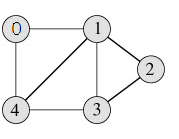
// Create a graph from Adjacency List



#include <stdio.h>

#include<stdlib.h>

#include<string.h>

#include<assert.h>

struct AdjListNode {

int data;

structAdjListNode \*next;

};

struct AdjList {

struct AdjListNode \*head;

};

struct Graph {

int V; // No of Nodes

struct AdjList \*array;

};

Struct Graph\* create\_Graph(int node)

{

struct Graph \*g;

g = (struct Graph \*)malloc(sizeof(struct Graph));

assert(g != NULL);

g->V = node;

g->array = (struct AdjList \*)malloc(sizeof(struct AdjList ) \* node); // Create an array of the length of nodes

assert(g->array != NULL);

return g;

}

struct AdjListNode\* createNode(int val)

{

struct AdjListNode \*node = (struct AdjListNode \*)malloc(sizeof(struct AdjListNode));

assert(node != NULL);

node->data = val;

node->next = NULL;

return node;

}

void add\_Edge(struct Graph \*g, int src, int dest) // Add a edge in a graph

{

struct AdjListNode \*node = newNode(dest);

node->next = g->array[src];

g->array[src] = node;

// if the graph is undirected

node = newNode(src);

node->next = g->array(dest);

g->array[dest] = node;

}

int main()

{

int node, edge, i, src, dest;

scanf(“%d%d”,&node,&edge);

struct Graph \*g = create\_Graph(node);

for(i = 0; i < edge; ++i) {

scanf(“%d%d”,&src,&dest);

add\_Edge(g, src, dest);

}

return 0;

}

// Creation of Graph in C takes 50 Lines of code

=====================================================================================

#include <iostream>

using namespace std;

vector<int> adj[10];

int main()

{

int x, y, node, edge;

cin >> node >> edge;

for(i = 0; i < node; i++)

{

cin >> x >> y;

adj[x].push\_back[y];

}

// print the graph

for( i = 1; i < node; i++)

{

for(int j = 0; j < adj[i].size(); ++j)

{

if(j == adj[j].size() -1) { // If it is last node

cout << adj[i][j] ;

}

else {

cout << adj[i][j] << “--🡪”;

}

}

}

return 0;

}

//Same thing in C++ within 10 lines of code with less complexity because function are already present

--------------------------------------------------------------------------------------------------------------------------------------