dp[i-1][j]>=dp[i][j-1]){
                dp[i][j]=dp[i-1][j];
                b[i][j]=1;
            }else{
                dp[i][j]=dp[i][j-1];
                b[i][j]=2;
            }
        }
    }
    cout<<"DP Table: "<<endl;
    for(i=0;i<=n;i++){
        for(j=0;j<=m;j++){
            cout<<dp[i][j]<<"";
        }
        cout<<endl;
    }
    cout<<"Direction Table: "<<endl;
    for(i=0;i<=n;i++){
        for(j=0;j<=m;j++){
            cout<<b[i][j]<<"";
        }
        cout<<endl;
    }
    cout<<"The LCS is: ";
    print_lcs(b,s,n,m);
    cout<<endl;
}
int main()
{
    string s,t;
    cin>>s;
    cin>>t;
    lcs(s,t,s.size(),t.size());
    return 0;
}
```

Keegan Vaz
SE CMPNB
42

Output:



```
HYDERABAD
AHMEDNAGAR
DP Table:
0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 1 1
0 0 1 1 1 2 2 2 2 2 2
0 0 1 1 2 2 2 2 2 2 2
0 0 1 1 2 2 2 2 2 2 3
0 1 1 1 2 2 2 3 3 3 3
0 1 1 1 2 2 2 3 3 3 3
0 1 1 1 2 2 2 3 3 4 4
0 1 1 1 2 3 3 3 3 4 4
Direction Table:
0 0 0 0 0 0 0 0 0 0 0
0 1 3 2 2 2 2 2 2 2 2
0 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 3 2 2 2 2 2
0 1 1 1 3 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 3
0 3 1 1 1 1 1 3 2 3 1
0 1 1 1 1 1 1 1 1 1 1
0 3 1 1 1 1 1 3 1 3 2
0 1 1 1 1 3 2 1 1 1 1
The LCS is: HDAA
```