```c
#include <stdio.h>

#define max 100 // số cung toi da

// cau truc cua mot cung
typedef struct{
    int x,y; // dinh dau dinh cuoi cua mot cung
}Edge;

// cau truc cua do thi G
typedef struct{
    int n,m; // so dinh vaf so cung cua do thi
    Edge edges[max];
}Graph;

// ham khoi tao do thi
void init_Graph(Graph *G, int n){
    G->n = n; // gan so dinh cho do thi
    G->m = 0; // gan so cung cua do thi = 0
}

// ham them cung cho do thi
void add_Edge(Graph *G, int x, int y){
    G->edges[G->m].x = x; // gan gia tri dinh dau  = x
    G->edges[G->m].y = y; // gan gia tri dimh cuoi  = y
    G->m++;
}

// kiem tra dinh ke
int adjacent(Graph *G, int x, int y){
    int e;
    for(e = 0; e<G->m; e++) // duyet qua tung cung
        // dinh dau = x va dinh cuoi  =y or nguoc lai
        if((G->edges[e].x == x &&  G->edges[e].y == y) || (G->edges[e].y == x &&
        G->edges[e].x == y) )
            return 1; // tra ve 1
    return 0; // nguocwj lai = 0
}

// xac dinh bac cua mot dinh
int degree(Graph *G, int x){
    int e, deg = 0;
    for(e = 0; e<G->m;e++){ // duyet tung cung
        if(G->edges[e].x == x) deg++; // neu x = dau tang bac
        if(G->edges[e].y == x) deg++; // neu x = cuoi tang bac
    }
    return deg;
}

void neighbours(Graph *G, int u){
    int v;
    for(v = 1; v<= G->n; v++)
        if(adjacent(G,u,v) != 0)
        printf("%d ", v);
        printf("\n");
}
int main(){

    Graph G;
    int x,y,e;
//  FILE *f = fopen("dothi.txt", "r");
//  fscanf(f, "%d%d", &x, &y);
//  init_Graph(&G, x);
//  int u, v;
//  for(e = 1; e<= y; e++){
//      fscanf(f, "%d%d", &u,&v);
//      add_Edge(&G, u, v);
//  }
    int u, v;
    freopen("dothi1.txt", "r", stdin);
    scanf("%d%d", &x,&y);
    init_Graph(&G, x);
    for(e = 1; e<=y; e++)
```

```
    {
        scanf("%d%d", &u, &v);
        add_Edge(&G, u,v);
    }
    int i,j;
    for(i = 1; i <= G.n; i++)
    printf("Degree(%d): %d\n", i, degree(&G, i));
    printf("------------------------------\n");
    int a;
    for(a = 1; a <= G.n; a++)
    for(j = 1; j <=G.n; j++){
        if(adjacent(&G, a, j) == 1){
                printf("%d ke %d\n", a, j);
        }
        else {
        printf("%d khong ke %d\n", a, j);
        }
    }

    printf("-------------------------\n");

        for(i = 1; i <= G.n; i++){
        printf("Neighbours cua %d la: ", i);
        neighbours(&G, i);
        printf("\n");
    }
    return 0;


}

```