

LCS Algorithm

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string X = "G TTCCTAATA";
    string Y = "CGATAATTGAGA";
    int L [11][13];

    int i;
    int j;
    int n =11;
    int m = 13;

    // initialize the first column to zero
    for (i = 0;i < n; i++)
    {
        L[i][0]=0;
    }
    // initialize the first rows to zero
    for (j = 1;j < m; j++)
    {
        L[0][j]=0;
    }
    for (i = 1;i <= n-1; i++)
    {
        for (j = 1;j <= m-1; j++) //
        {
            if (X[i - 1]==Y[j - 1])
            {
                L[i][j]=L[i-1][j-1] + 1;
                cout << "X[i - 1]" << i - 1 << " " << X[i - 1] << endl;
                cout << "Y[j - 1]" << j - 1 << " " << Y[j - 1]<< endl;
            }
            else
            {
                L[i][j]=max(L[i-1][j],L[i][j-1]);
            }
        }
    }
    // print the array
    i=j=0;
    while ( i < n )
```

```

        {
            cout << L[i][j];
            ( j == m - 1 ) ? cout << '\n' : cout << '\t' ;
            ( j == m - 1 ) ? i++,j=0 : j++;
        }
    return 0;
}

```

//output

```

X[i - 1]0 G
Y[j - 1]1 G
X[i - 1]0 G
Y[j - 1]8 G
X[i - 1]0 G
Y[j - 1]10 G
X[i - 1]1 T
Y[j - 1]3 T
X[i - 1]1 T
Y[j - 1]6 T
X[i - 1]1 T
Y[j - 1]7 T
X[i - 1]2 T
Y[j - 1]3 T
X[i - 1]2 T
Y[j - 1]6 T
X[i - 1]2 T
Y[j - 1]7 T
X[i - 1]3 C
Y[j - 1]0 C
X[i - 1]4 C
Y[j - 1]0 C
X[i - 1]5 T
Y[j - 1]3 T
X[i - 1]5 T
Y[j - 1]6 T
X[i - 1]5 T
Y[j - 1]7 T
X[i - 1]6 A
Y[j - 1]2 A
X[i - 1]6 A
Y[j - 1]4 A
X[i - 1]6 A
Y[j - 1]5 A
X[i - 1]6 A
Y[j - 1]9 A

```

X[i - 1]6 A
Y[j - 1]11 A
X[i - 1]7 A
Y[j - 1]2 A
X[i - 1]7 A
Y[j - 1]4 A
X[i - 1]7 A
Y[j - 1]5 A
X[i - 1]7 A
Y[j - 1]9 A
X[i - 1]7 A
Y[j - 1]11 A
X[i - 1]8 T
Y[j - 1]3 T
X[i - 1]8 T
Y[j - 1]6 T
X[i - 1]8 T
Y[j - 1]7 T
X[i - 1]9 A
Y[j - 1]2 A
X[i - 1]9 A
Y[j - 1]4 A
X[i - 1]9 A
Y[j - 1]5 A
X[i - 1]9 A
Y[j - 1]9 A
X[i - 1]9 A
Y[j - 1]11 A

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	2	2	2	2	2	2	2	2	2
0	0	1	1	2	2	2	3	3	3	3	3	3
0	1	1	1	2	2	2	3	3	3	3	3	3
0	1	1	1	2	2	2	3	3	3	3	3	3
0	1	1	1	2	2	2	3	4	4	4	4	4
0	1	1	2	2	3	3	3	4	4	5	5	5
0	1	1	2	2	3	4	4	4	4	5	5	6
0	1	1	2	3	3	4	5	5	5	5	5	6
0	1	1	2	3	4	4	5	5	5	6	6	6