```
CODE 1
   # include < Stdio. 47
   A clefine max 100 // Số cung rối đã
    1/ can truc eua mot cong
  typedel STRUCT &
   } Edge;
   Il can touc cua để thi G.
  typedeg struct }
      int n,m
      Edge edges [max];
 } Graph ;
 Il ham What tow do the
 10iel invir_Graph (Graph & inth) {
G -> m=n; 1/gan so' cung cho fr=0
}.
   Il haim them cang the off This.
void add _ Edge (Graph & / Int x, int y) {

G -> edges [G->m] x = x; // Gan gia mi (sinh dan = x.

G -> edges [G->m], y=y, // gan gia Tri dinh cuòi = y.
 " widn' rra dinh trèn ce
  ad Tacena ( Graph & G, Intx, inty) }
       C+ -> odges
 int
   ( e=0; e<6+m; e++) // duyét qua rung cung.

// dinh dan = x din's cuit = y or ngilic (aci
```

```
ig ((G) edge[e]. x == x &8 G -> edges[e]. y == y)
         11 ( to -) edges [e]. y = = x 88 G-) edges [e]. x = = y)
         return 1; // tra' ve 1
return 0; // mopiet lou tra ve 0.
11 par d'ins baix cua mot d'ins.
 int clegree (Graph &, int oc) {
       int e, chegio;
   for (e=0; e < G-) m ; e ++) { // oluyer tring cung.
           if (G-) edges [e]. X = = x) deg ++; 1/ mên x = adn Tio
if (G-) edges [e]. y = = x) deg ++; 1/ nên x = cuối try bà
       retium deg;
Void neighbours (Graph Of , in a u) {
  gor ( 1=1; V (= G-)n; (1++) {
```

if (ad Jacons (G, U, U) 1=0)

prinag ("%d", V);

printy ("\n");

```
CODE 2.
# include < stdio.h>
# define may 100
# define Max_E 100
// clayer to this theo dang Binh - 19inh.
supeded smuts
  int A [MAY-E][MAY-E];
  11 Tạo Sơ đố định định.
& Graph;
void init_ graph (Graph G, int n) {
  ina 1,T;
 G-)n=n;
  for (i = 1; i < z  G - n; i + t)  for (j = 1; j < z  G - n; j + t) { G -> A[i][j] = 0; }
    Ichdi tạo đố thị Đinh cung
void init-graph 1 (Graph &G/inIn, int m) {
    ina visi
   G-) n:n
  g-1 m=m (i = q = nh; i++) {
    for (i=1; i = q = nh; i++) {
        for (j=1; j = q -nh; j++) {
            G-)A[i][j]=0;
        }
                     vai de thi dinh-dinh
```

```
11 theo cung vai of the clin's - olin's void adel edge ( Graph & Grint x, int y) {
     G-) A[X][y]=2;
 Il them cung voio đổ thị định - Hinh cung
 Void add edge 1 (Graph & G, int x, inty) {
     G -> A[20][2]= 1 ;
 1/ likim tra có phai lang gióng.
int adjacent (Graph &G, intra, inty) {
     return G >> A[rc] [y] == 1;
Tinh sc'bac cua met ofins
int degree (Graph G, intil , inty) {
     int oleg = 0;
    for (i=1; i2= G-)n; i++) {
    ig (G-) ACUJ[i] = = 2) {
          R cleg ++;
     return deg ;
Win do the dinh dinh .
    for (i=2; i < G.n; T++) }
       gor (Tilite (+.n ; J++) {
          prints ( ingd , G.ALi][j];
      3 Frints ("(n");
```

```
Cope 3.
  11 Include (STOlich)
           max 100 11 So cung Til ora.
   11 Ceers
sypedef STAUCT &
                 11 se din's và cung qua orc phi
    int A[max][max]
11 ham Cher Too of This.
Word init_ Graph (Graph &G, int n) {
   Connin // gam St' Tinh cho Go the
   (I-) m=0; 11 gan So cung the do thi =0
    ins μ, ν;
for (u=1; u = n; u++)
    for (V=1; V= h; V++)
    G-)A[U][v] =0;
ham then cung the at this
sid add - Edge [Graph &G, in 7 x, in xy ] }
G-) A[][[]=2/
G-> ALY][]=2,
 G-) m+t
Il Kiem tra dinh Ice
  adjacent (Graph &G, int 72, inty) {
  Verurn G -) A(20)(4) 7 G
I xaé d'inh bai cua mêt d'inh.
no degree (Graph &G, in T 2) }
  for (e=1; e(=6-)n(e++) { // duyer rung cung.
cleg += G > AN[e];
```

Được quét bằng CamScanner

```
return deg + G > A [n] [n];
        neighbours (Graph &G, int u) {
   ins V
  gor ( V=1; V <= G→n; V++)
   18(9-) A[1][1] != 0)
    Print ("% of ", V);
   printf (( \h")
 Il in of the dinh dinh.
void in (Graph G) }
  int i, T
  for (i=1, ic= n; i++) {
    gor (T=1; T C= G.n; J++) {
    prints ( &d" , G. A[i][J] );
     Pring ("\n");
11 in do on Thi dinh - cuny.
void ind (Graph G) {
ins i, Ji
 for (i=1; i cm (J.h; i+1) {
    for (J=1, J \ (r.n , J+1) {
        printg (%d ", G. A \ () [J]);
    }
     prints ("\n");
```