**Assignment 5**

**GitHub Link:**

[https://github.com/sufiyanjunaidi13/Advance\_Algorithm\_Assignment-5 (github.com)](https://github.com/sufiyanjunaidi13/Advance_Algorithm_Assignment-5)

#include <iostream>

#include <string>

int longestCommonSubstring(std::string X, std::string Y)

{

int m = X.length();

int n = Y.length();

// Create a table to store lengths of longest common suffixes of substrings

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

// Initialize result to 0

int result = 0;

// Fill the LCSuff table in a bottom-up manner

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

{

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

{

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

}

else

LCSuff[i][j] = 0;

}

}

// Return the length of the longest common substring

return result;

}

int main()

{

std::string X = "BABA";

std::string Y = "ABAB";

std::cout << "Length of Longest Common Substring is "

<< longestCommonSubstring(X, Y) << std::endl;

return 0;

}

