```
//read graph as adjacency list and find out indegree,outdegree
//and total degree of each node in graph.
#include<stdio.h>
#include<stdlib.h>

typedef struct node

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

//heads of linked list
node *ll[20];
int n;
//create adjacency list
void read_graph();
//insert an edge (vi,vj) in the adjacency list
void insert(int,int);
void degree();

void main()
{
int i,v;
read_graph();
degree(n);
printf("\n\n");
}

void read_graph()
{
int i,vi,vj,no_of_edges;
printf("\nEnter number of vertices:");
scanf("%d",&n);

//initialise ll[] with a null
ll[0]=NULL;
//read edges and insert them in G[]

printf("\nEnter number of edges:");
scanf("%d",&no_of_edges);

for(i=0;i<no_of_edges;i++)
{
printf("Enter an edge(u v):");
```

```
scanf("%d%d",&vi,&vj);
insert(vi,vj);
}
}

void insert(int vi,int vj)
{
node *p,*q;

//acquire memory for the new node
q=(node*)malloc(sizeof(node));
q->vertex=vj;
q->next=NULL;

//insert the node in the linked list number vi
if(ll[vi]==NULL)

ll[vi]=q;
else
{
//go to end of the linked list
p=ll[vi];

while(p->next!=NULL)
p=p->next;
p->next=q;
}
}

void degree(int n)
{
int i,indegree[10],outdegree[10];
struct node *p;
for(i=0;i<10;i++)indegree[i]=0;
for(i=0;i<10;i++)outdegree[i]=0;
for (i = 0; i < n ; i++ )
{

p=ll[i];
while(p!=NULL)
{
indegree[p->vertex]+=1;
outdegree[i]+=1;
p=p->next;
}
```

```
} // for
for(i=0;i<n;i++)
printf("\nIndegree, Outdegree and Total Degree of vertex %d is %d, %d,

%d",i,indegree[i],outdegree[i],indegree[i]+outdegree[i]);
}
```