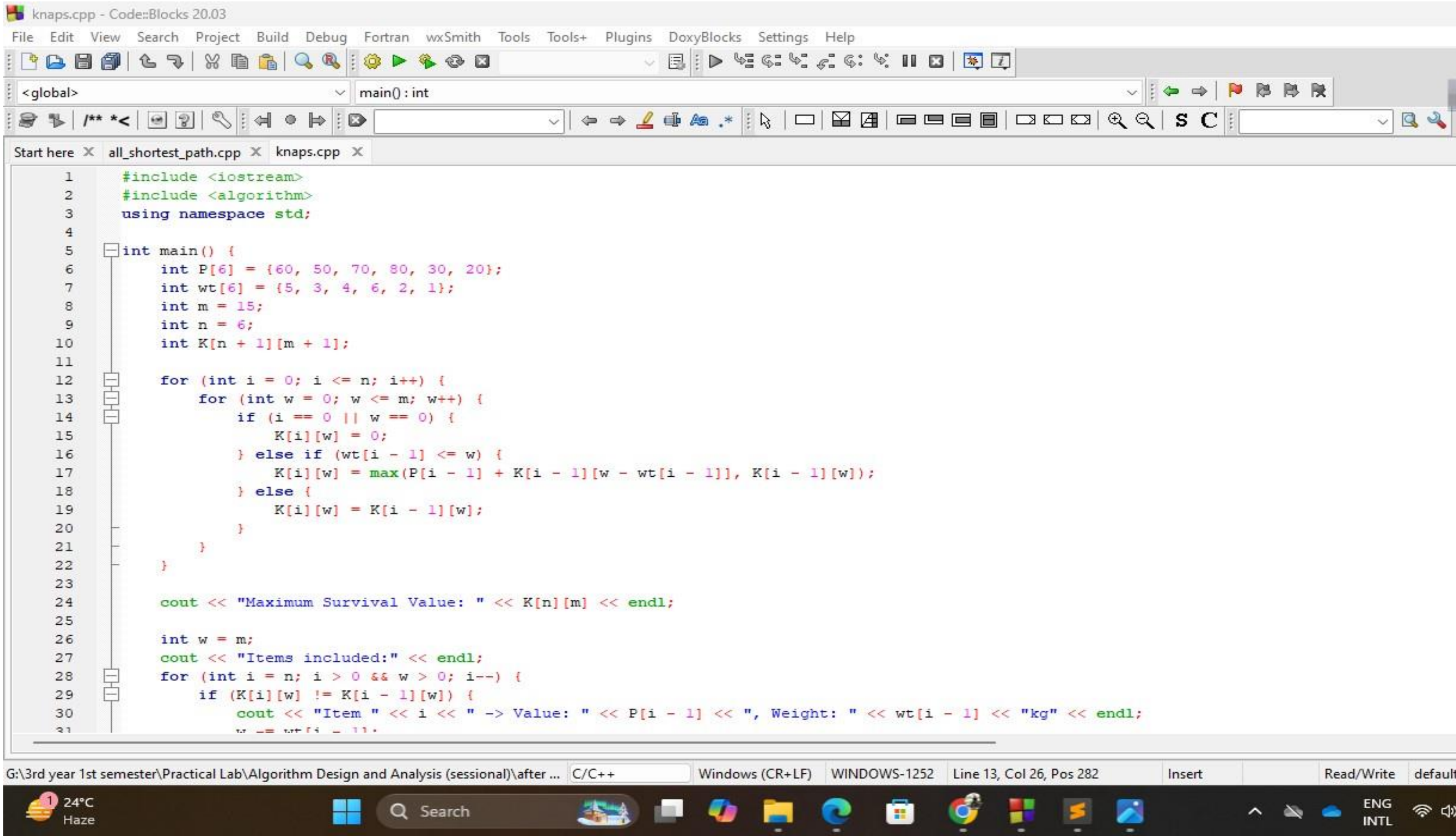
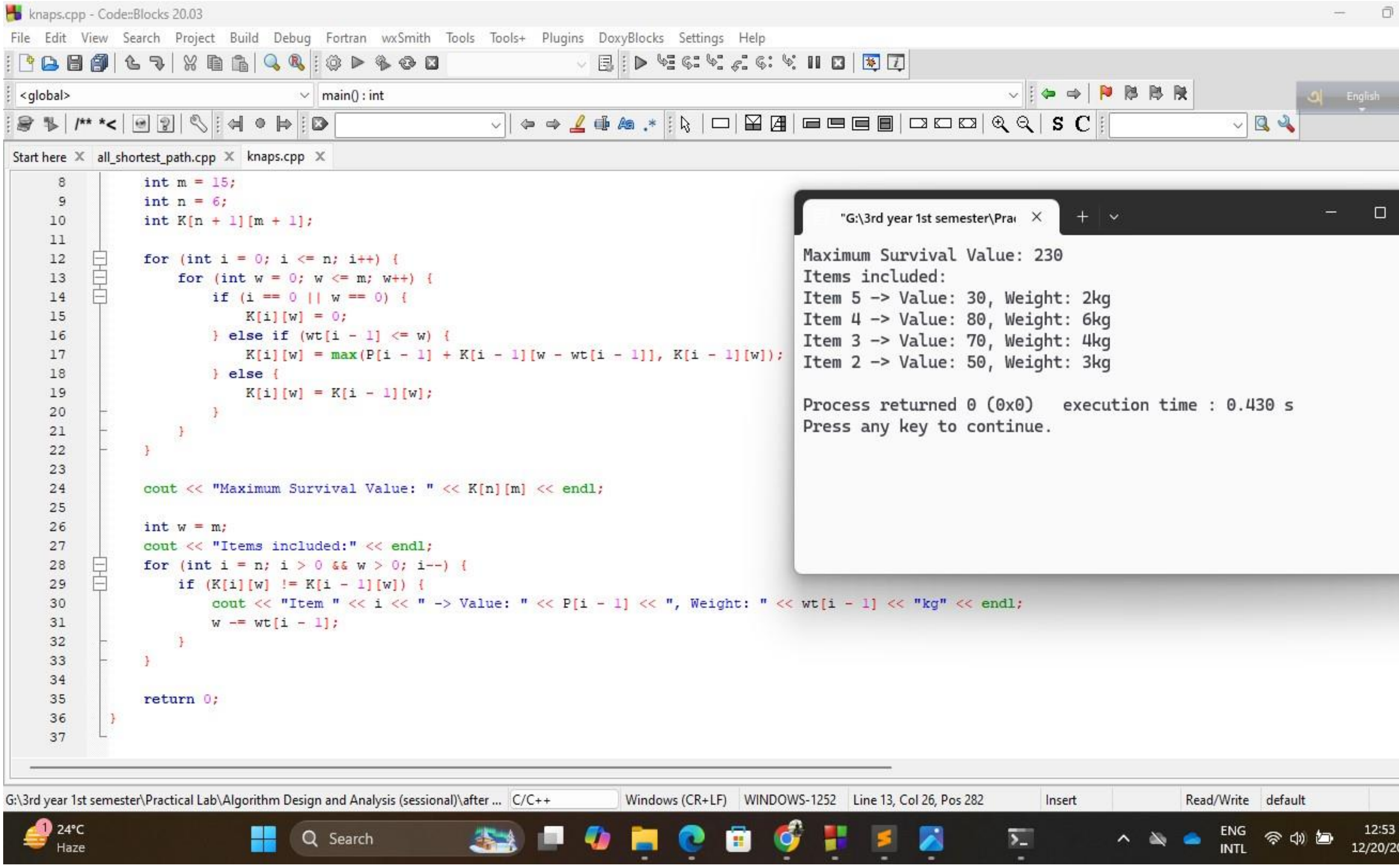
Task1.Implement the 0/1 Knapsack Problem: Packing a Survival Kit backpack.


knaps.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
Start here
all_shortest_path.cpp
knaps.cpp
G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ...
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 13, Col 26, Pos 282
Insert
Read/Write
default
24°C
Haze
Search
ENG
INTL

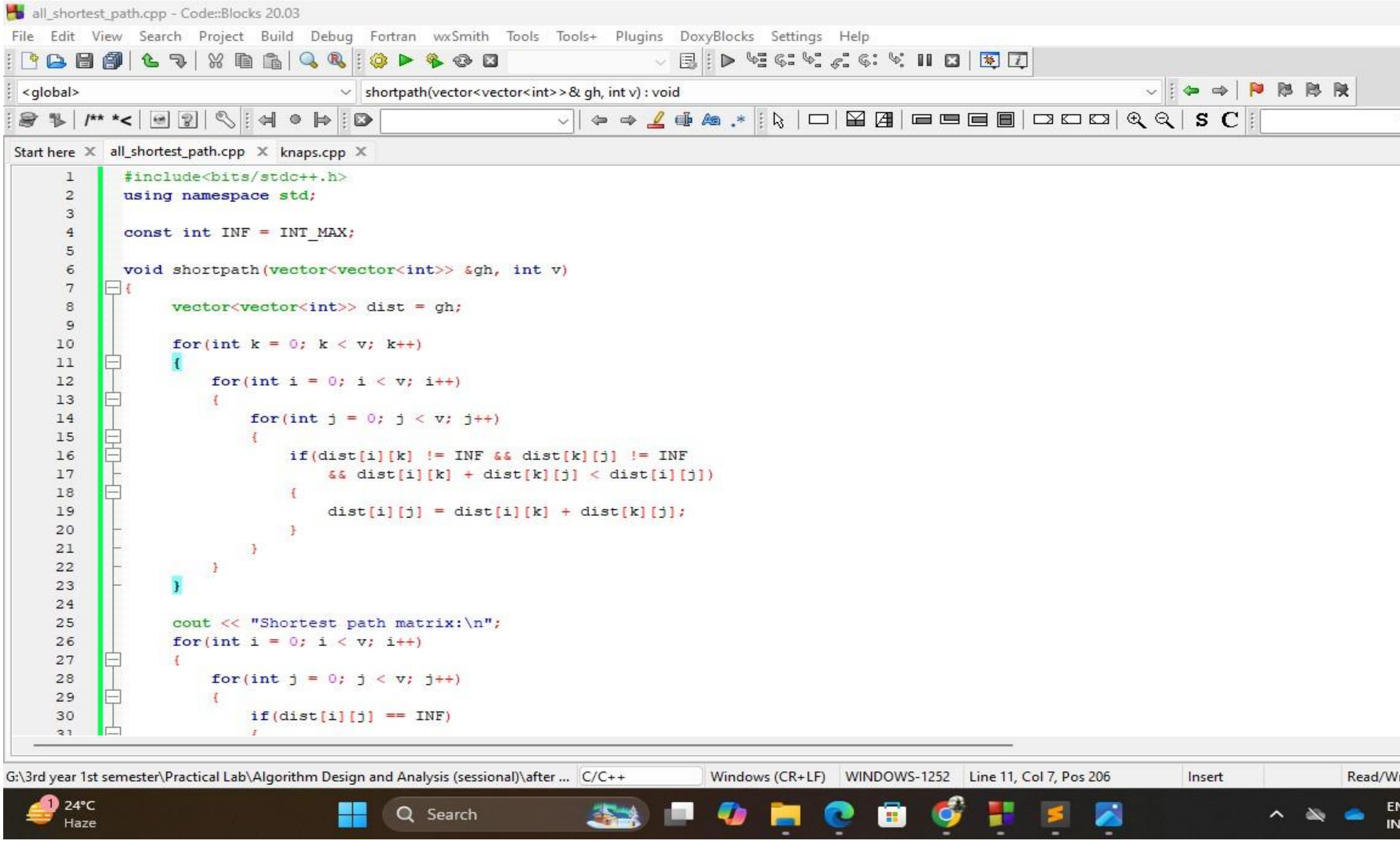
```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int P[6] = {60, 50, 70, 80, 30, 20};
    int wt[6] = {5, 3, 4, 6, 2, 1};
    int m = 15;
    int n = 6;
    int K[n + 1][m + 1];

    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= m; w++) {
            if (i == 0 || w == 0) {
                K[i][w] = 0;
            } else if (wt[i - 1] <= w) {
                K[i][w] = max(P[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
            } else {
                K[i][w] = K[i - 1][w];
            }
        }
    }

    cout << "Maximum Survival Value: " << K[n][m] << endl;

    int w = m;
    cout << "Items included:" << endl;
    for (int i = n; i > 0 && w > 0; i--) {
        if (K[i][w] != K[i - 1][w]) {
            cout << "Item " << i << " -> Value: " << P[i - 1] << ", Weight: " << wt[i - 1] << "kg" << endl;
```


knaps.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
English
Start here
all_shortest_path.cpp
knaps.cpp
"G:\3rd year 1st semester\Prac
G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ...
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 13, Col 26, Pos 282
Insert
Read/Write
default
24°C
Haze
Search
ENG
INTL
12:53
12/20/2

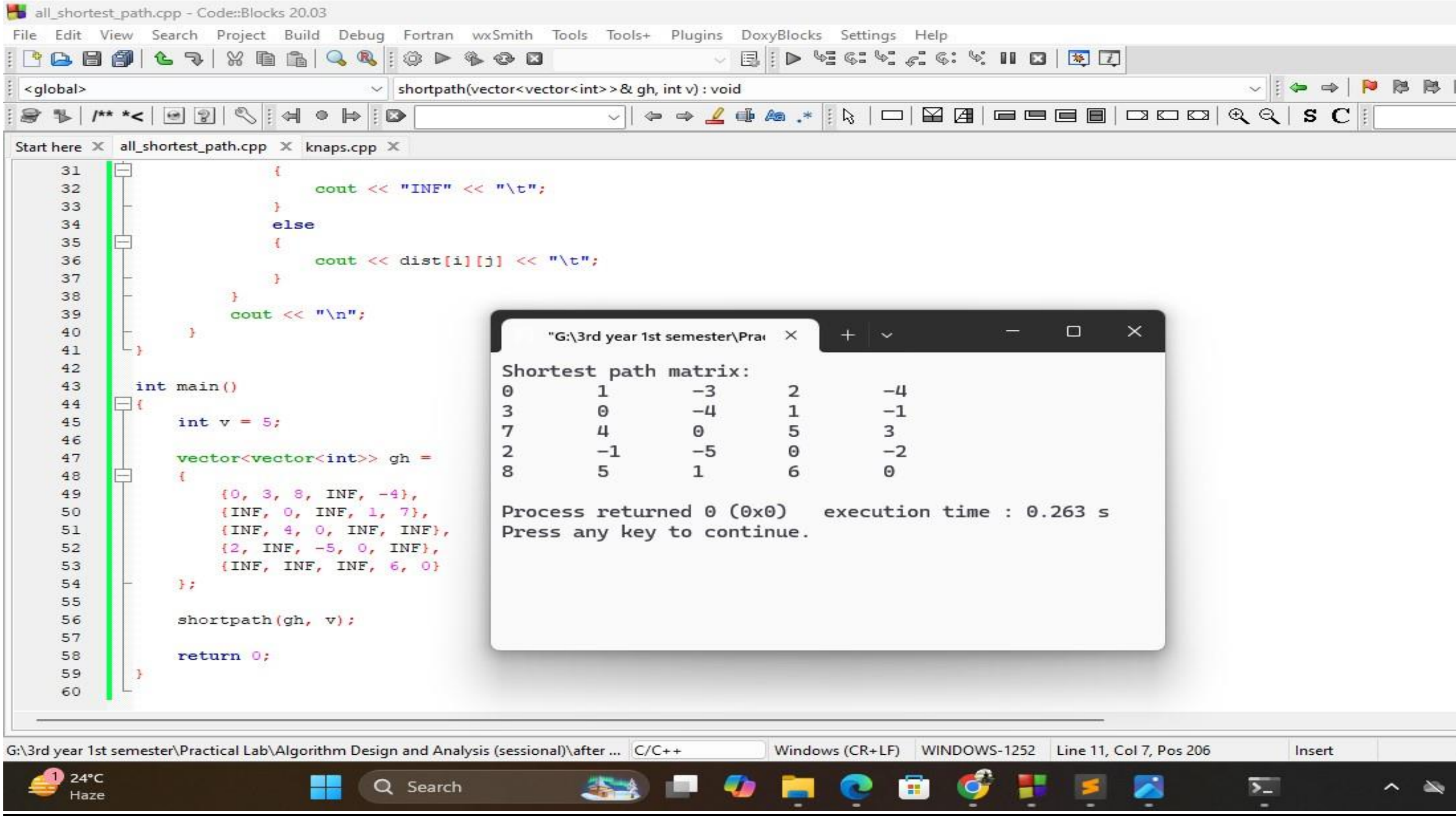
```cpp
    int m = 15;
    int n = 6;
    int K[n + 1][m + 1];

    for (int i = 0; i <= n; i++) {
        for (int w = 0; w <= m; w++) {
            if (i == 0 || w == 0) {
                K[i][w] = 0;
            } else if (wt[i - 1] <= w) {
                K[i][w] = max(P[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
            } else {
                K[i][w] = K[i - 1][w];
            }
        }
    }

    cout << "Maximum Survival Value: " << K[n][m] << endl;

    int w = m;
    cout << "Items included:" << endl;
    for (int i = n; i > 0 && w > 0; i--) {
        if (K[i][w] != K[i - 1][w]) {
            cout << "Item " << i << " -> Value: " << P[i - 1] << ", Weight: " << wt[i - 1] << "kg" << endl;
            w -= wt[i - 1];
        }
    }

    return 0;
}
```

```
Maximum Survival Value: 230
Items included:
Item 5 -> Value: 30, Weight: 2kg
Item 4 -> Value: 80, Weight: 6kg
Item 3 -> Value: 70, Weight: 4kg
Item 2 -> Value: 50, Weight: 3kg

Process returned 0 (0x0)   execution time : 0.430 s
Press any key to continue.
```

Task:2.Implement the Floyd-Warshall algorithm to find the shortest paths between all vertices in the directed weighted graph


all_shortest_path.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
shortpath(vector<vector<int>>& gh, int v) : void
Start here
all_shortest_path.cpp
knaps.cpp
G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ...
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 11, Col 7, Pos 206
Insert
Read/W
24°C
Haze
Search

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

const int INF = INT_MAX;

void shortpath(vector<vector<int>> &gh, int v)
{
    vector<vector<int>> dist = gh;

    for(int k = 0; k < v; k++)
    {
        for(int i = 0; i < v; i++)
        {
            for(int j = 0; j < v; j++)
            {
                if(dist[i][k] != INF && dist[k][j] != INF
                    && dist[i][k] + dist[k][j] < dist[i][j])
                {
                    dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
    }

    cout << "Shortest path matrix:\n";
    for(int i = 0; i < v; i++)
    {
        for(int j = 0; j < v; j++)
        {
            if(dist[i][j] == INF)
```


all_shortest_path.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
shortpath(vector<vector<int>>& gh, int v) : void
Start here
all_shortest_path.cpp
knaps.cpp
"G:\3rd year 1st semester\Pra
G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ...
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 11, Col 7, Pos 206
Insert
24°C
Haze
Search

```cpp
            {
                cout << "INF" << "\t";
            }
            else
            {
                cout << dist[i][j] << "\t";
            }
        }
        cout << "\n";
    }
}

int main()
{
    int v = 5;

    vector<vector<int>> gh =
    {
        {0, 3, 8, INF, -4},
        {INF, 0, INF, 1, 7},
        {INF, 4, 0, INF, INF},
        {2, INF, -5, 0, INF},
        {INF, INF, INF, 6, 0}
    };

    shortpath(gh, v);

    return 0;
}
```

```
Shortest path matrix:
0       1       -3      2       -4
3       0       -4      1       -1
7       4       0       5       3
2       -1      -5      0       -2
8       5       1       6       0

Process returned 0 (0x0)   execution time : 0.263 s
Press any key to continue.
```