**Programs**

1. Write a program for implementation of graph traversals by applying:
2. BFS b. DFS

Ans:

#include<stdio.h>

#include<conio.h>

int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];

int delete();

void add(int item);

void bfs(int s,int n);

void dfs(int s,int n);

void push(int item);

int pop();

void main()

{

int n,i,s,ch,j;

clrscr();

printf("Enter number of vertices:");

scanf("%d",&n);

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

{

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

{

printf("Enter 1 if %d has a node with %d else 0:",i,j);

scanf("%d",&a[i][j]);

}

}

printf("The adjacency matrix is:\n");

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

{

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

{

printf(" %d",a[i][j]);

}

printf("\n");

}

while(ch!=3)

{

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

vis[i]=0;

printf("\n-----------------MENU-----------------");

printf("\n1.B.F.S \t2.D.F.S \t3.Exit");

printf("\n--------------------------------------");

printf("\nEnter your choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:printf("Enter the source vertex:");

scanf("%d",&s);

bfs(s,n);

break;

case 2:printf("Enter the source vertex:");

scanf("%d",&s);

dfs(s,n);

break;

case 3:exit(0);

default:printf("Invalid choice..");

}

}

getch();

}

void bfs(int s,int n)

{

int p,i;

add(s);

vis[s]=1;

p=delete();

if(p!=0)

printf("%d\t",p);

while(p!=0)

{

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

{

if((a[p][i]!=0)&&(vis[i]==0))

{

add(i);

vis[i]=1;

}

p=delete();

if(p!=0)

printf("%d\t",p);

}

}

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

{

if(vis[i]==0)

bfs(i,n);

}

}

void add(int item)

{

if(rear==19)

printf("Queue full");

else

{

if(rear==-1)

{

q[++rear]=item;

front++;

}

else

q[++rear]=item;

}

}

int delete()

{

int k;

if((front>rear)||(front==-1))

return(0);

else

{

k=q[front++];

return(k);

}

}

void dfs(int s,int n)

{

int i,k;

push(s);

vis[s]=1;

k=pop();

if(k!=0)

printf("%d\t",k);

while(k!=0)

{

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

{

if((a[k][i]!=0)&&(vis[i]==0))

{

push(i);

vis[i]=1;

}

k=pop();

if(k!=0)

printf("%d\t",k);

}

}

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

{

if(vis[i]==0)

dfs(i,n);

}

}

void push(int item)

{

if(top==19)

printf("Stack overflow ");

else

stack[++top]=item;

}

int pop()

{

int k;

if(top==-1)

return(0);

else

{

k=stack[top--];

return(k);

}

}

Output:



