ADA
LAB TEST

USN :- 1BM19CS088

NAME :- MODI PARTANYA H.
SEM :- 4-B
DEPT :- CSE

Q 4

(a) Print all the nodes reachable from given starting node in a ~~gp~~ digraph using BFS method

```
#include <stdio.h>
# define size 20
# define true 1
# define false 0
 int queue[size], visit[20], rear = -1, front = 0;
 int n, s, adj[20][20], flag = 0;
 void insertq (int v)
 {
     queue [++rear] = v;

  }
  int deleteq ()
  {    return (queue [front++]);

   }
   int qempty()
   {
      if (rear < front)
      {
          return 1;
       }
       else
       {   return 0;
        }
    }
```

```
void bfs(int v)
{
    int w;
    visit[v]=1;
    insertq(v);
    while(!qempty())
    {
        v=deleteq();
        for (w=1; w<=n; w++)
        {
            if ((adj[v][w]==1)~~&&~~(visit[w]==0))
            {
                visit[w]=1;
                flag=1;
                printf("v %d\t",w);
                insertq(w);
            }
        }
    }
}

void main()
{
    int v,w;
    printf("Enter the no. of vertex : \n");
    scanf("%d",&n);
    printf("Enter the adjacency matrix :\n");
    for (v=1; v<=n; v++)
    {
        for (w=1; w<=n; w++)
        {  scanf("%d",&adj[v][w]);
        }
    }
    printf("Enter the start vertex :");
    scanf("%d",&s);
printf("Reachability of vertex %d \n", s);
for (v=1; v<=n; v++)
{
    visit[v]= 0;
}
bfs(s);
if (flag==0)
{
    printf("No path found!! \n");
}
}
```

(b) Check whether a given graph is connected or not using DFS method

```
#include<stdio.h>
#include<conio.h>
int a[20][20], reach[20], n;

void dfs(int v)
{
    int i;
    reach[v]=1;
    for (i=1; i<=n; i++)
    {   if (a[v][i] && !reach[i])
        {  printf("\n %d -> %d", v, i);
           dfs(i);
        }
} }
```

```
void main()
{
    int i,j,count=0;
    printf("\n Enter the number of vertices:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        reach[i]=0;
        for(j=1;j<=n;j++)
        {
            a[i][j]=0;
        }
    }
    printf("\n Enter the adjacency matrix:\n");

    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    dfs(1);
    printf("\n");
    for(i=1;i<=n;i++)
    {
        if(reach[i])
        count++;
    }
    if(count==n)
        printf("\n Graph is connected");
    else
        printf("\n Graph is not connected");
}
```