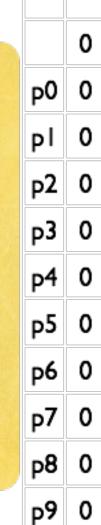```
function LCSLength (X[1..m],Y[1..n]) {
  C = array (0..m, 0..n)
   for row=0..m
     C[row,0] = 0;
   for col =0..n
     C[0,col] = 0
   for row=1..m
     for col = 1..n
      if X[row] = Y[col]
        C[row,col] = C[row-1, col-1] +1
      else
        C[row,col] = max(C[row, col-1], C[row-1, col])
   return C[row, col]
```

```
function backTrace (C[0..m, 0..n], X[1..m],Y[1..n], row,
col) {
   if row=0 or col=0
     return ""
   else if X[row] = Y[col]
     return backTrace(C, X,Y, row-1, col-1) +X[row]
   else
       if C[row, col-1] > C[row-1, col]
           return backTrace(C, X, Y, row, col-1)
       else
           return backTrace(C, X, Y, row-1, col)
```

		c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
		0	0	0	0	0	0	0	0	0	0	0	0
p0	0	1	1	1	1	1	1	1	1	1	1	1	1
p1	0	1	1	2	2	2	2	2	2	2	2	2	2
p2	0	1	1	2	3	3	3	3	3	3	3	3	3
p3	0	1	1	2	3	4	4	4	4	4	4	4	4
p4	0	1	1	2	3	4	5	5	5	5	5	5	5
p5	0	1	1	2	3	4	5	5	6	6	6	6	6
p6	0	1	1	2	3	4	5	5	6	6	7	7	7
p7	0	1	1	2	3	4	5	5	6	7	7	7	7
p8	0	1	1	2	3	4	5	5	6	7	7	8	8
p9	0	1	1	2	3	4	5	6	6	7	7	8	9