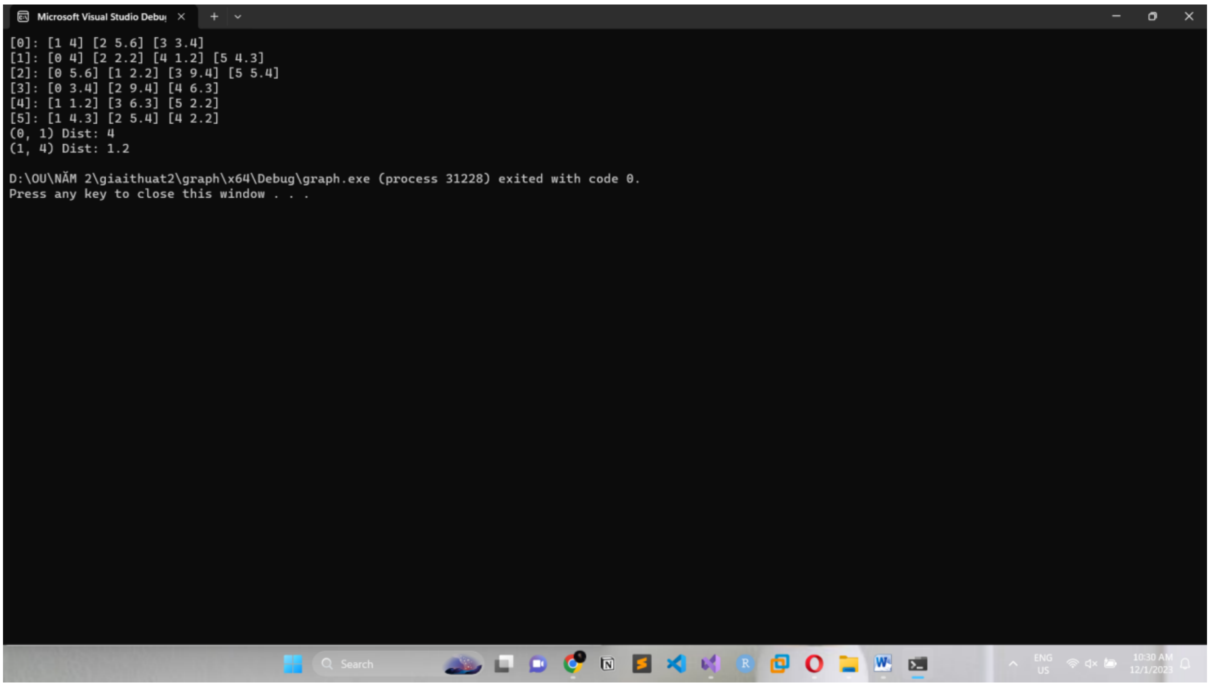
```
vector <float> dijkstra(lgraph l, int u, vector <int>& prev) {
        priority_queue <vertex, vector <vertex>, cmp > q;

        vector <bool> visited(l.size(), false);
        q.push(vertex{ u,0. });

        visited[0] = true;
        vector <float> dist(l.size(), INT_MAX);
        // distance from the source to itself is zero
        dist[u] = 0;

        while (!q.empty()) {
                vertex b = q.top(); q.pop();
                // loop all neighbor of b
                float w = b.w;
                for (auto e : l[b.u]) {
                        if (dist[e.v] > w + e.w) {
                                dist[e.v] = w + e.w;
                                q.push(vertex{ e.v, dist[e.v] });

                                prev[e.v] = b.u;
                        }
                }
        }
        return dist;
}

void printPath(vector <int> prev, vector <float> dist, int source, int
destination) {
        if (destination < 0) {
                return;
        }
        printPath(prev, dist, source, prev[destination]);
        if (destination != -1 && prev[destination] != -1) {
                cout << "(" << prev[destination] << ", " << destination << ") ";
                cout << "Dist: " << dist[destination] - dist[prev[destination]] <<
endl;
        }
}

int main() {
        lgraph l;
        l = createLgraph();

        printLgraph(l);
        vector <int> prev(l.size());
        for (int i = 0; i < prev.size(); i++) {
                prev[i] = -1;
        }
        vector <float> d = dijkstra(l, 0, prev);
        printPath(prev, d, 0, 4);
        return 0;
}
```



```
[0]: [1 4] [2 5.6] [3 3.4]
[1]: [0 4] [2 2.2] [4 1.2] [5 4.3]
[2]: [0 5.6] [1 2.2] [3 9.4] [5 5.4]
[3]: [0 3.4] [2 9.4] [4 6.3]
[4]: [1 1.2] [3 6.3] [5 2.2]
[5]: [1 4.3] [2 5.4] [4 2.2]
(0, 1) Dist: 4
(1, 4) Dist: 1.2

D:\OU\NĂM 2\giaithuat2\graph\x64\Debug\graph.exe (process 31228) exited with code 0.
Press any key to close this window . . .
```