1. **BFS**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_VERTICES 100

void bfs(int graph[MAX\_VERTICES][MAX\_VERTICES], int vertices, int start) {

int queue[MAX\_VERTICES];

int visited[MAX\_VERTICES] = {0};

int front = -1, rear = -1;

queue[++rear] = start;

visited[start] = 1;

while (front != rear) {

int currentVertex = queue[++front];

printf("%d ", currentVertex);

for (int i = 0; i < vertices; ++i) {

if (graph[currentVertex][i] == 1 && !visited[i]) {

queue[++rear] = i;

visited[i] = 1;

}

}

}

}

int main() {

int vertices = 5, edges = 4;

int graph[MAX\_VERTICES][MAX\_VERTICES] = {{0, 1, 1, 0, 0},{1, 0, 1, 1, 0},{1, 1, 0, 0, 1},{0, 1, 0, 0, 1},{0, 0, 1, 1, 0}};

int startVertex = 0;

printf("BFS traversal starting from vertex %d: ", startVertex);

bfs(graph, vertices, startVertex);

return 0;

}

1. **DFS**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_VERTICES 100

void dfs(int graph[MAX\_VERTICES][MAX\_VERTICES], int vertices, int currentVertex, int visited[]) {

printf("%d ", currentVertex);

visited[currentVertex] = 1;

for (int i = 0; i < vertices; ++i)

if (graph[currentVertex][i] == 1 && !visited[i])

dfs(graph, vertices, i, visited);

}

int main() {

int vertices = 5, edges = 4;

int graph[MAX\_VERTICES][MAX\_VERTICES] = {{0, 1, 1, 0, 0},{1, 0, 1, 1, 0},{1, 1, 0, 0, 1},{0, 1, 0, 0, 1},{0, 0, 1, 1, 0}};

int startVertex = 0;

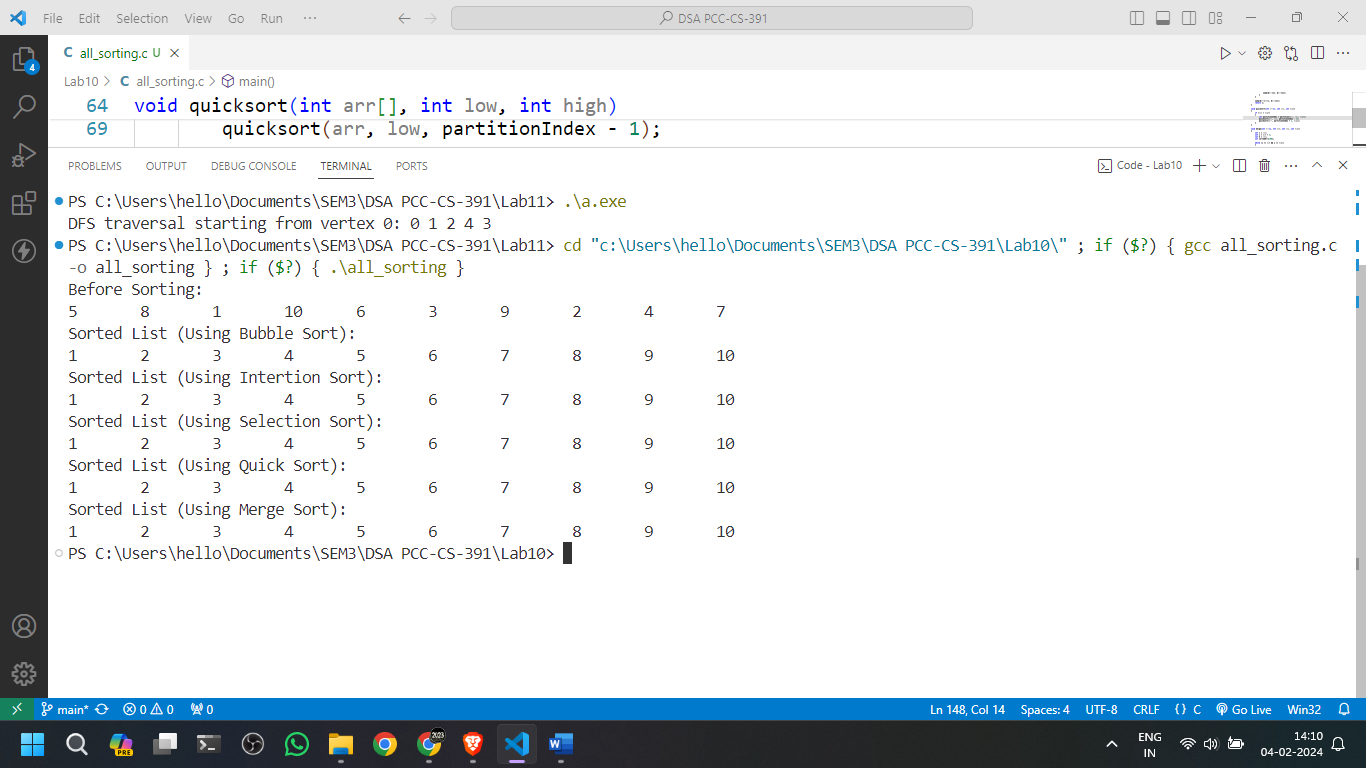
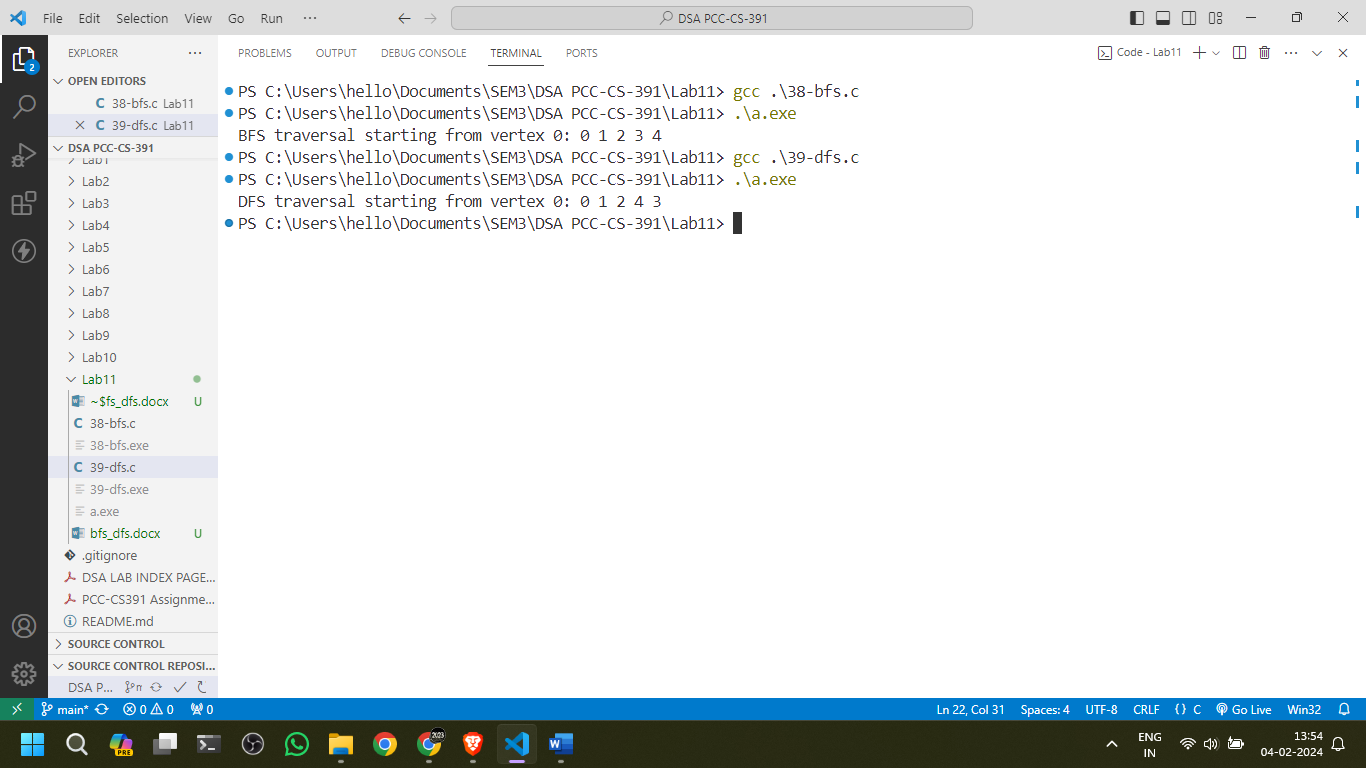
int visited[MAX\_VERTICES] = {0};

printf("DFS traversal starting from vertex %d: ", startVertex);

dfs(graph, vertices, startVertex, visited);

return 0;

}

**Output -**