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 DFS method.

Code:

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

int a[10][10],n,vis[10];
int dfs(int root){
    int j;
     vis[root]=1;
     for(j=1;j<=n;j++)
      if(a[root][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,root,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",&root);
 ans=dfs(root);
 if(ans==1)
  printf("\nGraph is connected\n");
 else
  printf("\nGraph is not connected\n");
 getch();
}
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc ada.c -o ada } ; if ($?) { .\ada }

Enter the no of nodes:  4

Enter the adjacency matrix:
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0

Enter the source node:  1

Graph is connected
```

Code:

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

int a[15][15],n;
void bfs(int);

void main()  {
 int i,j,root;

 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",&root);
```

```
 bfs(root);
}
void bfs(int root) {
 int q[15],f=0,r=-1,vis[15],i,j;
 for(j=1;j<=n;j++)
    vis[j]=0;
 vis[root]=1;
 r=r+1;
 q[r]=root;
 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:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc ada.c -o ada } ; if ($?) { .\ada }

Enter the no of nodes:  4

Enter the adjacency matrix:
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0

Enter the source node:  1

Node 1 is reachable
Node 2 is reachable
Node 3 is reachable
Node 4 is reachable
PS D:\VS Code\OS>
```

Observation:

15-06-23

① ~~Print~~ Write the code for the following.

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

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

ⓑ

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

int a[10][10], n, vis[10];
int dfs(int root){

    int j;
    vis[root]=1;
    for(j=1; j<=n; j++)
      if(a[root][j]==1 && vis[j]!=1)
        dsf(j);
    for(j=1; j<=n; j++){

        if(vis[j]!=1)
          return 0;
        else
          return 1;
  }

void main(){
 int i, j, root, ans;
 for(j=1; j<=n; j++)
    vis[j]=0;
 printf("\nEnter number of nodes : ");
 scanf("%d", &n);
 printf("\n Enter adjacency matrix : \n");
  for(i=1; i<=n; i++)
    for(j=0; j<=n; j++)
      scanf("%d", &a[i][j]);
 printf("\n Enter Source node : ");
 scanf("%d", &root);
 ans= dfs(root);
 if(ans==1)
   printf("\n Graph is connected \n");
 else printf("\n Graph not connected ");
```

Output

Enter no of nodes : 4

Enter adjacency matrix :

0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0.

Enter source node : 1

Graph is connected.

(a)
```
#include <stdio.h>
#include <conio.h>

int a[15][15], n;

void bfs (int root){
  int q[15], f=0, r=-1, vis[15], i, j;
    for (j=1; j<=n; j++)
        vis[j]=0;
        vis[root]=1;

    r=r+1;
    q[r]=root;

  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("\n Node %d is not reachable", j);
```

```
else
    printf
}
}
```

Output:

Enter no
Enter a
0 1 1
0 0 0
0 0 0
0 0
Enter s
Node 1
Node 2
Node 3
Node 4

```
else
    printf(" \n Node %d is reachable", j);
}
}
```

Output:

Enter no. of nodes: 4

Enter adjacency matrix:

0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0

Enter source node: 1

Node 1 is reachable
Node 2 is reachable
Node 3 is reachable
Node 4 is reachable