9. Topological Sort

```c
#include <stdio.h>

int nodes, adjMatrix[10][10], result[10], visited[10], stackTop = 0;

void depthFirstSearch(int currentNode, int nodes, int adjMatrix[][10]) {
    visited[currentNode] = 1;
    for (int i = 0; i < nodes; i++) {
        if (adjMatrix[currentNode][i] == 1 && visited[i] == 0) {
            depthFirstSearch(i, nodes, adjMatrix);
        }
    }
    result[stackTop++] = currentNode;
}

void topologicalSort(int nodes, int adjMatrix[][10]) {
    for (int i = 0; i < nodes; i++) {
        visited[i] = 0;
    }
    for (int i = 0; i < nodes; i++) {
        if (visited[i] == 0) {
            depthFirstSearch(i, nodes, adjMatrix);
        }
    }
}

int main() {
    printf("Enter the number of nodes: ");
    scanf("%d", &nodes);
    printf("Enter the adjacency matrix:\n");
    for (int i = 0; i < nodes; i++) {
        for (int j = 0; j < nodes; j++) {
            scanf("%d", &adjMatrix[i][j]);
        }
    }

    topologicalSort(nodes, adjMatrix);

    printf("Topological Sort (DFS): ");
    for (int i = nodes - 1; i >= 0; i--) {
        printf("%d ", result[i]);
    }
    printf("\n");

    return 0;
}
```

```
Enter the number of nodes: 6
Enter the adjacency matrix:
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
0 0 0 0 0 0
Topological Sort (DFS): 5 4 3 2 1 0
```