```
/*11.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 DFS/BFS method */

#include <stdio.h>
#include <stdlib.h>

int adj[20][20],queue[20],stk[20],n,i,j,front = 0,rear = -1, top =-1;

void create()
{
    printf("\n Enter the number of vertices(Cities):");
      scanf("%d",&n);
      printf("\n Enter graph data in matrix form:\n");
      for (i = 1;i <= n;i++)
        for (j = 1;j <= n;j++)
         scanf("%d",&adj[i][j]);
}
void bfs(int v)
{
    int reached[20]={0},u;
    queue[++rear] = v;
    while(front<=rear)
    {
        u = queue[front++];
        for(i=1; i<=n; i++)
        {
            if(adj[u][i] == 1 && !reached[i])
            {
                queue[++rear] = i;
                reached[i] = 1;
            }
        }
    }
    printf("\nThe reachable vertices are: \n");
    for(i = 1;i <= n;i++)
    {
        if(reached[i]==1)
            printf("%d ---> %d\n",v,i);
        else
            printf("%d -X-> %d\n", v,i);
    }

}
void dfs(int v)
{
    int visited[20]={0}, count=0, k = 1,u;
     stk[++top] = v;
     visited[v] = 1;
     printf("\n The cities which are reachable are:\n");
    while(top!=-1)
    {
        u = stk[top];
        if(adj[u][k]== 1 && !visited[k])
        {
```

```
                stk[++top] = k;
                visited[k] = 1;
                printf("%d --->%d\n",u,k);
                k = 1;
        }
        else
            k++;

        if(k > n)
        {
            top--;
            k = 1;
        }

    }
                    for(i = 1;i <= n;i++)
                    {
                        if(visited[i])
                        count++;
                    }
                    if(count==n)
                        printf("\n Graph is connected");
                    else
                        printf("\n Graph is not connected");
}

void main()
{
      int v;
      char ch;
    while (1)
    {
        printf("\n\n\n********** Graph(G) of Cities Menu**********\n\n");
           printf("1. Create a Digraph of N cities\n");
           printf("2. Display reachable cities using BFS\n");
           printf("3. Display reachable cities using DFS\n");
           printf("4. Exit\n");
           printf("Enter your choice:\n");
           scanf("%d", &ch);
           switch (ch)
           {
               case 1: create();
                    break;
            case 2: printf("\n Enter the starting vertex:");
                    scanf("%d",&v);
                    bfs(v);
                    printf("\n The cities which are reachable are:\n");
                    break;
            case 3: printf("\n Enter the starting vertex:");
                    scanf("%d",&v);
                    dfs(v);
                    break;
            case 4: exit(0);
            default: printf("Enter a Valid Choice\n");
                    break;
           }
      }
}
```