8. IMPLEMENTATION OF BFS SHORTEST PATH:

```
#include<stdio.h>
int search(int arr[],int n,int key)
{
int i;
for(i=0;i<n;i++)
if(arr[i]==key)
return 0;
//return 1;
}
int indexof(int a[],int n,int key1)
{
int i;
for(i=0;i<n;i++)
if(a[i]==key1)
return i;
//return 0;
}
int main()
{
int i,j,n,k=0,l=0,k1=0,flag=0,arr[10][10],p=0,start,end,root[10],bfs[10],z=-1,in=0,re[10],indo=0;
scanf("%d",&n);
for(i=0;i<n;i++)
for(j=0;j<n;j++)
scanf("%d",&arr[i][j]);
```

```
scanf("%d",&start);
scanf("%d",&end);
bfs[k++] = start;
root[k1++] = start;
l++;
while(k<n){
flag = 0;
for(i=0;i<n;i++)
if(arr[bfs[l-1]][i]==1 && search(bfs,n,i)){
bfs[k++] = i;
flag = 1;
root[k1++] = bfs[l-1];
}
l++;
if(search(bfs,n,end)==0)
break;
}
for(i=0;i<k;i++)
printf("%d ",bfs[i]);
printf("\n");
for(i=0;i<k1;i++)
printf("%d ",root[i]);
printf("\n");
z = end;
p = k-1;
```

```
while(z!=start && in<n){
re[indo++] = z;
p = indexof(bfs,n,root[p-1]);
z = bfs[p];
++in;
}
//printf("%d",start);
re[indo++] = start;
printf("path length = %d",indo-1);
printf("\npath:\n");
for(j=indo-1;j>=0;j--)
printf("%d ",re[j]);
return 0;
}
```

9. IMPLEMENTATION OF DFS:

```
#include<stdio.h>
#include<conio.h>
#define max 20
int g[max][max],v[max],n;
void DFS(int);
int main()
{
int i,u,v,e,s;
printf("Enter No of Edges : ");
```

```
scanf("%d",&e);
printf("Enter No of Nodes : ");
scanf("%d",&n);
for(i=1;i<=e;i++)
{
printf("Enter Edge %d : ",i);
scanf("%d%d",&u,&v);
g[u][v]=g[v][u]=1;
}
printf("Enter Source : ");
scanf("%d",&s);
DFS(s);
getch();
}
void DFS(int s){
int i;
printf(" %d",s);
v[s]=1;
for(i=1;i<=n;i++)
{
if(g[s][i]!=0&&v[i]==0)
DFS(i);
}
}
```