Institute of Computer Technology

B. Tech. Computer Science and Engineering

Sub: DS

Course Code: 2CSE302

Practical – 16

Name: Jaymin Gondaliya

Enrollment No: 23162171007

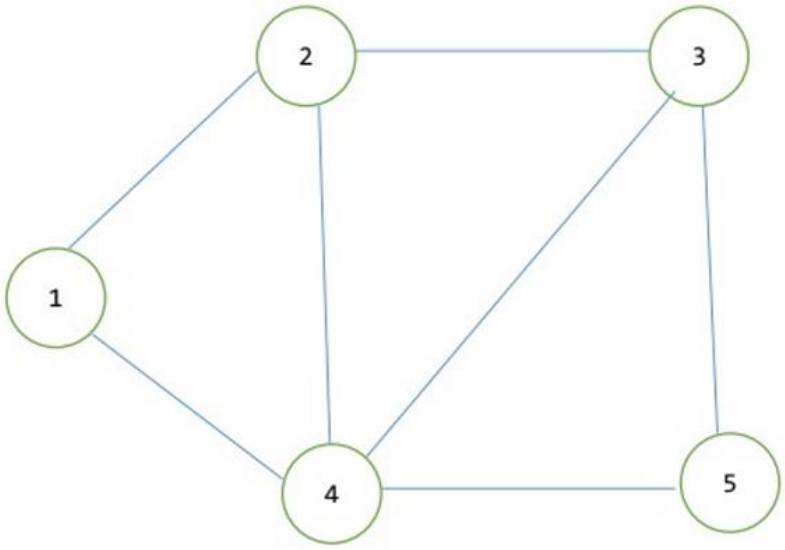
Sem - 3

Branch: CS

Class: A

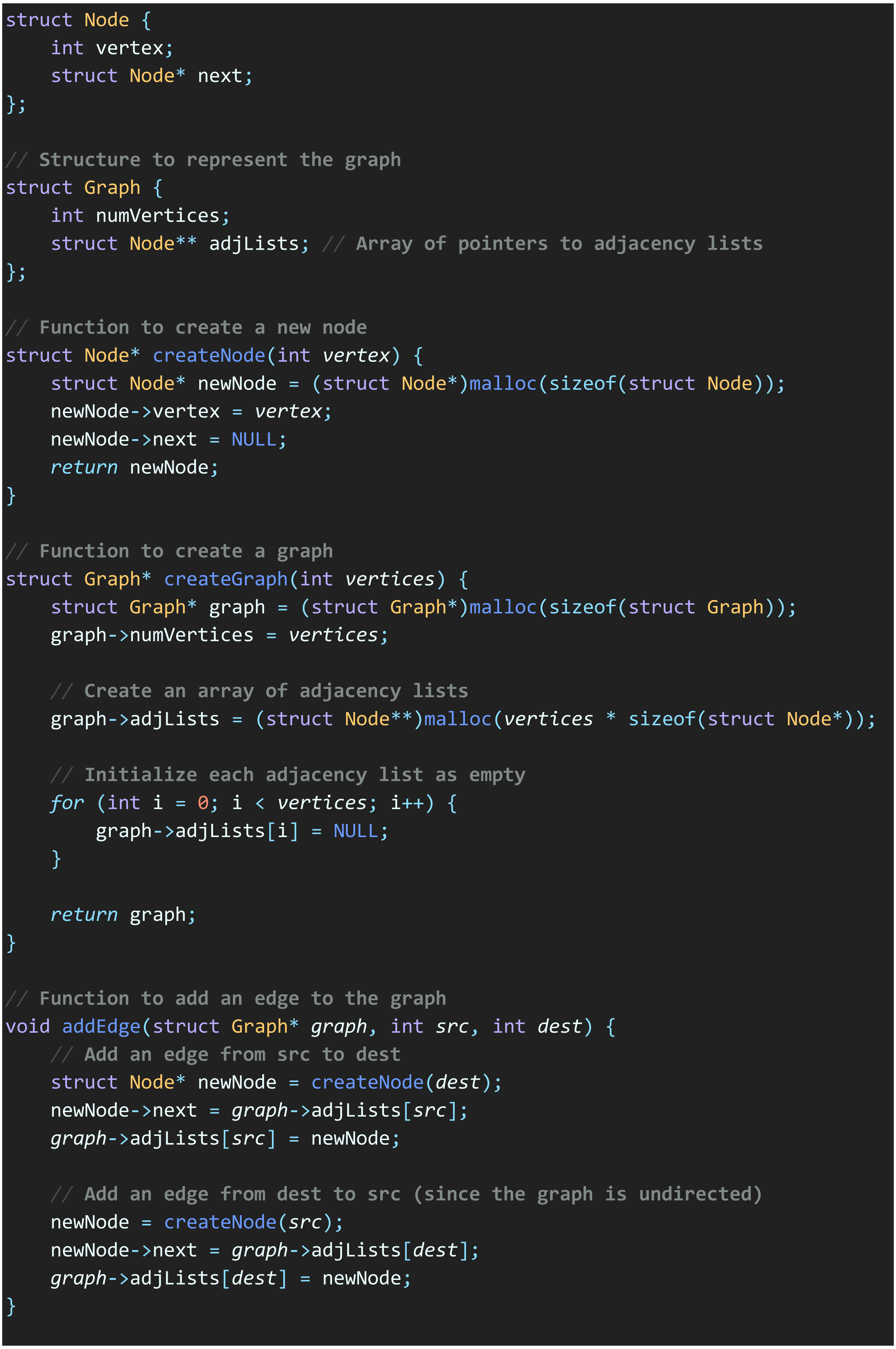
Batch: 32

Implement below mentioned graph data structure using C/C++ language.

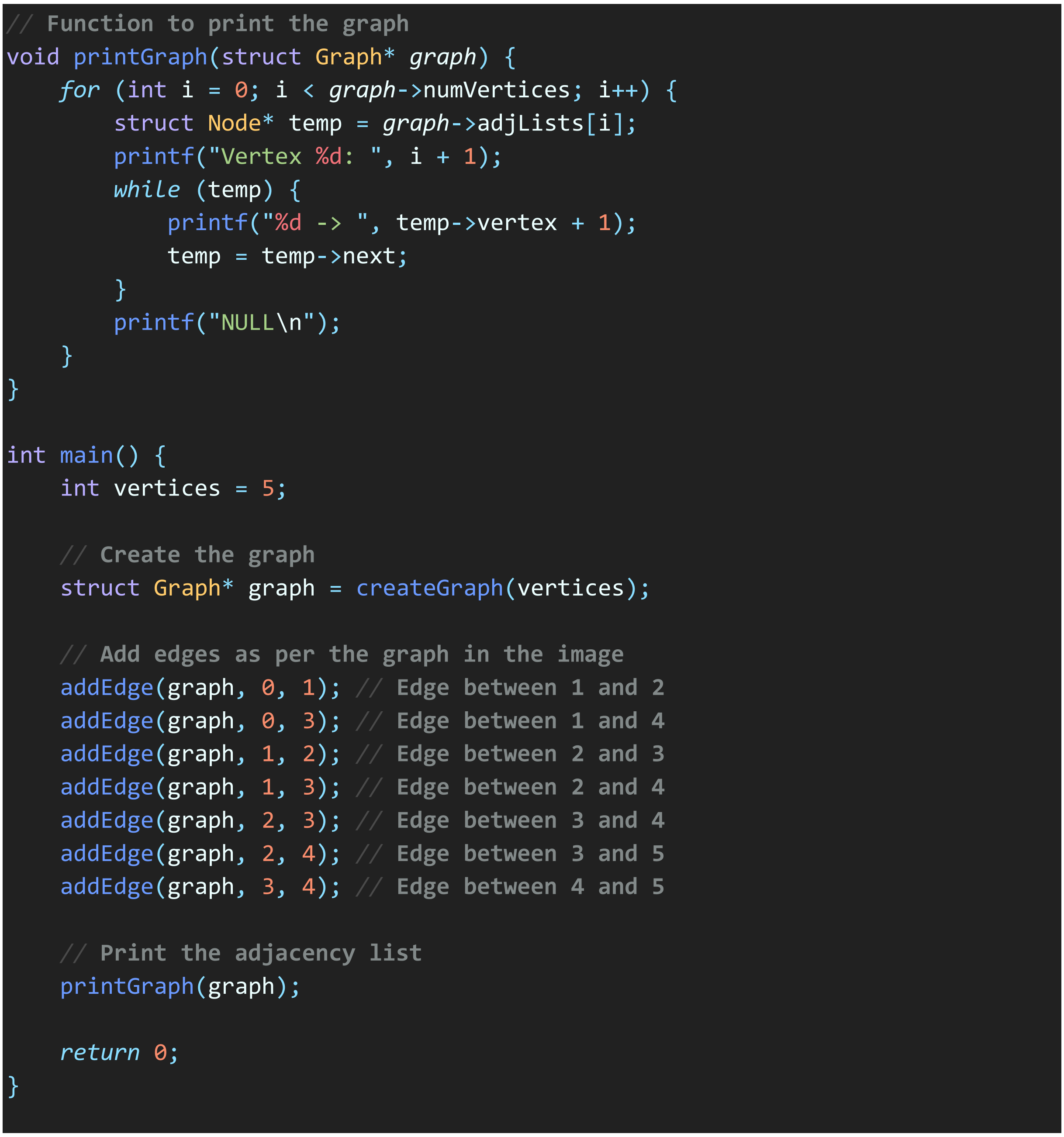


Code:

```
#include <stdio.h>
#include <stdlib.h>

// Structure to represent a node in the adjacency list
```

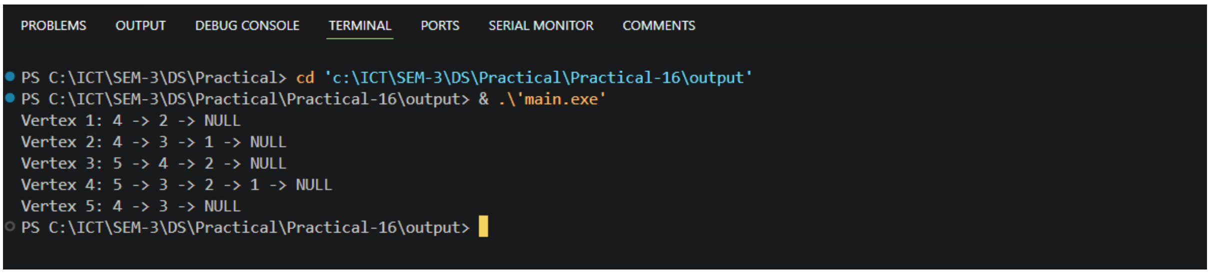


```c
struct Node {
    int vertex;
    struct Node* next;
};

// Structure to represent the graph
struct Graph {
    int numVertices;
    struct Node** adjLists; // Array of pointers to adjacency lists
};

// Function to create a new node
struct Node* createNode(int vertex) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->vertex = vertex;
    newNode->next = NULL;
    return newNode;
}

// Function to create a graph
struct Graph* createGraph(int vertices) {
    struct Graph* graph = (struct Graph*)malloc(sizeof(struct Graph));
    graph->numVertices = vertices;

    // Create an array of adjacency lists
    graph->adjLists = (struct Node**)malloc(vertices * sizeof(struct Node*));

    // Initialize each adjacency list as empty
    for (int i = 0; i < vertices; i++) {
        graph->adjLists[i] = NULL;
    }

    return graph;
}

// Function to add an edge to the graph
void addEdge(struct Graph* graph, int src, int dest) {
    // Add an edge from src to dest
    struct Node* newNode = createNode(dest);
    newNode->next = graph->adjLists[src];
    graph->adjLists[src] = newNode;

    // Add an edge from dest to src (since the graph is undirected)
    newNode = createNode(src);
    newNode->next = graph->adjLists[dest];
    graph->adjLists[dest] = newNode;
}
```



```
// Function to print the graph
void printGraph(struct Graph* graph) {
    for (int i = 0; i < graph->numVertices; i++) {
        struct Node* temp = graph->adjLists[i];
        printf("Vertex %d: ", i + 1);
        while (temp) {
            printf("%d -> ", temp->vertex + 1);
            temp = temp->next;
        }
        printf("NULL\n");
    }
}

int main() {
    int vertices = 5;

    // Create the graph
    struct Graph* graph = createGraph(vertices);

    // Add edges as per the graph in the image
    addEdge(graph, 0, 1); // Edge between 1 and 2
    addEdge(graph, 0, 3); // Edge between 1 and 4
    addEdge(graph, 1, 2); // Edge between 2 and 3
    addEdge(graph, 1, 3); // Edge between 2 and 4
    addEdge(graph, 2, 3); // Edge between 3 and 4
    addEdge(graph, 2, 4); // Edge between 3 and 5
    addEdge(graph, 3, 4); // Edge between 4 and 5

    // Print the adjacency list
    printGraph(graph);

    return 0;
}
```

Output:



```
PS C:\ICT\SEM-3\DS\Practical> cd 'c:\ICT\SEM-3\DS\Practical\Practical-16\output'
PS C:\ICT\SEM-3\DS\Practical\Practical-16\output> & .\'main.exe'
Vertex 1: 4 -> 2 -> NULL
Vertex 2: 4 -> 3 -> 1 -> NULL
Vertex 3: 5 -> 4 -> 2 -> NULL
Vertex 4: 5 -> 3 -> 2 -> 1 -> NULL
Vertex 5: 4 -> 3 -> NULL
PS C:\ICT\SEM-3\DS\Practical\Practical-16\output> 
```