**Code:-**

#include <stdio.h>

#define limit 40

int queue[limit],stack[limit],front=-1,rear=-1,top=-1,visit[limit]={};

void enqueue(int val){

if(rear==limit-1)

printf("Queue is full");

else{

rear+=1;

queue[rear]=val;

}

}

int dequeue(){

int val;

if(front==rear)

{

val=queue[front++];

front=rear=-1;

return(val);

}

else

{

val=queue[front++];

return(val);

}

}

int empty\_queue(){

if(front==-1)

return 1;

return 0;

}

void bfs(int a[20][20],int n,int v1){

int v2;

enqueue(v1);

visit[v1]==1;

while(!empty\_queue()){

v1=dequeue();

printf("%d\t",v1);

for(v2=0;v2<n;v2++){

if(a[v1][v2]==1 && visit[v2]==0){

enqueue(v2);

visit[v2]==1;

}

}

}

}

void push(int num){

if(top==limit-1)

printf("Stack is Full");

else{

top +=1;

stack[top]=num;

}

}

int pop(){

int val;

if(top==-1)

printf("Stack is empty");

else{

val=stack[top];

top -=1;

return val;

}

}

int empty(){

if(top==-1)

return 1;

return 0;

}

void dfs(int a[20][20],int n,int val){

int v1,v2;

push(val);

while(!empty()){

v1=pop();

if(visit[v1]==0){

printf("%d\t",v1);

visit[v1]=1;

}

for(v2=0;v2<n;v2++){

if(a[v1][v2]==1 && visit[v2]==0)

push(v2);

}

}

}

create\_graph(int a[20][20],int n){

int flag,v1,v2,c;

printf("You Want a Directed or Undirected Graph 0/1 ");

scanf("%d",&flag);

while(c!=0){

printf("Enter edge for v1,v2 : ");

scanf("%d %d",&v1,&v2);

if(flag==1){

a[v1][v2]=1;

a[v2][v1]=1;

}

else

a[v1][v2]=1;

printf("do you want onr more edge: ");

scanf("%d",&c);

}

}

display\_matrix(int a[20][20],int n){

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

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

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

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

}

printf("\n");

}

}

int main(void) {

int s,n,a[20][20]={},ch=0,c;

printf("Enter Number of vertices : ");

scanf("%d",&n);

do{

printf("Enter Your Choice\n1.Create\n2.DFS\n3.BFS\n4.Exit\n");

scanf("%d",&c);

switch(c){

case 1:

create\_graph(a,n);

break;

case 2:

printf("Enter Start Point: ");

scanf("%d",&s);

dfs(a,n,s);

break;

case 3:

visit[limit]={};

printf("Enter Start Point: ");

scanf("%d",&s);

bfs(a,n,s);

break;

}while(ch!=4);

}

return 0;

}

**Output:-**



