dfs\_07.cpp

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

**Run:** ./dfs\_07

**Program:**

#include<iostream>

Using namespace std;

intadj[20][20];

int visited[20];

structst

{

int top;

int stack[20];

}s;

void push(int item)

{

s.stack[++s.top]=item;

}

int pop()

{

returns.stack[s.top--];

}

int empty()

{

if(s.top==-1)

return 1;

else

return 0;

}

voiddfs(intinode,int n)

{

intu,i;

s.top=-1;

push(inode);

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

while(!empty())

{

u=pop();

if(visited[u]==0)

{

cout<<"\n\n"<<u;

visited[u]=1;

}

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

{

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

{

push(u);

visited[i]=1;

cout<<"\t"<<i;

u=i;

goto back;

}

}

}

}

void main()

{

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

cout<<"\n\n\t\t\tDepth 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;

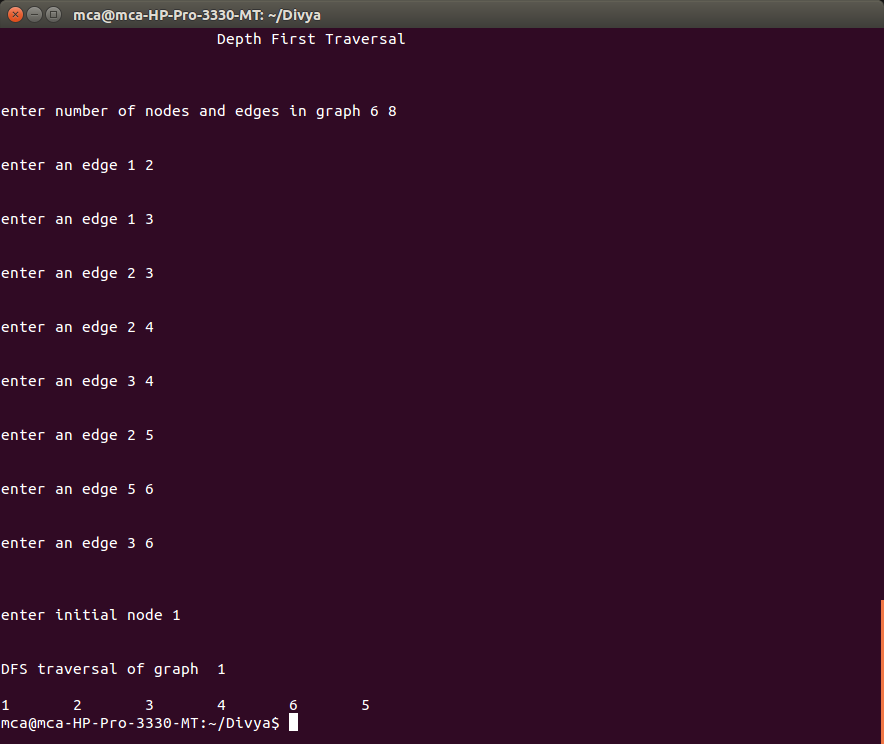
for(i=0;i<n;i++)

visited[i]=0;

dfs(v,n);

}

**Output:**

****