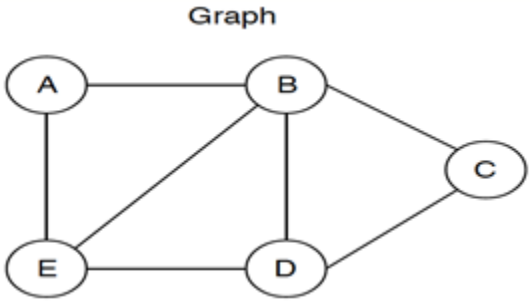
9) Design, Develop and Implement a Program in C for the following operations on Graph (G) of Cities

a. Create a Graph of N cities using Adjacency Matrix.

b. Print all the nodes reachable from a given starting node in a digraph using the DFS / BFS method



DFS:

```
#include<stdio.h>
int stack[10];
int top=-1;
int adj[10][10];
int vis[10]={0};
void main()
{
        int n, s, u, v, i, j;
        int found=0;
        printf("\n Enter the number of vertex:");
        scanf("%d",&n);
        printf("\n Enter the adj matrix:\n");
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        scanf("%d",&adj[i][j]);
                }
        }
        printf("\n Enter the source vertex:");
        scanf("%d",&s);
        stack[++top]=s;
        vis[s]=1;
        printf("source %d:",s);
        while(top!=-1)
        {
                found=0;
                u=stack[top];
                for(v=0;v<n && found==0;v++)
                {
                        if(adj[u][v]==1 && vis[v]==0)
                        {
                                printf("->%d",v);
                                stack[++top]=v;
                                vis[v]=1;
                                found=1;
```

```
                        }
                }
                if(found==0)
                {
                        top--;
                }
        }
}
```

BFS:

```
#include<stdio.h>
int q[10];
int r=-1, f=0;
int adj[10][10];
int vis[10]={0};
void main()
{
        int n, i, j, s, v, u;
        printf("\n Enter the number of vertex:");
        scanf("%d",&n);
        printf("\n Enter the Adj matrix:\n ");
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        scanf("%d",&adj[i][j]);
                }
        }
        printf("\n Enter the source vertex:");
        scanf("%d",&s);
        q[++r]=s;
        vis[s]=1;
        printf("%d: ",s);
        while(f<=r)
        {
                u=q[f++];
                for(v=0;v<n;v++)
                {
                        if(adj[u][v]==1 && vis[v]==0)
                        {
                                        printf("->%d",v);
                                        vis[v]=1;
                                        q[++r]=v;
                        }
                }
        }
}
```