GRAPH by MATRIX

#include<iostream>

#include<iomanip>

using namespace std;

void printmat(int mat[][10],int n)

{

int i ,j;

cout<<"\n\n "<<setw(4)<<" ";

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

cout<<setw(3)<<"("<<i+1<<")";

cout<<"\n\n";

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

{

cout<<setw(3)<<"("<<i+1<<")";

for(j = 0;j<n;j++)

{

cout<<setw(4)<<mat[i][j];

}

cout<<"\n\n";

}

}

int main()

{

int i,j,v;

cout<<"enter the no of vetex";

cin>>v;

int mat[10][10];

cout<<"\n";

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

for(j = i;j<v;j++)

{

if(i!=j)

{

cout<<"enter 1 if vertex "<<i+1<<" is adjacent "<<j+1<<" ,other wise 0 ";

cin>>mat[i][j];

mat[j][i]=mat[i][j];

}

else

mat[i][j]=0;

}

}

printmat(mat,v);

return 0;

}

LINKED LIST

/\*

Graph implementation following tutorial http://www.geeksforgeeks.org/graph-and-its-representations/

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

//struct for an adjacency list node

struct AdjListNode{

int data;

AdjListNode \*next;

};

//struct for an adjacency list

struct AdjList{

AdjListNode \*head; //pointer to head node of list

};

//struct for a graph. A graph as an array of adjacency lists

//Size of array will be V (total vertices)

struct Graph{

int V;

AdjList \*arr;

};

//create a new node

AdjListNode\* newAdjListNode(int data){

AdjListNode \*nptr=new AdjListNode;

nptr->data=data;

nptr->next=NULL;

return nptr;

}

//function to create a graph of V - vertices

Graph\* createGraph(int V){

Graph \*graph=new Graph;

graph->V=V;

//create an array of adjacency list. size of array - V

graph->arr=new AdjList[V];

//initialize with NULL (e.g root=NULL)

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

graph->arr[i].head=NULL;

}

return graph;

}

//add an edge to an undirected Graph

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

//Add an edge from src to dest. A new node added to the adjacency list of src

//node added at beginning

AdjListNode \*nptr=newAdjListNode(dest);

nptr->next=graph->arr[src].head;

graph->arr[src].head=nptr;

//connect from dest to src (since undirected)

nptr=newAdjListNode(src);

nptr->next=graph->arr[dest].head;

graph->arr[dest].head=nptr;

}

//function to print the graph

void printGraph(Graph\* graph){

//loop over each adjacent list

for(int i=0;i<graph->V;i++){

AdjListNode \*root=graph->arr[i].head;

cout<<"Adjacency list of vertex "<<i<<endl;

//loop over each node in list

while(root!=NULL){

cout<<root->data<<" -> ";

root=root->next;

}

cout<<endl;

}

}

int main(){

//create a new graph

int totalVertices=4;

Graph \*graph;

graph=createGraph(totalVertices);

//connect edges

addEdge(graph,0,1);

addEdge(graph,0,2);

addEdge(graph,0,3);

addEdge(graph,1,3);

addEdge(graph,2,3);

//printGraph(graph);

}