**FNU Hassan**

**Assignment 5**

***GitHub Link:***

[***https://github.com/FNU-Hassan/Advance-Algorithm-Assignment-5***](https://github.com/FNU-Hassan/Advance-Algorithm-Assignment-5)

/\*solution to

find length of the

longest common substring \*/

#include <iostream>

#include <string.h>

using namespace std;

/\* Returns length of longest

common substring of X[0..m-1]

and Y[0..n-1] \*/

int LCSubStr(char\* X, char\* Y, int m, int n)

{

// Create a table to store

// lengths of longest

// common suffixes of substrings.

// Note that LCSuff[i][j] contains

// length of longest common suffix

// of X[0..i-1] and Y[0..j-1].

int LCSuff[m + 1][n + 1];

int result = 0; // To store length of the

// longest common substring

/\* Following steps build LCSuff[m+1][n+1] in

bottom up fashion. \*/

for (int i = 0; i <= m; i++)

{

for (int j = 0; j <= n; j++)

{

// The first row and first column

// entries have no logical meaning,

// they are used only for simplicity

// of program

if (i == 0 || j == 0)

LCSuff[i][j] = 0;

else if (X[i - 1] == Y[j - 1]) {

LCSuff[i][j] = LCSuff[i - 1][j - 1] + 1;

result = max(result, LCSuff[i][j]);

}

else

LCSuff[i][j] = 0;

}

}

return result;

}

// Driver code

int main()

{

char X[] = "BABA";

char Y[] = "ABAB";

int m = strlen(X);

int n = strlen(Y);

cout << "Length of Longest Common Substring is "

<< LCSubStr(X, Y, m, n);

return 0;

}

