**//DFS for a graph**

#include<stdio.h>

#include<conio.h>

int mat[10][10],stack[10],Next,top, rare,v,open[10],close[10];

int i,j,k,n;

char ch='A';

int main()

{

int m;

printf("\n\t DFS for graph \n");

printf("\n Enter no of vertices : ");

scanf("%d",&n);

printf("\n Enter no of edges : ");

scanf("%d",&m);

printf("\n Enter Edges : \n");

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

{

printf("\n\n For edge %d) : ",k);

printf("\t Enter 1st vertex : ");

scanf("%d",&i);

printf("\t\t Enter 2nd vertex : ");

scanf("%d",&j);

mat[i][j]=1;

printf("\t\t\t\t [ %c,%c ] ",ch+(i-1),ch+(j-1));

}

printf("\n\n Enter initial vertex : ");

scanf("%d",&v);

printf("\t\t\t ( %c ) ",ch+(v-1));

printf("\n\n DFS - Visited vertices path : \t");

printf("[ %d ]",v);

printf("( %c ) ",ch+(v-1));

close[v]=1;

k=1;

while(k<n)

{

for(j=n;j>=1;j--)

if(mat[v][j]!=0 && close[j]!=1 && open[j]!=1)

{

open[j]=1;

stack[top]=j;

top++;

}

v=stack[--top];

printf(" -> %d ",v);

printf("( %c )",ch+(v-1));

k++;

open[v]=0;

close[v]=1;

}

getch();

}

/\*

**Output :**

DFS for graph

Enter no of vertices : 7

Enter no of edges : 6

Enter Edges :

For edge 1) : Enter 1st vertex : 1

Enter 2nd vertex : 2

[ A,B ]

For edge 2) : Enter 1st vertex : 1

Enter 2nd vertex : 3

[ A,C ]

For edge 3) : Enter 1st vertex : 2

Enter 2nd vertex : 4

[ B,D ]

For edge 4) : Enter 1st vertex : 2

Enter 2nd vertex : 5

[ B,E ]

For edge 5) : Enter 1st vertex : 3

Enter 2nd vertex : 6

[ C,F ]

For edge 6) : Enter 1st vertex : 3

Enter 2nd vertex : 7

[ C,G ]

Enter initial vertex : 1 ( A )

DFS - Visited vertices path : [ 1 ]( A ) -> 2 ( B ) -> 4 ( D ) -> 5 ( E ) -> 3 ( C ) -> 6 ( F ) -> 7 ( G )

\*/

**//BFS for a graph**

#include<stdio.h>

#include<conio.h>

int mat[10][10],qu[10],Next,rare,v,

open[10],close[10];

int i,j,k,n;

char ch='A';

int main()

{

int m;

printf("\n\t BFS for graph \n");

printf("\n Enter no of vertices : ");

scanf("%d",&n);

printf("\n Enter no of edges : ");

scanf("%d",&m);

printf("\n Enter Edges : \n");

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

{

printf("\n\n For edge %d) : ",k);

printf("\t Enter 1st vertex : ");

scanf("%d",&i);

printf("\t\t Enter 2nd vertex : ");

scanf("%d",&j);

printf("\t\t\t\t [%c, %c] ",ch+(i-1),ch+(j-1));

mat[i][j]=1;

}

printf("\n\n Enter initial vertex : ");

scanf("%d",&v);

printf("\t\t\t ( %c ) ",ch+(v-1));

printf("\n\n BFS - Visited vertices path : ");

printf("\t [ %d ]",v);

printf("[ %c ] ",ch+(v-1));

close[v]=1;

k=1;

while(k<n)

{

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

if(mat[v][j]!=0 && close[j]!=1 && open[j]!=1)

{

open[j]=1;

qu[rare++]=j;

}

v=qu[Next++];

printf(" -> %d ",v);

k++;

open[v]=0;

close[v]=1;

printf("( %c )",ch+(v-1));

}

getch();

}

/\*

**Output :**

BFS for graph

Enter no of vertices : 7

Enter no of edges : 6

Enter Edges :

For edge 1) : Enter 1st vertex : 1

Enter 2nd vertex : 2

[A, B]

For edge 2) : Enter 1st vertex : 1

Enter 2nd vertex : 3

[A, C]

For edge 3) : Enter 1st vertex : 2

Enter 2nd vertex : 4

[B, D]

For edge 4) : Enter 1st vertex : 2

Enter 2nd vertex : 5

[B, E]

For edge 5) : Enter 1st vertex : 3

Enter 2nd vertex : 6

[C, F]

For edge 6) : Enter 1st vertex : 3

Enter 2nd vertex : 7

[C, G]

Enter initial vertex : 1

( A )

BFS - Visited vertices path : [ 1 ][ A ] -> 2 ( B ) -> 3 ( C ) -> 4 ( D ) -> 5 ( E ) -> 6 ( F ) -> 7 ( G )

\*/