bfs\_07.cpp

**Compile:** g++ bfs\_07.cpp -o bfs\_07

**Run:** ./bfs\_07

**Program:**

#include<iostream>

using namespace std;

intadj[20][20];

int visited[20];

structst

{

intfront,rear;

int queue[20];

}s;

int empty()

{

if(s.front==s.rear+1)

return 1;

else

return 0;

}

void insert(int item)

{

s.queue[++s.rear]=item;

if(s.rear==1) s.front++;

}

int remove()

{

if(empty()==1) cout<<"\n\nQueue is empty ";

else

returns.queue[s.front++];

}

voidbfs(intinode,int n)

{

intu,i;

u=inode;

visited[inode]=1;

cout<<"\n\nBFS traversal of graph \t"<<inode;

insert(inode);

while(!empty())

{

u=remove();

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

{

if((adj[u][i]==1) && (visited[i]==0))

{

insert(i);

visited[i]=1;

cout<<"\t"<<i;

}

}

}

}

int main()

{

inti,j,m,a,b,v,n;

cout<<"\n\n\t\t\tBreadth First Traversal ";

cout<<"\n\n\n\nenter number of nodes and edges in graph ";

cin>>n>>m;

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

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

adj[i][j]=0;

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

{

cout<<"\n\nenter an edge ";

cin>>a>>b;

adj[a][b]=1;

adj[b][a]=1;

}

cout<<"\n\n\nenter initial node ";

cin>>v;

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

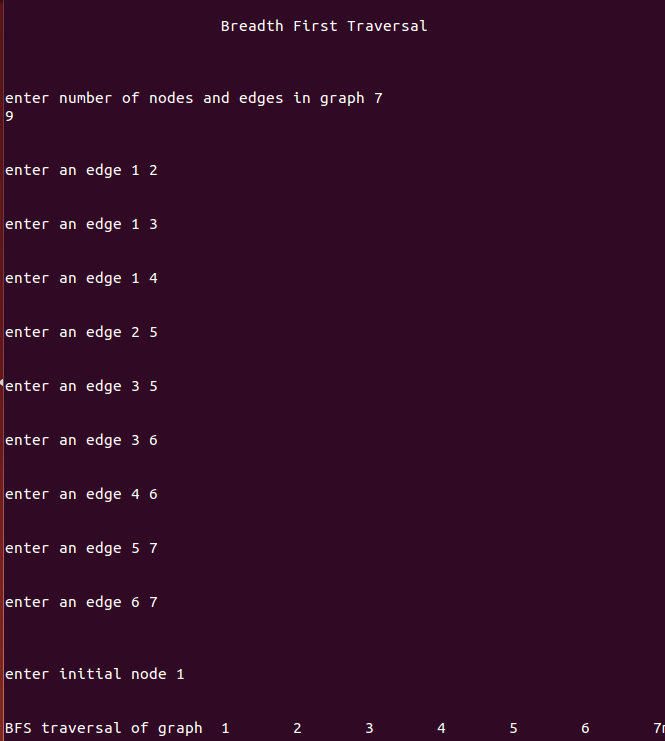
visited[i]=0;

bfs(v,n);

}

**Output:**

****

****