```
#include<iostream>
using namespace std;

class graph
{
   private:
       int size, edge, graph[20][20];
   public:
       void info()
       {
           cout << "Enter size: \n";
           cin >> size;
           cout << "Enter edges: \n";
           cin >> edge;
           for(int i = 1; i <= size; i++)
           {
               for(int j = 1; j <= size; j++)
               {
                   graph[i][j] = 0;
                }
            }

           for(int k = 1; k <= edge; k++)
           {
               int l, m;
               cout << "Enter start: ";
               cin >> l;
               cout << "Enter end: ";
               cin >> m;
               graph[l][m] = 1;
           }
        }

       void print()
       {
           for(int i = 1; i <= size; i++)
           {
               for(int j = 1; j <= size; j++)
               {
                   cout << graph[i][j] << " ";
                }
               cout << endl;
            }
       }

       int stack[20];
       int index = 0;
       void putstack(int z)
       {
           stack[index] = z;
```

```
    index++;
}
int takestack()
{
    index--;
    return stack[index];
}
void depthTrav(int k)
{
    putstack(k);
    while(index != 0)
    {
        int j = takestack();
        cout << j << " ";
        for(int i = size; i > 0; i--)
        {
            if(graph[j][i] == 1)
            {
                putstack(i);
            }
        }
    }

}

int queue[20];
int rear = 0;
int front = -1;
void queueput(int k)
{
    queue[rear] = k;
    rear++;
}
int queuetake()
{
    front++;
    return queue[front];
}
void breadthTrav(int k)
{
    queueput(k);
    while(rear > front + 1)
    {
        int j = queuetake();
        cout << j << " ";
        for(int i = 1; i <= size; i++)
        {
            if(graph[j][i] == 1)
            {
                queueput(i);
```

```
                   }
               }
           }
       }
};

int main()
{
   graph temp;
   temp.info();
   cout << endl;
   temp.print();
   cout << endl;
   temp.depthTrav(4);
   cout << endl;
   temp.breadthTrav(4);

   return 0;
}

```