Program 3

```
#include <stdio.h>
#include<stdlib.h>
void dfs(int a[10][10], int n, int v[10], int source)
{
 int i;
 v[source] = 1; //add source to v(indicated source is visited)
 for(i=0; i<n; i++)
 if(a[source][i] == 1 && v[i] == 0)
 {
 printf("%d ",i);
 dfs(a,n,v,i);//recursive call for dfss

 }
}
void bfs(int a[10][10], int n, int v[10], int source)
{
 int q[10], front=0, rear=-1;
 int node, i;
 v[source] = 1; //add source to v(indicated source is visited)
 q[++rear] = source;
 while(front <= rear)//as long as queue is empty
 {
 node = q[front++];/*delete the next vertex to be explored from q*/
 for(i=0;i<n;i++)
 if(a[node][i] == 1 && v[i] == 0)
 {
 printf("%d ",i);
 v[i] = 1;
 q[++rear] = i;/*insert new vertex to q for exploration*/
 }
 }//end while
}//end bfs
int main()
{
 int n,ch;
 int a[10][10];
 int v[10];
 int source;
 int i, j;
int count = 0;
printf("Enter no of nodes: ");
scanf("%d",&n);//read the total number of nodes
printf("\n Read Adjacency matrix \n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);//read the adjacency matrix
printf("\n DFS\n");
for(i=0;i<n;i++) //mark all as unvisited
v[i] = 0;

for(i=0;i<n;i++)
{
if(v[i] == 0)
{
printf("%d ",i);
dfs(a,n,v,i);//call dfs method to check connectivity
count++;
}
}
if(count == 1)
printf("\nGraph is Connected\n");
else
printf("\nGraph is NOT Connected with %d Components\n",count);
printf("\n BFS\n");
for(i=0;i<n;i++)
v[i] = 0;

count = 0;
for(i=0;i<n;i++)
{
if(v[i] == 0)
{
printf("%d ",i);
bfs(a,n,v,i);//call bfs method to check connectivity
count++;
}
}
if(count == 1)
printf("\nGraph is Connected");
else
printf("\nGraph is NOT Connected with %d Components\n",count);
return 0;
```

```
}
```