WEEK 1

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.

BFS

```
#include<stdio.h>
#include<math.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v)
{
        for(i=1;i<=n;i++)
        if(a[v][i] && !visited[i])
        q[++r]=i;
        if(f<=r)
        {
           visited[q[f]]=1;
           bfs(q[f++]);
        }
}
void main()
{
        int v;
```

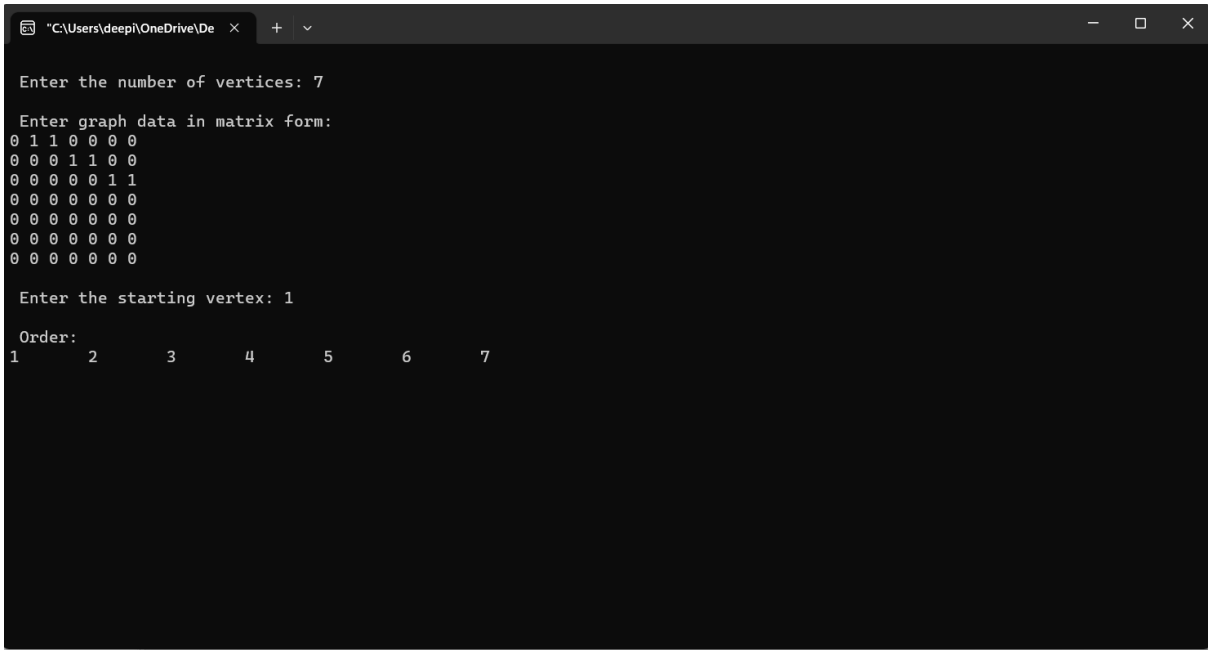
```
        printf("\n Enter the number of vertices:");
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
           q[i]=0;
           visited[i]=0;
        }
        printf("\n Enter graph data in matrix form:\n");
        for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        scanf("%d",&a[i][j]);
        printf("\n Enter the starting vertex:");
        scanf("%d",&v);
        bfs(v);
        printf("\n Order:\n");
         printf("%d\t",v);
        for(i=0;i<=n;i++)
        if(visited[i])
   {

      printf("%d\t",i);

   }

        getch();
}
```

OUTPUT :



DFS

```
#include<stdio.h>
#include<math.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",i);
        dfs(i);
        }
}
void main()
{
        int i,j,v,count=0;
        int f=1;
        printf("\n Enter 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]);
        printf(" %d",f);
        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");
        getch();
}
```

OUTPUT :