

```

1  #include <stdio.h>
2
3  #define max 100 // sos cung toi da
4
5  // cau truc cua do thi G
6  typedef struct{
7      int n,m; // so dinh vaf so cung cua do thi
8      int A[max][max];
9  }Graph;
10
11 // ham khoi tao do thi
12 void init_Graph(Graph *G, int n){
13     G->n = n; // gan so dinh cho do thi
14     G->m = 0; // gan so cung cua do thi = 0
15     int u,v;
16     for(u = 1; u <= n; u++)
17         for(v = 1; v <= n; v++)
18             G->A[u][v] = 0;
19 }
20
21 // ham them cung cho do thi
22 void add_Edge(Graph *G, int x, int y){
23     G->A[x][y] = 1;
24     G->A[y][x] = 1;
25     G->m++;
26 }
27
28 // kiem tra dinh ke
29 int adjacent(Graph *G, int x, int y){
30     return G->A[x][y] > 0;
31 }
32
33 // xac dinh bac cua mot dinh
34 int degree(Graph *G, int x){
35     int e, deg = 0;
36     for(e = 1; e <= G->n; e++){ // duyett tung cung
37         deg += G->A[x][e];
38     }
39     return deg + G->A[x][x];
40 }
41
42 void neighbours(Graph *G, int u){
43     int v;
44     for(v = 1; v <= G->n; v++){
45         if(G->A[u][v] != 0)
46             printf("%d ", v);
47         printf("\n");
48     }
49 }
50
51 int main(){
52     Graph G;
53     int x,y,e;
54     // FILE *f = fopen("dothi.txt", "r");
55     // fscanf(f, "%d%d", &x, &y);
56     // init_Graph(&G, x);
57     // int u, v;
58     // for(e = 1; e <= y; e++){
59     //     fscanf(f, "%d%d", &u,&v);
60     //     add_Edge(&G, u, v);
61     // }
62     int u, v;
63     freopen("dothi1.txt", "r", stdin);
64     scanf("%d%d", &x,&y);
65     init_Graph(&G, x);
66     for(e = 1; e <= y; e++)
67     {
68         scanf("%d%d", &u, &v);
69         add_Edge(&G, u,v);
70     }

```

```

70 // init_Graph(&G, 4);
71 // add_Edge(&G, 1,2);
72 // add_Edge(&G, 2,3);
73 // add_Edge(&G, 3,4);
74 // add_Edge(&G, 4,1);
75 // add_Edge(&G, 2,4);
76 int i,j;
77 for(i = 1; i <= G.n; i++)
78 printf("Degree(%d): %d\n", i, degree(&G, i));
79 printf("-----\n");
80 int a;
81 for(a = 1; a <= G.n; a++)
82 for(j = 1; j <=G.n; j++){
83     if(adjacent(&G, a, j) == 1){
84         printf("%d ke %d\n", a, j);
85     }
86     else {
87         printf("%d khong ke %d\n", a, j);
88     }
89 }
90
91 printf("-----\n");
92 for(i = 1; i <= G.n; i++){
93     printf("Neighbours cua %d la: ", i);
94     neighbours(&G, i);
95     printf("\n");
96 }
97 return 0;
98
99
100 }
101

```