LCS

Etapa 3 - relaţia de recurenţă

- `Lmax[i][0]=0, i=1,...,n`
- `Lmax[0][j]=0, j=1,...,m`
- `Lmax[i][j]=Lmax[i-1][j-1]+1, daca X[i]=Y[j]`
- `Lmax[i][j]=max{Lmax[i-1][j], Lmax[i][j-1],`

```
for (i = 0; i <= m; i++)
        for (j = 0; j <= n; j++)
                if((i == 0) || (j == 0)) lcs[i][j] = 0;
                else
                        if(s[i-1] == t[j-1])  lcs[i][j] = 1 + lcs[i-1][j-1];
                        else
                           lcs[i][j] = lcs[i-1][j]>lcs[i][j-1] ? lcs[i-1][j] : lcs[i][j-1];

printf("Lungimea maxima a subsirului comun: %d\n" , lcs[n][m]);

i = m; j = n;

printf("Subsirul maximal comun:\n");
while(lcs[i][j] != 0)
        if (s[i-1] == t[j-1]) { printf("%c " , s[i-1]); i--;  j--; }
        else
                if (lcs[i][j] == lcs[i-1][j]) i--;
                else j--;
```

Primul sir: program
Al doilea sir: roman

0 0 0 0 0 0
0 0 0 0 0 0
0 1 1 1 1 1
0 1 2 2 2 2
0 1 2 2 2 2
0 1 2 2 2 2
0 1 2 2 3 3
0 1 2 3 3 3