#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* createNode(int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = NULL;

return newNode;

}

void addEdge(struct Node\*\* graph, int src, int dest) {

struct Node\* newNode = createNode(dest);

newNode->next = graph[src];

graph[src] = newNode;

newNode = createNode(src);

newNode->next = graph[dest];

graph[dest] = newNode;

}

void DFS(struct Node\*\* graph, int vertex, int\* visited) {

visited[vertex] = 1;

printf("%d ", vertex);

struct Node\* temp = graph[vertex];

while (temp != NULL) {

int adjVertex = temp->data;

if (visited[adjVertex] == 0) {

DFS(graph, adjVertex, visited);

}

temp = temp->next;

}

}

void DFSTraversal(struct Node\*\* graph, int numVertices) {

int\* visited = (int\*)calloc(numVertices, sizeof(int));

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

if (visited[i] == 0) {

DFS(graph, i, visited);

}

}

free(visited);

}

int main() {

int numVertices = 5;

struct Node\*\* graph = (struct Node\*\*)malloc(numVertices \* sizeof(struct Node\*));

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

graph[i] = NULL;

}

addEdge(graph, 0, 1);

addEdge(graph, 0, 2);

addEdge(graph, 1, 3);

addEdge(graph, 1, 4);

DFSTraversal(graph, numVertices);

return 0;

}

A screenshot of a computer

Description automatically generated