Name: Vishal Bhimgonda Desai

Roll no: B78

PRN : 2425010086

C Program to Implement Adjacency List

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

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

struct Graph {
   int numVertices;
   struct Node** adjLists;
   int isDirected;
};

struct Node* createNode(int v) {
   struct Node* newNode = malloc(sizeof(struct Node));
   newNode->vertex = v;
   newNode->next = NULL;
   return newNode;
}

struct Graph* createGraph(int vertices, int isDirected) {
   struct Graph* graph = malloc(sizeof(struct Graph));
   graph->numVertices = vertices;
   graph->isDirected = isDirected;
```

```
    graph->adjLists = malloc(vertices * sizeof(struct Node*));
    for (int i = 0; i < vertices; i++) {
        graph->adjLists[i] = NULL;
    }
    return graph;
}

void addEdge(struct Graph* graph, int src, int dest) {
    struct Node* newNode = createNode(dest);
    newNode->next = graph->adjLists[src];
    graph->adjLists[src] = newNode;

    if (!graph->isDirected) {
        newNode = createNode(src);
        newNode->next = graph->adjLists[dest];
        graph->adjLists[dest] = newNode;
    }
}

void printGraph(struct Graph* graph) {
    printf("Vertex:  Adjacency List\n");
    for (int v = 0; v < graph->numVertices; v++) {
        struct Node* temp = graph->adjLists[v];
        printf("%d --->", v);
        while (temp) {
            printf(" %d ->", temp->vertex);
            temp = temp->next;
        }
        printf(" NULL\n");
    }
}
```

```
int main() {
    struct Graph* undirectedGraph = createGraph(3, 0);
    addEdge(undirectedGraph, 0, 1);
    addEdge(undirectedGraph, 0, 2);
    addEdge(undirectedGraph, 1, 2);

    printf("Adjacecncy List for Undirected Graph:\n");
    printGraph(undirectedGraph);

    struct Graph* directedGraph = createGraph(3, 1);
    addEdge(directedGraph, 1, 0);
    addEdge(directedGraph, 1, 2);
    addEdge(directedGraph, 2, 0);

    printf("\nAdjacecncy List for Directed Graph:\n");
    printGraph(directedGraph);

    return 0;
}
```

Output :

```
Adjacecncy List for Undirected Graph:
Vertex:  Adjacency List
0 ---> 2 -> 1 -> NULL
1 ---> 2 -> 0 -> NULL
2 ---> 1 -> 0 -> NULL

Adjacecncy List for Directed Graph:
Vertex:  Adjacency List
0 ---> NULL
1 ---> 2 -> 0 -> NULL
2 ---> 0 -> NULL
```