```
// Adjacency Matrix Representation of Graph

#include<stdio.h>
#define V 50

struct Graph //Graph as Data Structure
{
		int adj[V][V];
		int e;
		int v;
};


void init(stuct Graph *ptr)	//initialize adj martix to 0
{
		int i,j;
		for(i = 0; i < V; i++)
			for(j = 0; j < V; j++)
				ptr->adj[i][j] = 0;
}

//Add an edge.
void addEdge(struct Graph *ptr,int src, int dest) //set adj[src][dest] = 1
{
		ptr->adj[src][dest] = 1;
}



void printAdjMatrix(struct Graph gh)  //print the adjMatrix
{
		int i, j;

		for(i = 0; i < gh.v; i++)
		{
			for(j = 0; j < gh.v; j++)
			{
				printf("%d\t", gh.adj[i][j]);
			 }
			printf("\n");
		}
}


int main()
{
        struct Graph g; //create a graph

        printf("Enter the number of vertices :");
        scanf("%d",&g.v);
        printf("Enter the number of edges :");
        scanf("%d",&g.e);

        init(&g);  //initialize adj matrix

        for(i=1;i<=g.e;i++) //add edge
        {
                printf("\nEnter the source node value :");
                scanf("%d",&s);

                printf("\nEnter the destination node value :");
                scanf("%d",&d);

                addEdge(&g,s,d);
        }

        printAdjMatrix(g); //print graph

        return 0;
}
```