75 Days of Code
Day 64
Problem no : Leetcode1143
Problem Title :Longest Common Subsequence
Problem type : DP

.
Given two strings text1 and text2, return the length of their longest common subsequence. If there is no common subsequence, return 0.

A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, "ace" is a subsequence of "abcde".
A common subsequence of two strings is a subsequence that is common to both strings.

Example 1:

Input: text1 = "abcde", text2 = "ace"
Output: 3
Explanation: The longest common subsequence is "ace" and its length is 3.
Example 2:

Input: text1 = "abc", text2 = "abc"
Output: 3
Explanation: The longest common subsequence is "abc" and its length is 3.
Example 3:

Input: text1 = "abc", text2 = "def"
Output: 0
Explanation: There is no such common subsequence, so the result is 0.

```typescript
function longestCommonSubsequence(text1: string, text2: string): number {
    const maxElementLen :number[] = new Array(text2.length+1).fill(0);
    let prev:number ;
    let cur :number;

    for(let text1Index = 1;text1Index<=text1.length ;text1Index++){

        prev =0;
        for(let text2Index=1; text2Index <=text2.length ;text2Index++){
            cur = maxElementLen[text2Index];
                if(text1[text1Index-1] === text2[text2Index-1]){
                    maxElementLen[text2Index]= 1+prev;
                }else{
                     maxElementLen[text2Index]= Math.max(maxElementLen[text2Index-1],maxElementLen[text2Index])

                }
                prev = cur;
        }
    }
    return maxElementLen[text2.length]
}
```