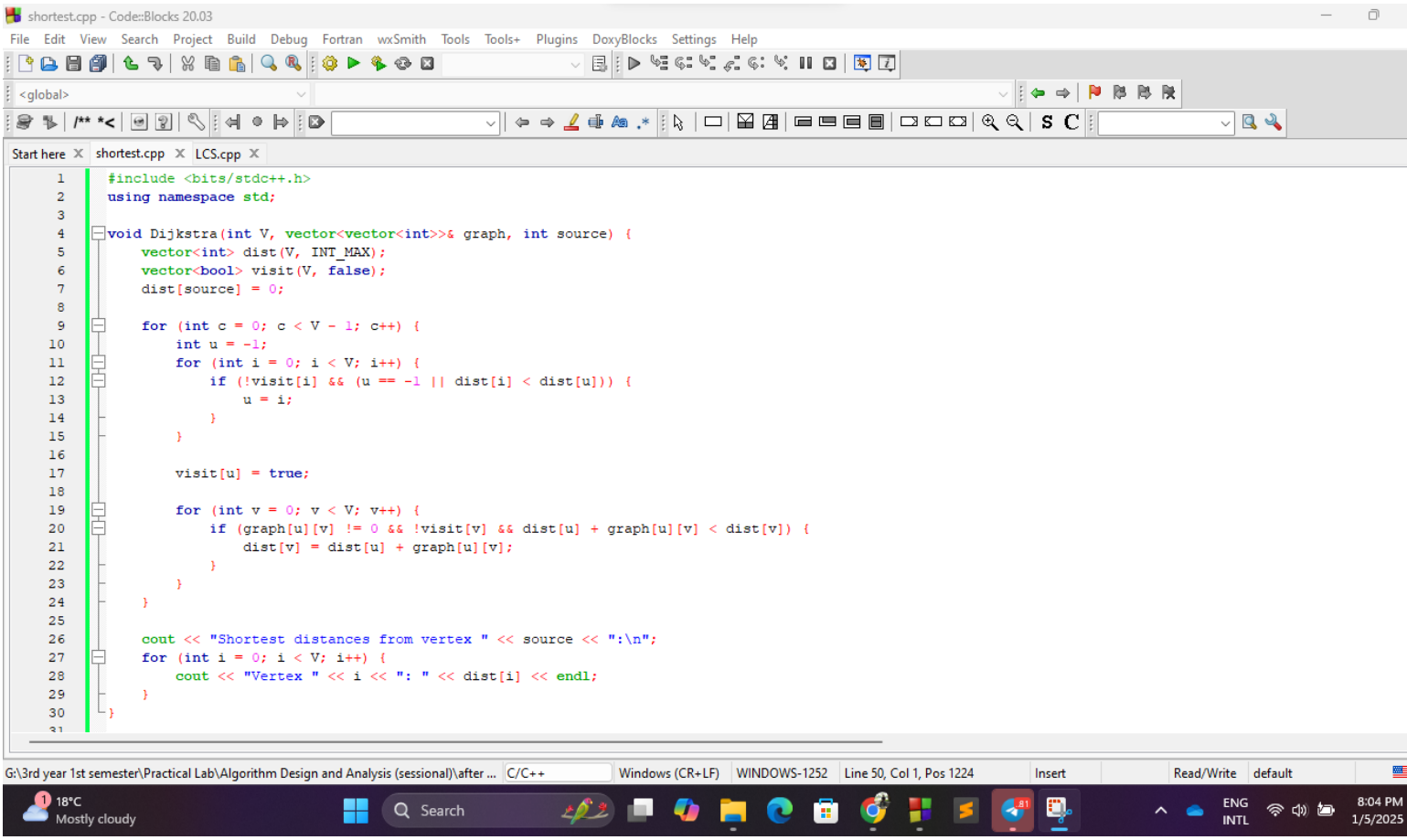
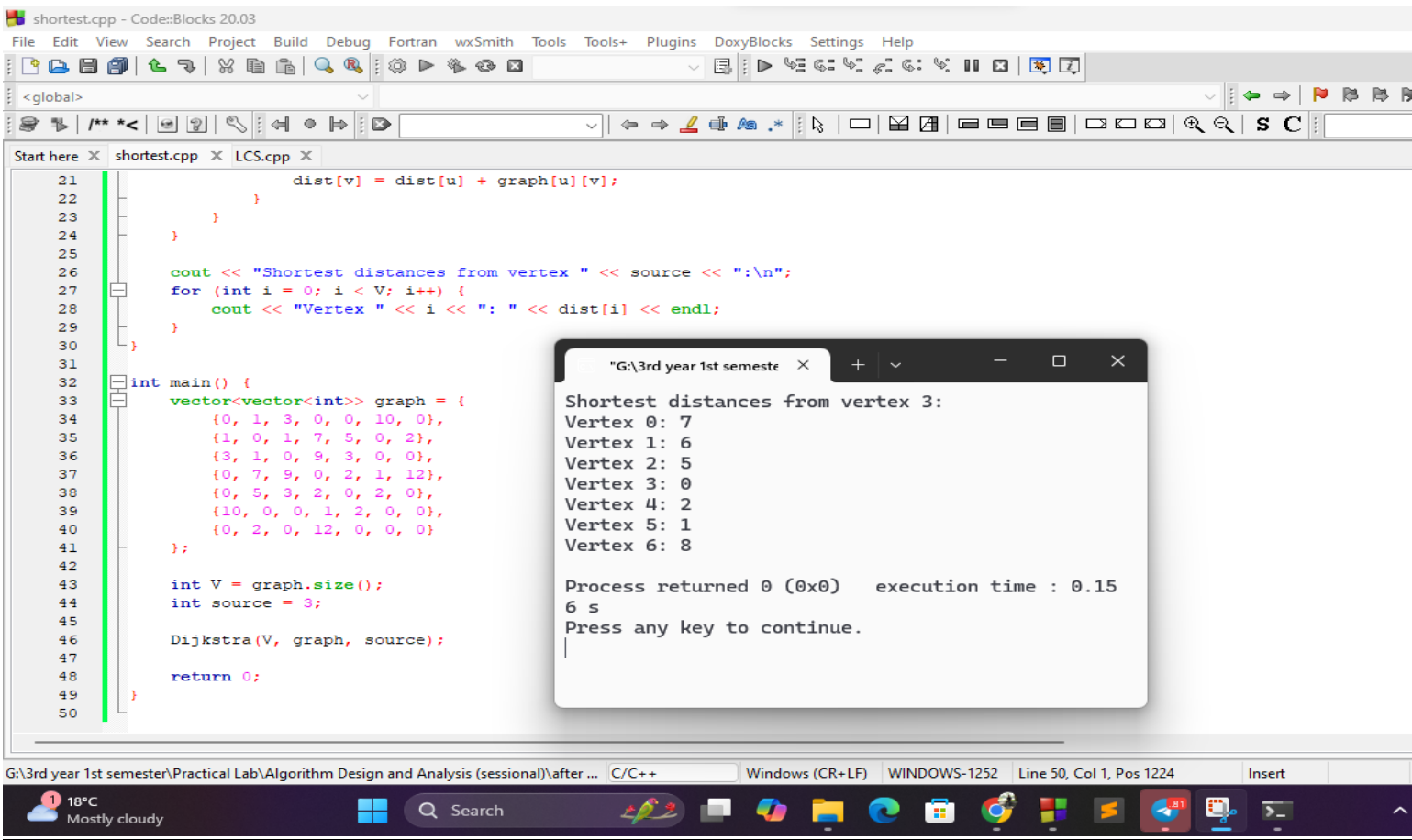
1.Write a program Shortest path finding using c++



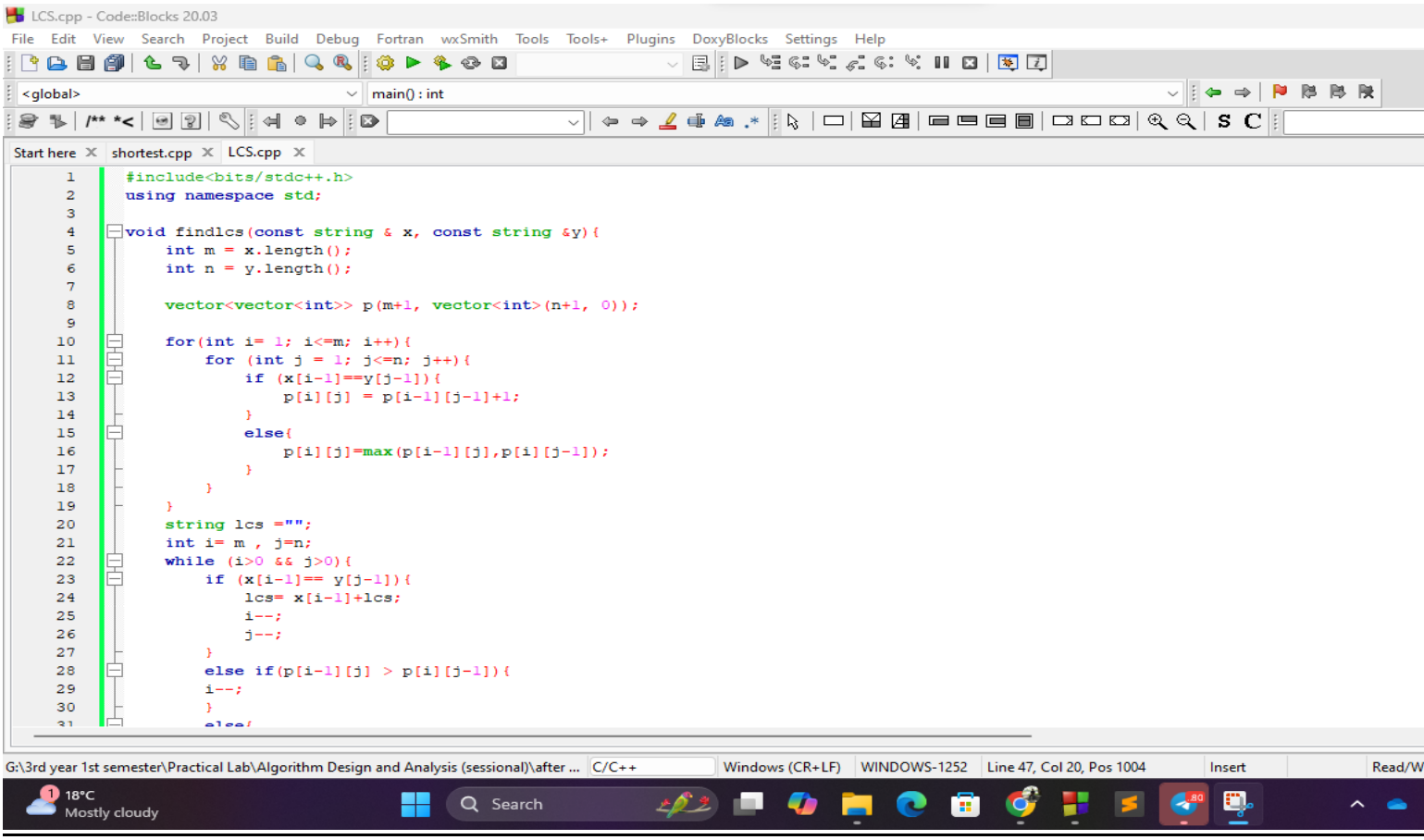
```cpp
#include <bits/stdc++.h>
using namespace std;

void Dijkstra(int V, vector<vector<int>>& graph, int source) {
    vector<int> dist(V, INT_MAX);
    vector<bool> visit(V, false);
    dist[source] = 0;

    for (int c = 0; c < V - 1; c++) {
        int u = -1;
        for (int i = 0; i < V; i++) {
            if (!visit[i] && (u == -1 || dist[i] < dist[u])) {
                u = i;
            }
        }

        visit[u] = true;

        for (int v = 0; v < V; v++) {
            if (graph[u][v] != 0 && !visit[v] && dist[u] + graph[u][v] < dist[v]) {
                dist[v] = dist[u] + graph[u][v];
            }
        }
    }

    cout << "Shortest distances from vertex " << source << ":\n";
    for (int i = 0; i < V; i++) {
        cout << "Vertex " << i << ": " << dist[i] << endl;
    }
}
```



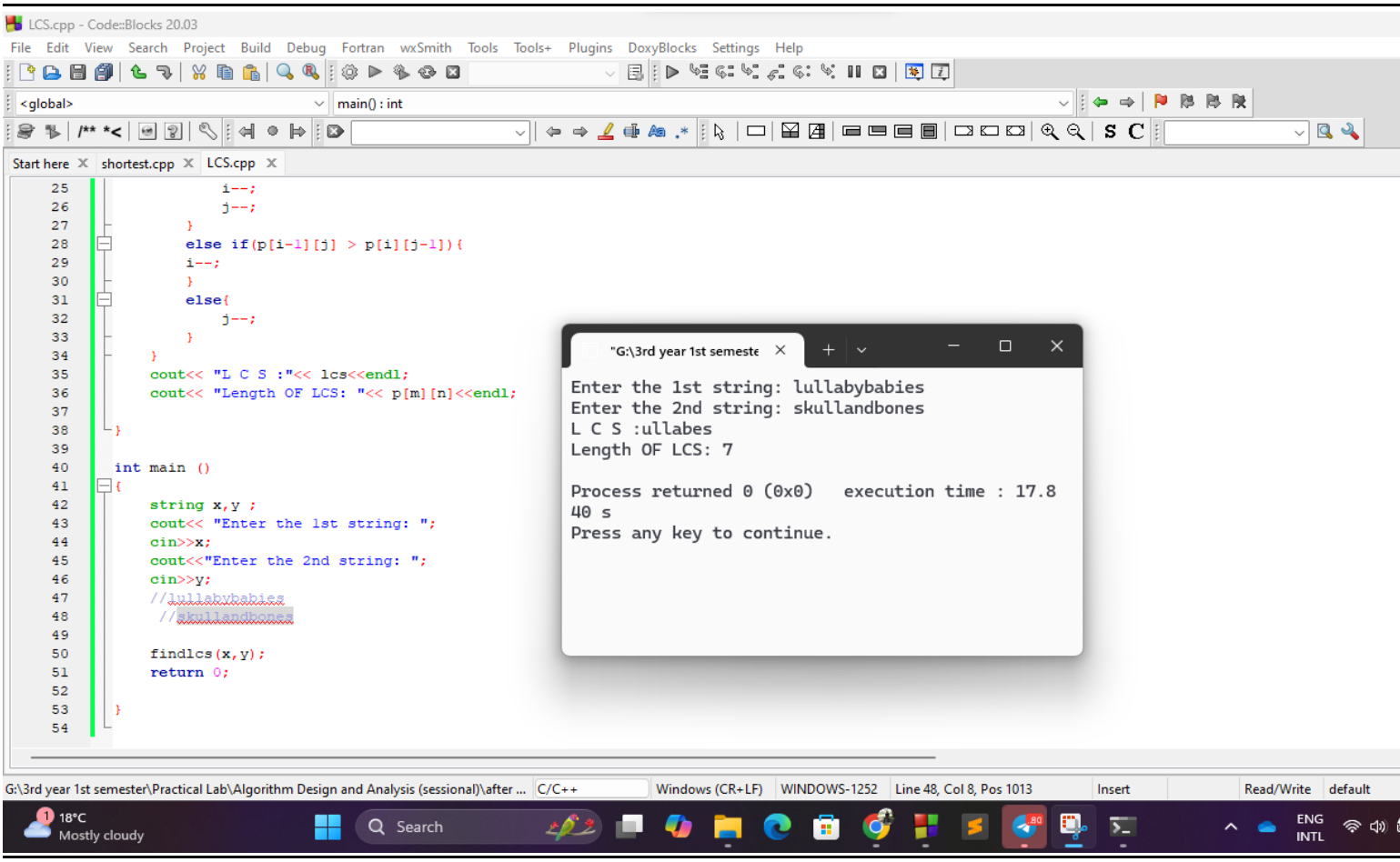
```cpp
int main() {
    vector<vector<int>> graph = {
        {0, 1, 3, 0, 0, 10, 0},
        {1, 0, 1, 7, 5, 0, 2},
        {3, 1, 0, 9, 3, 0, 0},
        {0, 7, 9, 0, 2, 1, 12},
        {0, 5, 3, 2, 0, 2, 0},
        {10, 0, 0, 1, 2, 0, 0},
        {0, 2, 0, 12, 0, 0, 0}
    };

    int V = graph.size();
    int source = 3;

    Dijkstra(V, graph, source);

    return 0;
}
```

```
"G:\3rd year 1st semeste
Shortest distances from vertex 3:
Vertex 0: 7
Vertex 1: 6
Vertex 2: 5
Vertex 3: 0
Vertex 4: 2
Vertex 5: 1
Vertex 6: 8

Process returned 0 (0x0)   execution time : 0.15
6 s
Press any key to continue.
```

2.Write a program Longest common subsequence(LCS) using c++.



```
#include<bits/stdc++.h>
using namespace std;

void findlcs(const string & x, const string &y){
    int m = x.length();
    int n = y.length();

    vector<vector<int>> p(m+1, vector<int>(n+1, 0));

    for(int i= 1; i<=m; i++){
        for (int j = 1; j<=n; j++){
            if (x[i-1]==y[j-1]){
                p[i][j] = p[i-1][j-1]+1;
            }
            else{
                p[i][j]=max(p[i-1][j],p[i][j-1]);
            }
        }
    }
    string lcs ="";
    int i= m , j=n;
    while (i>0 && j>0){
        if (x[i-1]== y[j-1]){
            lcs= x[i-1]+lcs;
            i--;
            j--;
        }
        else if(p[i-1][j] > p[i][j-1]){
        i--;
        }
        else{
```



```
            i--;
            j--;
        }
        else if(p[i-1][j] > p[i][j-1]){
        i--;
        }
        else{
            j--;
        }
    }
    cout<< "L C S :"<< lcs<<endl;
    cout<< "Length OF LCS: "<< p[m][n]<<endl;

}

int main ()
{
    string x,y ;
    cout<< "Enter the 1st string: ";
    cin>>x;
    cout<<"Enter the 2nd string: ";
    cin>>y;
    //lullabybabies
     //skullandbones

    findlcs(x,y);
    return 0;

}
```

```
"G:\3rd year 1st semeste
Enter the 1st string: lullabybabies
Enter the 2nd string: skullandbones
L C S :ullabes
Length OF LCS: 7

Process returned 0 (0x0)   execution time : 17.8
40 s
Press any key to continue.
```