// Java implementation to find the

// length of longest common subsequence

// which contains all vowel characters

public class LeastVowel

{

// function to check whether 'ch'

// is a vowel or not

static boolean isVowel(char ch)

{

if (ch == 'a' || ch == 'e' ||

ch == 'i' || ch == 'o' ||

ch == 'u')

return true;

return false;

}

// function to find the length of

// longest common subsequence which

// contains all vowel characters

static int lcs(String X, String Y,

int m, int n)

{

int L[][] = new int[m + 1][n + 1];

int i, j;

// Following steps build L[m+1][n+1]

// in bottom up fashion. Note that

// L[i][j] contains length of LCS of

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

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

{

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

{

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

L[i][j] = 0;

else if ((X.charAt(i - 1) == Y.charAt(j - 1)) &&

isVowel(X.charAt(i - 1)))

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

else

L[i][j] = Math.max(L[i - 1][j],

L[i][j - 1]);

}

}

// L[m][n] contains length of LCS

// for X[0..n-1] and Y[0..m-1]

// which contains all vowel characters

return L[m][n];

}

// Driver Code

public static void main(String[] args)

{

String X = "aieef";

String Y = "klaief";

int m = X.length();

int n = Y.length();

System.out.println("Length of LCS = " +

lcs(X, Y, m, n));

}

}