```
#include<stdio.h>
#include<conio.h>

int stk[10],adj[51][51],visited[51];

void DFS(int initial_node,int n);
void createGraph()
{
	int c,n,i,j,parent, adj_parent,initial_node;
	int ans=0,ans1=0;
	printf("\nEnter total number of elements  : ");
	scanf("%d",&n);
	for(i=1;i<=n;i++)
		for(j=1;j<=n;j++)
			adj[i][j]=0;
	for(c=1;c<=50;c++)
		visited[c]=0;
	do
	{
		printf("Enter parent node  : ");
		scanf("%d",&parent);
		do
		{
			printf("Enter adjacent node for node %d: ",parent);
			scanf("%d",&adj_parent);
```

```
                adj[parent][adj_parent]=1;
                adj[adj_parent][parent]=1;
                printf("Want to enter more adjacent nodes ? (press 1 for yes):");
                fflush(stdin);
                scanf("%d",&ans1);
        }while(ans1==1);
        printf("Continue to add another graph node ?(press 1 for yes):");
        fflush(stdin);
        scanf("%d",&ans);
}while(ans==1);
printf("\nAdjacency matrix for your graph: \n");
for(i=1;i<=n;i++)
{
        for(j=1;j<=n;j++)
                printf("%d ",adj[i][j]);
        printf("\n");
}
printf("\nYour undirected graph: ");
for(i=1;i<=n;i++)
{
        printf("\nVertex %d is connected to : ",i);
        for(j=1;j<=n;j++)
        {
                if(adj[i][j]==1)
                        printf("%d ",j);
```

```
                }
        }
        printf("\nEnter initial node for DFS traversal: ");
        scanf("%d",&initial_node);
        DFS(initial_node,n);
}
int top=-1;
void push(int item)
{
        if(top==9)
                printf("\nOVERFLOW.");
        else
                stk[++top]=item;
}
int pop()
{
        if(top==-1)
                printf("\nSTACK UNDERFLOW.");
        else
                return (stk[top--]);
        return 0;
}

void DFS(int initial_node,int n)
{
```

```
        int u,i;
        top=-1;
        push(initial_node);
        printf("\nDFS traversal for given graph is: ");
        while(top>=0)
        {
                u=pop();
                if(visited[u]==0)
                {
                        printf("%d ",+u);
                        visited[u]=1;
                }
                for(i=1;i<=n;i++)
                {
                        if((adj[u][i]==1)&&(visited[i]==0))
                        {
                                push(u);
                                visited[i]=1;
                                printf("%d ",i);
                                u=i;
                        }
                }
        }
}
void main()
```

```
{
        clrscr();
        createGraph();
        getch();
}
```