**Minimum number of deletions and insertions.**

Given two strings ‘str1’ and ‘str2’ of size m and n respectively. The task is to remove/delete and insert minimum number of characters from/in str1 so as to transform it into str2. It could be possible that the same character needs to be removed/deleted from one point of str1 and inserted to some another point.

**Input:**

The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. The first line of each test case contains integers N and M denoting the length of the strings str1 and str2. Both the strings consist of only small letter alphabets.

The second line of each test case contains the strings str1 and str2.

**Output:**

Print the total no of insertions and deletions to be done to convert str1 to str2. Output the answer in a new line.

**Constraints:**

1<= T <=100

1<= N, M <=1000

**Example:**

**Input:**

1

4 3

heap pea

**Output:**

3

* SOLUTION

#include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

int n,m;

cin>>n>>m;

string s1,s2;

cin>>s1>>s2;

int dp[n+1][m+1],i,j;

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

dp[i][0]=0;

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

dp[0][i]=0;

for(i=1;i<=n;i++)

{

for(j=1;j<=m;j++)

{

if(s1[i-1]==s2[j-1])

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

else

dp[i][j]=max(dp[i-1][j],dp[i][j-1]);

}

}

cout<<n+m-dp[n][m]\*2<<"\n";

}

}

[Minimum number of deletions and insertions to transform one string into another](https://practice.geeksforgeeks.org/editorial.php?pid=2228#trackTitle)

Given two strings 'str1' and 'str2' of size m and n respectively. The task is to remove/delete and insert minimum number of characters from/in str1 so as to transform it into str2. It could be possible that the same character needs to be removed/deleted from one point of str1 and inserted to some another point.  
  
  
**Examples:**

Input : str1 = "heap", str2 = "pea"   
Output : Minimum Deletion = 2 and  
 Minimum Insertion = 1  
**p** and **h** deleted from **heap**  
Then, **p** is inserted at the beginning  
One thing to note, though **p** was required yet  
it was removed/deleted first from its position and  
then it is inserted to some other position.  
Thus, **p** contributes one to the **deletion\_count**  
and one to the **insertion\_count**.  
  
Input : str1 = "geeksforgeeks", str2 = "geeks"  
Output : Minimum Deletion = 8  
 Minimum Insertion = 0

A **simple solution** is to consider all subsequences of str1 and for each subsequence calculate minimum deletions and insertions so as to transform it into str2. A very complex method and the time complexity of this solution is exponential.  
  
An **efficient approach** uses the concept of [finding the length of the longest common subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-4-longest-common-subsequence/)of the given two sequences.  
  
**Algorithm:**

-->**str1** and **str2** be the given strings.  
-->**m** and **n** be their lengths respectively.  
-->**len** be the length of the longest   
 common subsequence of **str1** and **str2**  
-->// minimum number of deletions   
 **minDel** = m - len  
-->// minimum number of Insertions   
 **minInsert** = n - len

**Output:**

Minimum number of deletions = 2  
Minimum number of insertions = 1

**Time Complexity:** O(m \* n)