```
 1: //Longest Common Subsequence
 2: #include<iostream>
 3: #include<fstream>
 4: #include<stdlib.h>
 5: #include<time.h>
 6: #include<iomanip>
 7: #include<bits/stdc++.h>
 8:
 9: using namespace std;
10:
11: long int cnt=0;
12:
13: //LCS - Dynamic Programming
14: int LCS_Length_DP(char *x, char *y, int m, int n,char **b,
    int **c)
15: {
16:     //m and n are the length of x and y resp.
17:
18:     for(int i=1;i<=m;i++)
19:     {
20:         c[i][0] = 0;
21:         cnt++;
22:     }
23:     for(int j=0;j<=n;j++)
24:     {
25:         c[0][j] = 0;
26:         cnt++;
27:     }
28:
29:     for(int i=1;i<=m;i++)
30:     {
31:         for(int j=1;j<=n;j++)
32:         {
33:             cnt++;
34:             if(x[i]==y[j])
35:             {
36:                 c[i][j] = c[i-1][j-1]+1;
37:                 b[i][j] = 'C';
38:             }
39:             else if(c[i-1][j]>=c[i][j-1])
40:             {
41:                 c[i][j] = c[i-1][j];
```

```
42:                 b[i][j] = 'U';
43:             }
44:             else
45:             {
46:                 c[i][j] = c[i][j-1];
47:                 b[i][j] = 'L';
48:             }
49:         }
50:     }
51:
52:     return c[m][n];
53:
54: }
55:
56: //Print LCS
57: void PrintLCS(char **b,char *x, int i, int j)
58: {
59:     if(i==0 || j==0)
60:     {
61:         return;
62:     }
63:
64:     if(b[i][j]=='C')
65:     {
66:         PrintLCS(b,x,i-1,j-1);
67:         cout<<x[i];
68:     }
69:     else if(b[i][j]=='U')
70:     {
71:         PrintLCS(b,x,i-1,j);
72:     }
73:     else
74:     {
75:         PrintLCS(b,x,i,j-1);
76:     }
77:
78: }
79:
80:
81: int max(int a, int b)
82: {
83:     return (a > b)? a : b;
```

```
 84: }
 85:
 86: //LCS - Divide & Conquer
 87: int LCS_Length_DC( char *X, char *Y, int m, int n )
 88: {
 89:     cnt++;
 90:     if (m == 0 || n == 0)
 91:         return 0;
 92:     if (X[m-1] == Y[n-1])
 93:         return 1 + LCS_Length_DC(X, Y, m-1, n-1);
 94:     else
 95:         return max(LCS_Length_DC(X, Y, m, n-1),
    LCS_Length_DC(X, Y, m-1, n));
 96: }
 97:
 98:
 99:
100:
101: int main()
102: {
103:     cout<<showpoint<<setprecision(12);
104:
105:     int n,m;
106:     char *x;
107:     char *y;
108:
109:     cout<<"\nEnter Length of First String: ";
110:     cin>>m;
111:     cout<<"\nEnter Length of Second String: ";
112:     cin>>n;
113:
114:
115:
116:     x = new char[m+1];
117:     y = new char[n+1];
118:
119:
120:     ofstream outf;
121:     ifstream inf;
122:
123:     srand((long int)clock());
124:
```

```
125:     //Loading numbers to input file
126:     char t;
127:     outf.open("in1.txt");
128:     for(int i=1;i<=m;i++)
129:     {
130: //       while(((t=(rand()%255)+1)<65 || (t>90&&t<97) ||
     t>122)); //For any alphabets
131: //       while(((t=(rand()%255)+1)<65 || (t>68&&t<97) ||
     t>100)); //For only a,b,c,d & A,B,C,D
132:         while(((t=(rand()%255)+1)<'A') || (t>'A'&&t<'C') ||
     (t>'C'&&t<'G') || (t>'G'&&t<'T') ||t>'T'); //For DNA Sequence
133:         outf<<"\t"<<t;
134:     }
135:     outf.close();
136:     outf.open("in2.txt");
137:     for(int i=1;i<=n;i++)
138:     {
139: //       while(((t=(rand()%255)+1)<65 || (t>90&&t<97) ||
     t>122)); //For any alphabets
140: //       while(((t=(rand()%255)+1)<65 || (t>68&&t<97) ||
     t>100)); //For only a,b,c,d & A,B,C,D
141:         while(((t=(rand()%255)+1)<'A') || (t>'A'&&t<'C') ||
     (t>'C'&&t<'G') || (t>'G'&&t<'T') ||t>'T'); //For DNA Sequence
142:         outf<<"\t"<<t;
143:     }
144:     outf.close();
145:
146:     //Reading input in array from input file
147:
148:     inf.open("in1.txt");
149:     for(int i=1;i<=m;i++)
150:     {
151:         inf>>x[i];
152:     }
153:     inf.close();
154:     x[m+1] = '\0';
155:     x[0]=' ';
156:     inf.open("in2.txt");
157:     for(int i=1;i<=n;i++)
158:     {
159:         inf>>y[i];
160:     }
```

```
161:     inf.close();
162:     y[n+1] = '\0';
163:     y[0]=' ';
164:
165:     cout<<"\n\nX: "<<x;
166:     cout<<"\n\nY: "<<y;
167:
168:
169:     char **b; //U - Up, L - Left & C - Cross
170:     int **c;
171:
172:     b = new char*[m+1];
173:     for(int i=0;i<=m;i++)
174:         b[i] = new char[n+1];
175:
176:     c = new int*[m+1];
177:     for(int i=0;i<=m;i++)
178:         c[i] = new int[n+1];
179:
180:     int lcs_length=0;
181:
182:     //LCS - Divide & Conquer
183:     cnt = 0;
184:     lcs_length = LCS_Length_DC(x,y,m,n);
185:
186:     cout<<"\n\nLongest Common Sub Sequence Length (D & C):
    "<<lcs_length;
187:     cout<<"\nNumber of Active Operations: "<<cnt;
188:
189:     //LCS - Dynamic Programming
190:
191:     cnt=0;
192:     lcs_length = LCS_Length_DP(x,y,m,n,b,c);
193:
194:     cout<<"\n\nLongest Common Sub Sequence Length (DP):
    "<<lcs_length;
195:     cout<<"\nNumber of Active Operations: "<<cnt;
196:     cout<<"\n\nLCS: ";
197:     PrintLCS(b,x,m,n);
198:
199:
200:     delete(b);
```

```
201:     delete(c);
202:
203: }
204:
```