Question [2 marks] Consider the following pseudocode for finding the length of the longest common subsequence of two strings:

```
Algorithm LCS(string x, string y)
    n ← x.length();
    m ← y.length();
    for i from 1 to n do llcs[i,0] ← 0;
    for j from 1 to m do llcs[0,j] ← 0
    for i from 1 to n do
        for j from 1 to m do
            if x[i] = y[j] then
                llcs[i, j] ← llcs[i-1, j-1] +1;
            else if llcs[i-1, j] > llcs[i, j-1] then
                llcs[i, j] ← llcs[i-1, j];
            else
                llcs[i, j] ← llcs[i, j-1];
    return llcs;
```

a. Now consider the two strings ‘abcdbb’ and ‘cbacbaaba’. Use the algorithm above to determine the length of their longest common subsequence? Fill out the dynamic programming table below that represents the array llcs[,].

llcs[,]		c	b	a	c	b	a	a	b	a
	0	0	0	0	0	0	0	0	0	0
a	0	0	0	1	1	1	1	1	1	1
b	0	0	1	1	1	2	2	2	2	2
c	0	1	1	1	2	2	2	2	2	2
d	0	1	1	1	2	2	2	2	2	2
b	0	1	2	2	2	3	3	3	3	3
b	0	1	2	2	2	3	3	3	4	4

b. What is the longest common subsequence of the two strings found in part a.?