```
#include<stdio.h>
#include<conio.h>
int items[10];
int front, rear;
void insert(int e)
{
	if(rear==9)
		printf("Queue overflow.");
	else
		items[++rear]=e;
}
int empty()
{
	return (rear<front? 1:0);
}
int remove1()
{
	int x=0;
	if(empty()==1)
	{
		printf("Queue underflow.");
		return 0;
	}
	else
	{
```

```
                x=items[front++];
        }
        return x;
}
int adj[51][51];
int visited[51];
void BFS(int initial_node,int n);
void createGraph()
{
        int n,i,c,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;
        printf("\nEnter graph structure for BFS:");
        do
        {
                printf("\nEnter parent node :");
                scanf("%d",&parent);
                do
                {
```

```
                printf("Enter adjacent node for parent node %d:",parent);
                scanf("%d",&adj_parent);
                adj[parent][adj_parent]=1;
                adj[adj_parent][parent]=1;
                printf("\nContinue to add adjacent node (press 1 for yes):");
                fflush(stdin);
                scanf("%d",&ans1);
        }while(ans1==1);
        printf("Continue to add graph node (press 1)?");
        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 matrix is : ");
for(i=1;i<=n;i++)
{
        printf("\nVertex %d ",i," is connected to :");
        for(j=1;j<=n;j++)
        {
                if(adj[i][j]==1)
```

```
                        printf(" %d",j);
                    }
                }
                printf("\nEnter initial node for BFS traversal : ");
                scanf("%d",&initial_node);
                BFS(initial_node,n);
        }
void BFS(int initial_node,int n)
{
        int u,i;
        u=initial_node;
        visited[initial_node]=1;
        printf("\nBFS traversal for given graph is: ");
        printf("%d",initial_node);
        insert(initial_node);
        while(!empty())
        {
                u=remove1();
                for(i=1;i<=n;i++)
                {
                        if((adj[u][i]==1)&&(visited[i]==0))
                        {
                                insert(i);
                                visited[i]=1;
                                printf(" %d",i);
```

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