WEEK 1

Write program to do the following:

a. Print all the nodes reachable from a given starting node in a digraph using

BFS method.

b. Check whether a given graph is connected or not using the DFS method.

a)BFS

Code:

```
#include<stdio.h>
#include<conio.h>
int a[15][15],n;
void bfs(int);
void main() {
 int i,j,src;
 printf("\nEnter the no of nodes:\t");
 scanf("%d",&n);
 printf("\nEnter the adjacency matrix:\n");
 for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
     scanf("%d",&a[i][j]);
 printf("\nEnter the source node:\t");
 scanf("%d",&src);
```

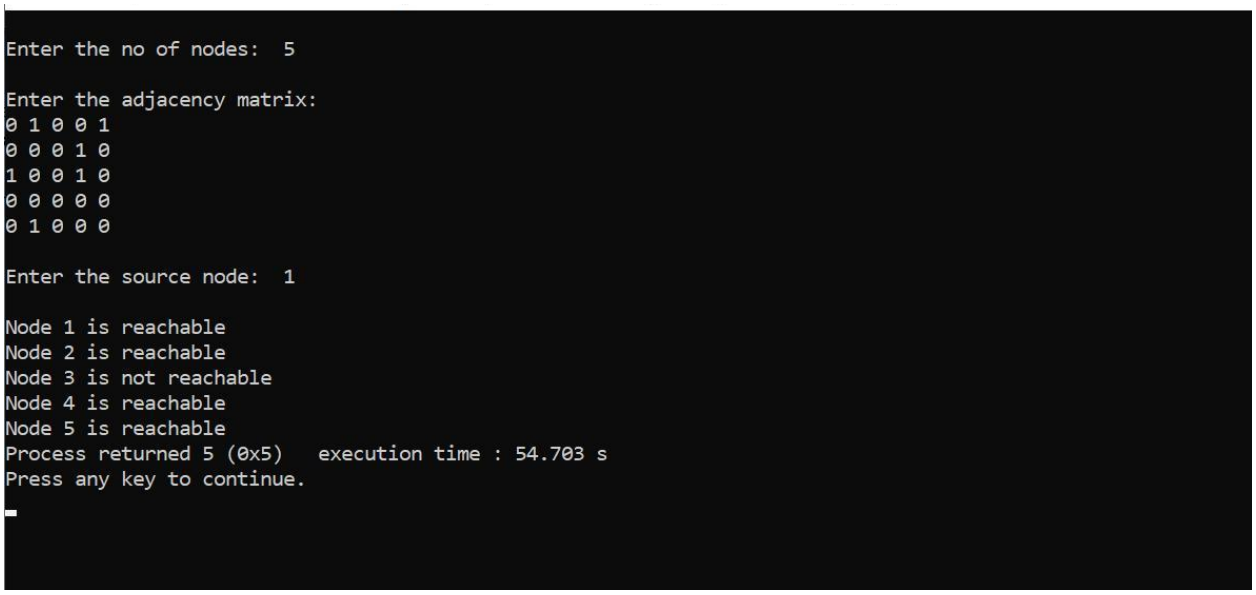
```
 bfs(src);
}
void bfs(int src) {

 int q[15],f=0,r=-1,vis[15],i,j;
 for(j=1;j<=n;j++)
    vis[j]=0;
 vis[src]=1;
 r=r+1;
 q[r]=src;
 while(f<=r) {
  i=q[f];
  f=f+1;
  for(j=1;j<=n;j++)
  {
   if(a[i][j]==1&&vis[j]!=1) {
    vis[j]=1;
    r=r+1;
    q[r]=j;
   }
  }
 }
 for(j=1;j<=n;j++) {
  if(vis[j]!=1)
   printf("\nNode %d is not reachable",j);
  else
```

```
   printf("\nNode %d is reachable",j);

 }
}
```

Output:



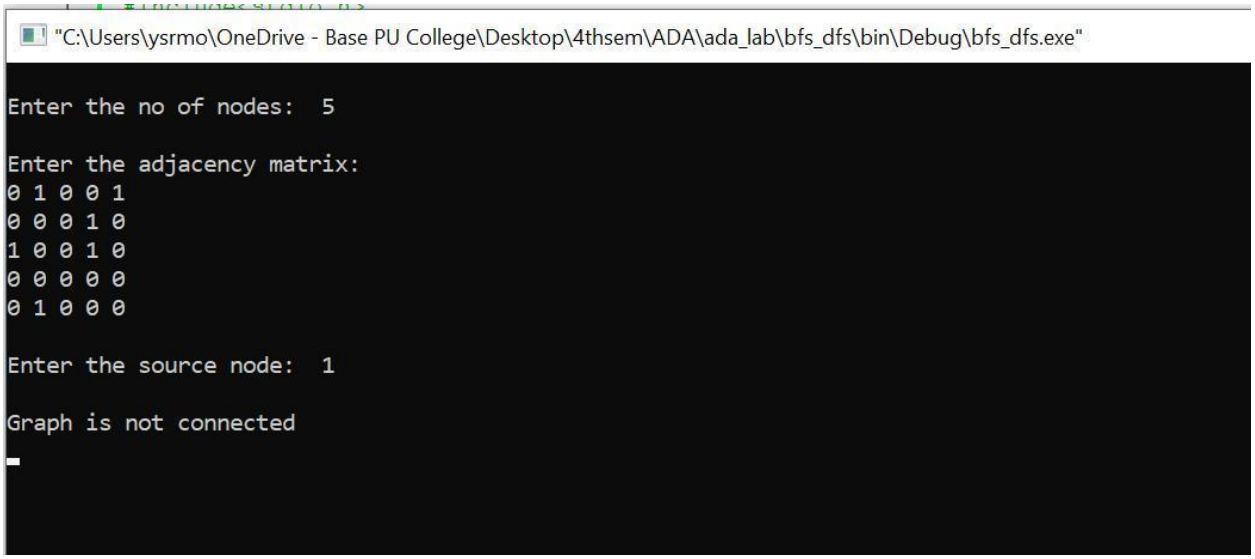
b)DFS

Code:

```
#include<stdio.h>
#include<conio.h>
int a[10][10],n,vis[10];
int dfs(int src){
    int j;
     vis[src]=1;
     for(j=1;j<=n;j++)
      if(a[src][j]==1&&vis[j]!=1)
       dfs(j);
     for(j=1;j<=n;j++) {
      if(vis[j]!=1)
       return 0;
     }
     return 1;
}
void main()
{
 int i,j,src,ans;
 for(j=1;j<=n;j++)
  vis[j]=0;
 printf("\nEnter the no of nodes:\t");
 scanf("%d",&n);
 printf("\nEnter the adjacency matrix:\n");
```

```
 for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
   scanf("%d",&a[i][j]);
 printf("\nEnter the source node:\t");
 scanf("%d",&src);
 ans=dfs(src);
 if(ans==1)
  printf("\nGraph is connected\n");
 else
  printf("\nGraph is not connected\n");
 getch();
}
```

Output:


"C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\ADA\ada_lab\bfs_dfs\bin\Debug\bfs_dfs.exe"
Enter the no of nodes: 5
Enter the adjacency matrix:
0 1 0 0 1
0 0 0 1 0
1 0 0 1 0
0 0 0 0 0
0 1 0 0 0
Enter the source node: 1
Graph is not connected