**Longest Common Subsequence**

**Medium**

Given two strings, find the length of longest subsequence present in both of them. Both the strings are in **uppercase**latin alphabets.

**Example 1:**

**Input:**

A = 6, B = 6

str1 = ABCDGH

str2 = AEDFHR

**Output:** 3

**Explanation:** LCS for input strings “ABCDGH” and “AEDFHR” is “ADH” of length 3.

**Example 2:**

**Input:**

A = 3, B = 2

str1 = ABC

str2 = AC

**Output:** 2

**Explanation:** LCS of "ABC" and "AC" is "AC" of length 2.

**Expected Time Complexity** : O(|str1|\*|str2|)  
**Expected Auxiliary Space**: O(|str1|\*|str2|)

**Constraints:**  
1<=size(str1),size(str2)<=103

**Company Tags**

[**Paytm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Paytm) [**VMWare**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=VMWare) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Citrix**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Citrix) [**MakeMyTrip**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=MakeMyTrip)

**Topic Tags**

[**Dynamic Programming**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Dynamic%20Programming) [**Algorithms**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Algorithms)

//{ Driver Code Starts

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class CodingMaxima {

public static void main (String[] args) {

Scanner sc=new Scanner(System.in);

int test=sc.nextInt();

while(test-- > 0){

int p=sc.nextInt(); // Take size of both the strings as input

int q=sc.nextInt();

String s1=sc.next(); // Take both the string as input

String s2=sc.next();

Solution obj = new Solution();

System.out.println(obj.lcs(p, q, s1, s2));

}

}

}

// } Driver Code Ends

class Solution

{

//Function to find the length of longest common subsequence in two strings.

static int lcs(int x, int y, String s1, String s2)

{

int[][] arr=new int[x+1][y+1];

for(int i=1;i<=x;i++){

for(int j=1;j<=y;j++){

if(s1.charAt(i-1)==s2.charAt(j-1)){

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

}

else

{

arr[i][j]=Math.max(arr[i-1][j],arr[i][j-1]);

}

}

}

return arr[x][y];

}

}