- 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 \le 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);</pre>
```

```
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:

Observation:

```
15-06-23
1 the write the code for the following.
@ Print all the nodes reachable from given Starting needs
   in a dignoph wing Bfs method
@ Check whether a graph is connected or not wing DFS
     onethal.
@ #include < stdio.h>
                                        Charles is connected
   #include Kconio.h}
    int a [10] [10], n, vis [10];
         dfs (int root) f
     thi
         int j:
          (1 = [toot] = 1;
         for (j=1; j <=n; j++)
            if ( a (root)[j]== 1 && vis[j] !=1)
              ds# (j);
          for (J=1; j <=n; j+1) }
                (1=1[[] = 1]
                  return 0;
               else
                   return 1;
       7
  void main() of
    int i, j, root, ans;
    for(j=1; i(=n; j++)
        10=[i] 2iv
    print ( " In Enter number of nools : " );
    Scant (" " ! od " , 6n);
    print+(" in Enter adjacency matrix: "");
     for(i=1). i <= n; i++)
        for (j=0; j <=n; j+t)
           Scant ( " olock", GA(17(37))
     print+ ("in Enter Source node: ");
     scanfl" "lod", broot);
      ans = dfs(root);
      if(ans==1)
      of (ans = = 1)

printf("in Graph is connected (n"))

else printf(" in Graph not connected "))
```

```
Enter 100 of needers: 4
output
 Enter adjacency moutrin:
                                                      prints
                                                    y
0111
  0001
  0 0 00
 0010.
                                                   output:
 Enter source made: 1
                                                   Enter no
 Cyraph is connected.
                                                   Enter
                                                   0 1
@ Hinclude (Shelio. h)
                                                   0 0
  # include (conio. h)
                                                    0 0
  int a (15] [17] n;
                                                    0 0
   void bts (int root) of
                                                   Enter 5
    int 2(15), $=0, Y=-1, vis (15], 1);
                                                   Node 1
      for (5=1; 5k=n; 5++)
                                                   Node 2
         vis (; )=0;
                                                   Mode
                                                   Node 1
         vis Croot Jeri
        Y= 7+1;
        2[7]= root;
      while ( *K= +) }
      "=2(F);
      f=+H',
      Porcs=1; ican; j++)
         it (a (i](i]==1 66 vis (j]: 21) }
           vis[;]=1;
           8=4+1;
           1 5=583 b
  for (j=1; ) (=n) ++) {
      (1=! []) Liv) +;
         print(min Node old is not reachable
```

printt (" In Nucle " od is reachable", i); thrive program to extende topological condening i output: na of nodes: 4 Enter Enter adjacency matrin: 0 1 11 Print Ceiters B redocum rating of their (nd, "bole ") has? 00 0 0 10 0 0 Enter Source noole: 1 minumpersonal to the first time Node 1 is reachable (CANDIDA "bar") took
Node 2 is reachable (CANDIDA "bar") took Node a is reachable. 11 mb (") "> prints 14: (0-20) Wilco

HI THE