vector <int>bfs(lgraph l, int start) {

vector<bool> visited(l.size(), false);

queue<int> q;

vector<int> tranversal;

visited[start] = true;

q.push(start);

while (!q.empty()) {

int current = q.front();

q.pop();

tranversal.push\_back(current);

for (int i = 0; i < l[current].size(); i++) {

int w = l[current][i].v;

if (!visited[w]) {

visited[w] = true;

q.push(w);

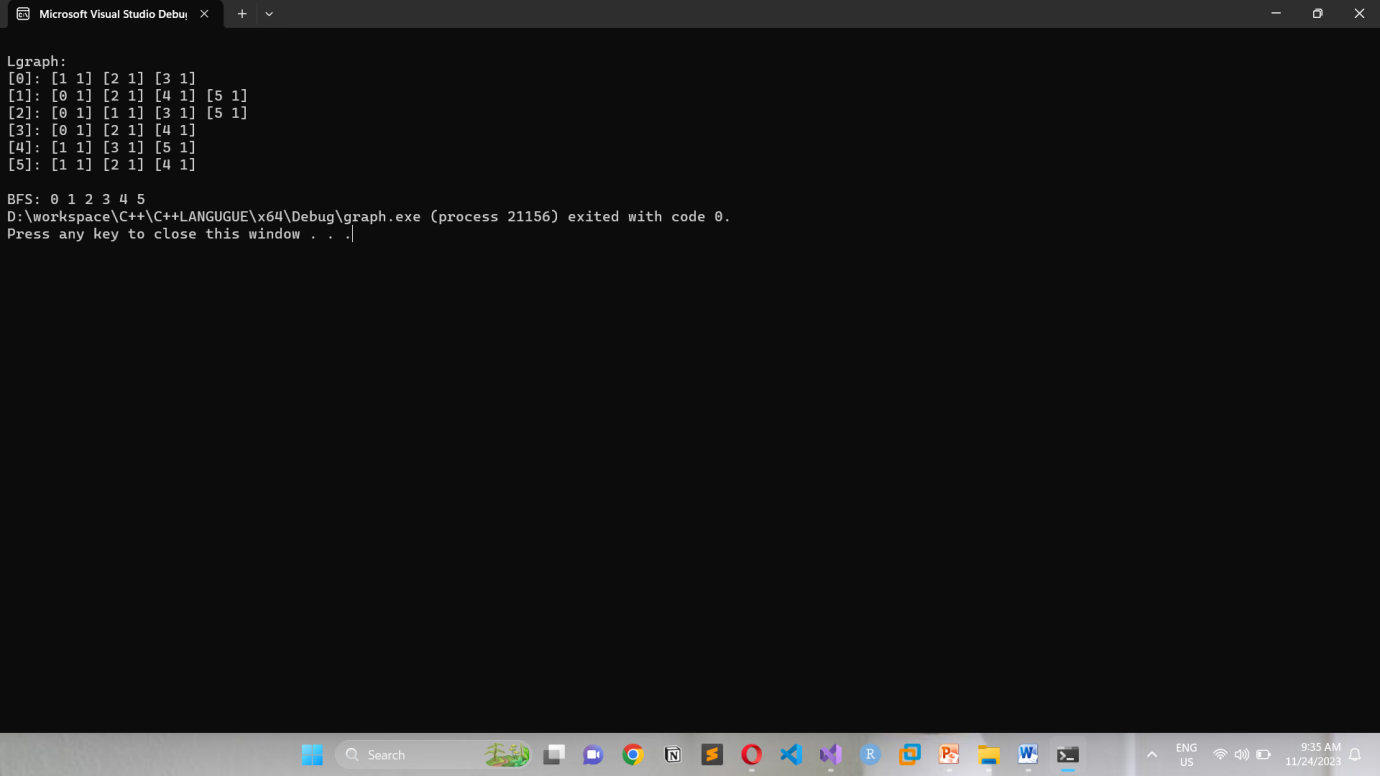
}

}

}

return tranversal;

}



vector <int>dfs(lgraph l, int start) {

vector<bool> visited(l.size(), false);

stack <int> st;

vector<int> tranversal;

visited[start] = true;

st.push(start);

while (!st.empty()) {

int current = st.top();

st.pop();

tranversal.push\_back(current);

for (int i = 0; i < l[current].size(); i++) {

int w = l[current][i].v;

if (!visited[w]) {

visited[w] = true;

st.push(w);

}

}

}

return tranversal;

}

