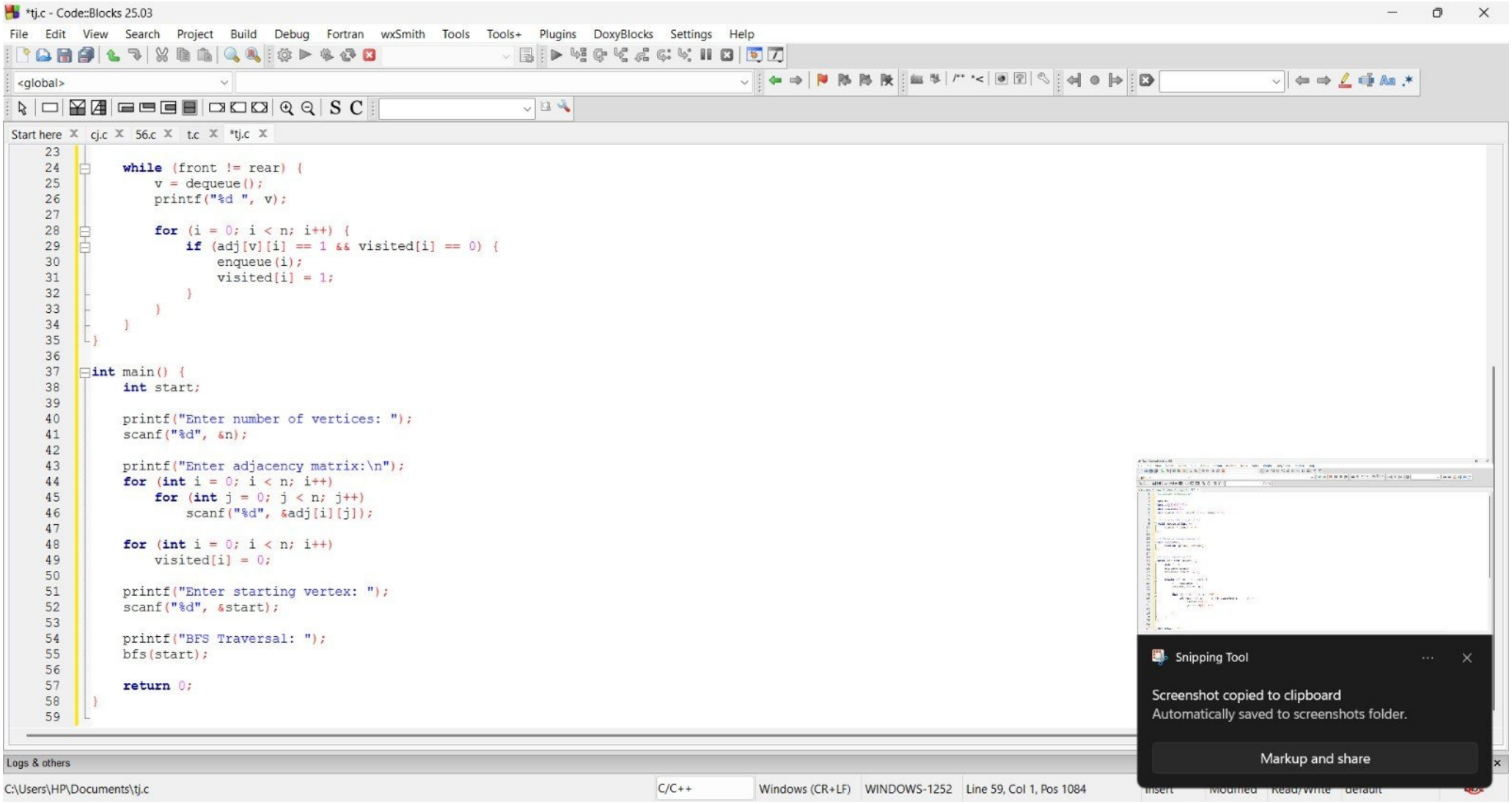
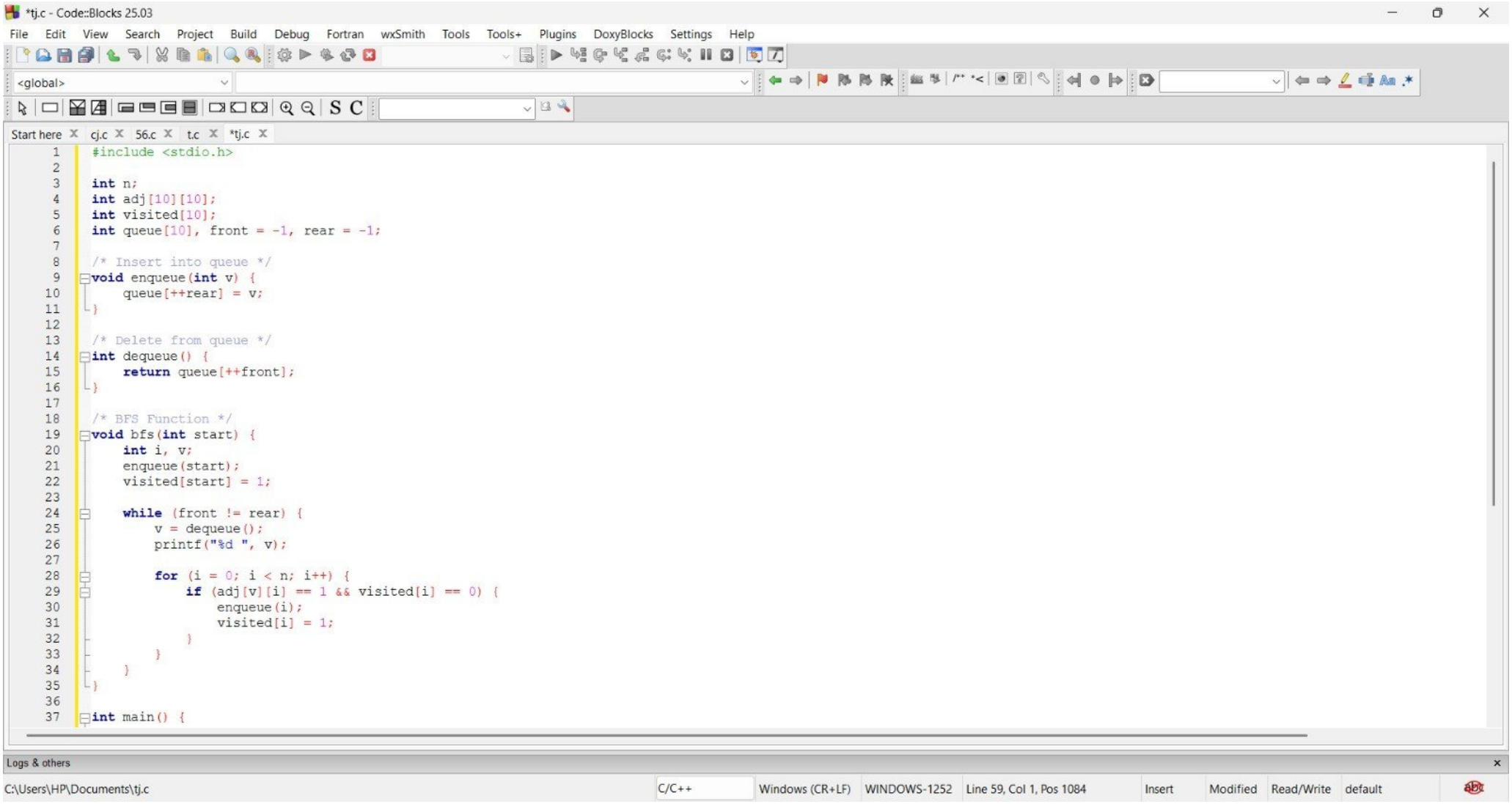
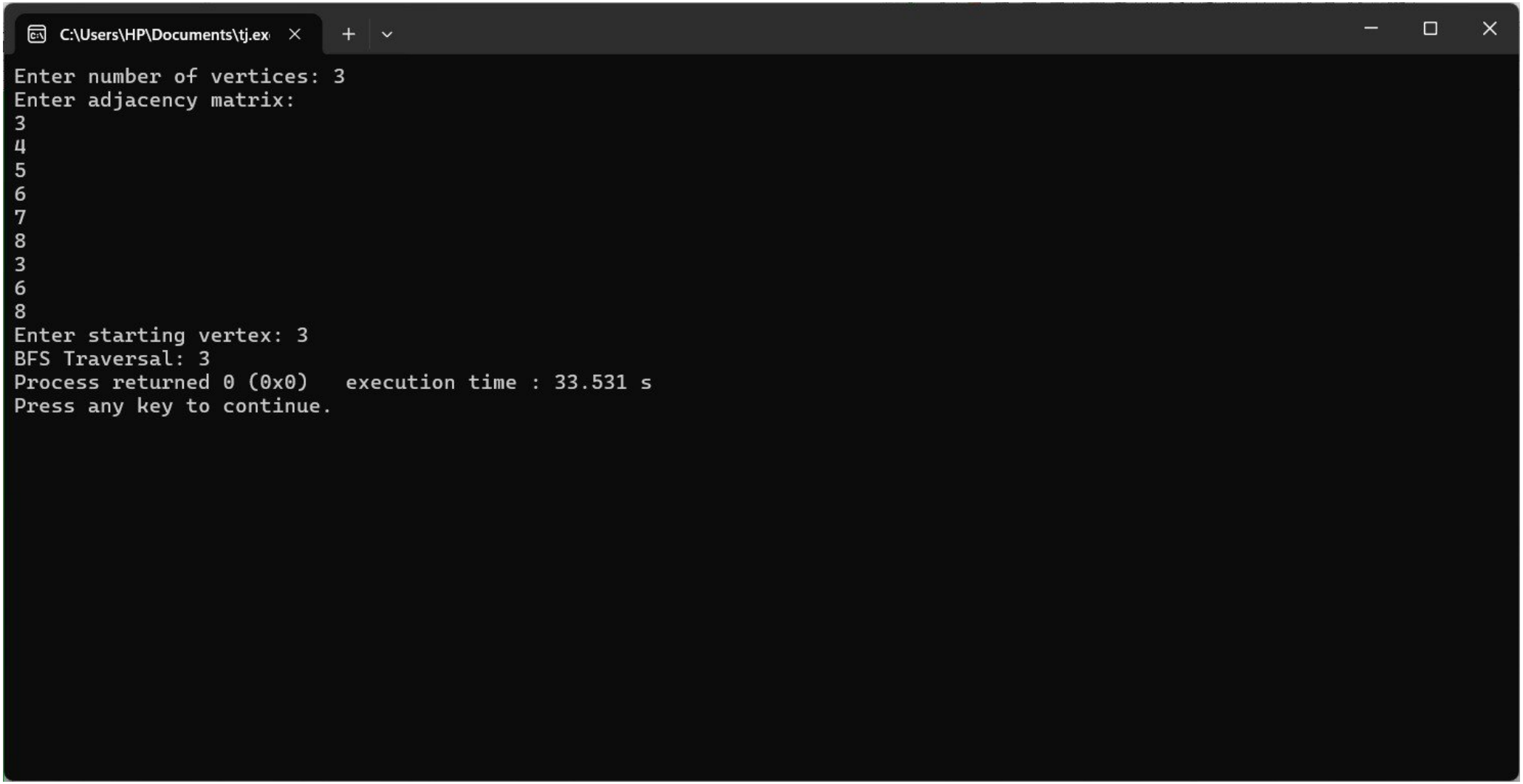
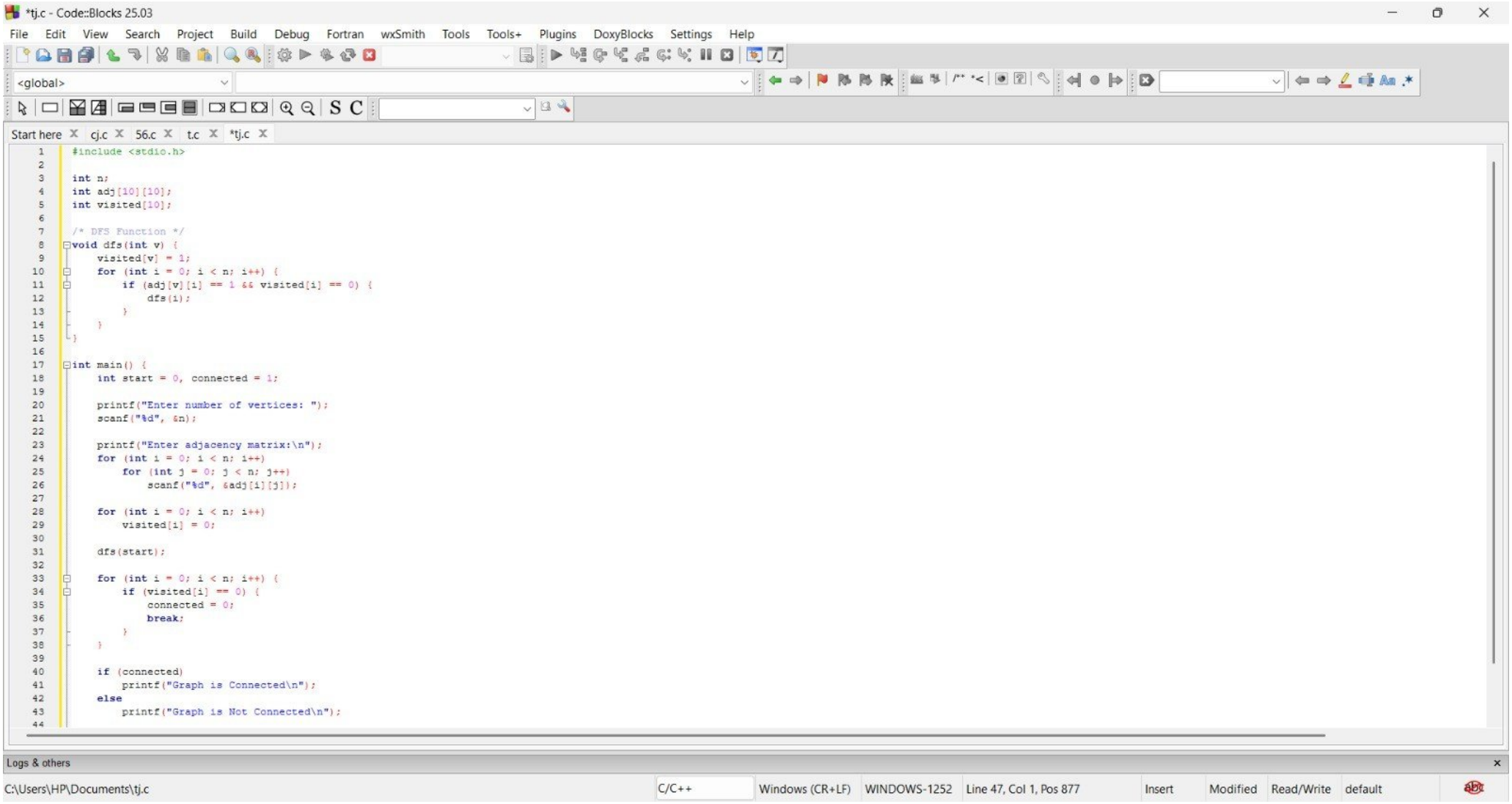
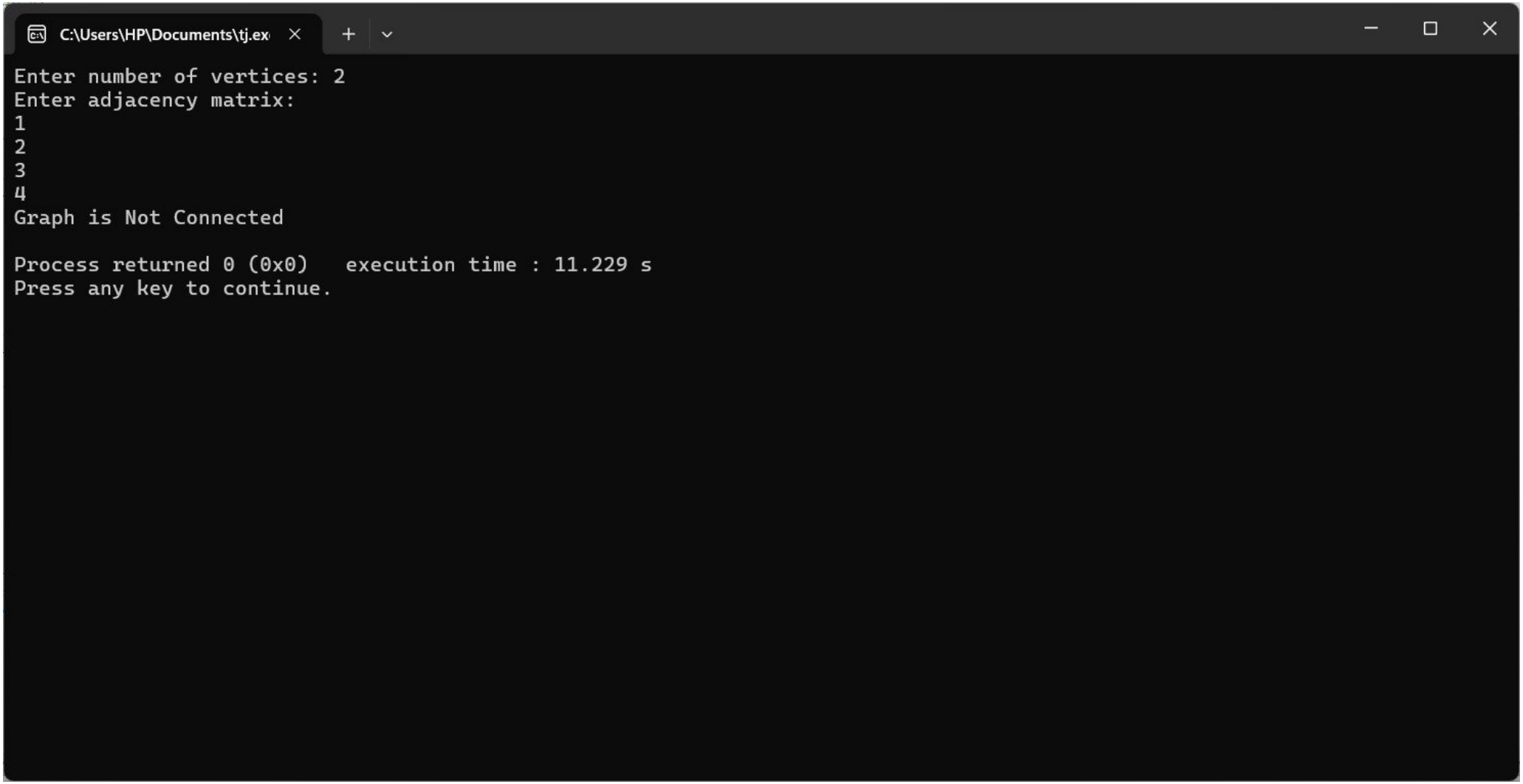
9	4	5	a) Write a program to traverse a graph using BFS method.
		5	b) Write a program to check whether given graph is connected or not using DFS method.


*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here cj.c 56.c t.c *tj.c
while (front != rear) {
v = dequeue();
printf("%d ", v);
for (i = 0; i < n; i++) {
if (adj[v][i] == 1 && visited[i] == 0) {
enqueue(i);
visited[i] = 1;
}
}
}
}
int main() {
int start;
printf("Enter number of vertices: ");
scanf("%d", &n);
printf("Enter adjacency matrix:\n");
for (int i = 0; i < n; i++)
for (int j = 0; j < n; j++)
scanf("%d", &adj[i][j]);
for (int i = 0; i < n; i++)
visited[i] = 0;
printf("Enter starting vertex: ");
scanf("%d", &start);
printf("BFS Traversal: ");
bfs(start);
return 0;
}
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 59, Col 1, Pos 1084
Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Markup and share


*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
cj.c
56.c
t.c
*tj.c
#include <stdio.h>
int n;
int adj[10][10];
int visited[10];
int queue[10], front = -1, rear = -1;
/* Insert into queue */
void enqueue(int v) {
queue[++rear] = v;
}
/* Delete from queue */
int dequeue() {
return queue[++front];
}
/* BFS Function */
void bfs(int start) {
int i, v;
enqueue(start);
visited[start] = 1;
while (front != rear) {
v = dequeue();
printf("%d ", v);
for (i = 0; i < n; i++) {
if (adj[v][i] == 1 && visited[i] == 0) {
enqueue(i);
visited[i] = 1;
}
}
}
}
int main() {
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 59, Col 1, Pos 1084
Insert
Modified
Read/Write
default


C:\Users\HP\Documents\tj.ex
Enter number of vertices: 3
Enter adjacency matrix:
3
4
5
6
7
8
3
6
8
Enter starting vertex: 3
BFS Traversal: 3
Process returned 0 (0x0)   execution time : 33.531 s
Press any key to continue.


*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
cj.c
56.c
t.c
*tj.c
#include <stdio.h>
int n;
int adj[10][10];
int visited[10];
/* DFS Function */
void dfs(int v) {
visited[v] = 1;
for (int i = 0; i < n; i++) {
if (adj[v][i] == 1 && visited[i] == 0) {
dfs(i);
}
}
}
int main() {
int start = 0, connected = 1;
printf("Enter number of vertices: ");
scanf("%d", &n);
printf("Enter adjacency matrix:\n");
for (int i = 0; i < n; i++)
for (int j = 0; j < n; j++)
scanf("%d", &adj[i][j]);
for (int i = 0; i < n; i++)
visited[i] = 0;
dfs(start);
for (int i = 0; i < n; i++) {
if (visited[i] == 0) {
connected = 0;
break;
}
}
if (connected)
printf("Graph is Connected\n");
else
printf("Graph is Not Connected\n");
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 47, Col 1, Pos 877
Insert
Modified
Read/Write
default


C:\Users\HP\Documents\tj.ex
Enter number of vertices: 2
Enter adjacency matrix:
1
2
3
4
Graph is Not Connected
Process returned 0 (0x0)   execution time : 11.229 s
Press any key to continue.