Luis Erique Franco
410921353

Base case:
algorithm To find longest common subsequence. This algorithm was explained in class， it uses dynamic orogramming To create a matrix. The matrix is filled with values according to the rules given.

First row and column are filled with O.
Next values are the highest value from the up and left 1 if the row and column coincide.
the Last value in the matrix is the longest common Subsequence

1。Create an array of size 30 for the characters
```
    store the value of i. Iterate over the sequence getting the value:
        First get the max value in the character array from 0 to j
        Find if the current value is the max and if it is replace it on
         the chacter array and in j
    Return j
    time omplexity O(n)
     space complexity O(1)
```

2。For the longest subSequence in one Sequence：
```
    —copy sequence in a second array
    —sort array
    —apply longest common subsequence
```
space complexity O(nˆ2)
time complexity O(nˆ2)

3。Create a character array of length equal to the length of the number gotten from lasr algorithm plus 1
Traverse the 2D array starting from L[m][n]. Do following for every cell L[i][j]
  a) If characters in column anf row corresponding to L[i][j] are same , then include this character.
  b) Else compare values of L[i-1][j] and L[i][j-1] and go in direction of greater value.
space complexity O(nˆ2)
time complexity O(nˆ2)