Problem A

```
Party



Suppose we have a party of n people. We know if any two of them are acquainted with each
other. We would like to know more about triples of people that are mutual acquaintances.
Please calculate:
        t  - the number of different triples of people that are mutual
             acquaintances,
        m - the maximum number of triples that a person belongs to, and
        k  - the number of persons that belong to exactly m triples.

Input
The first line of the input contains n - the number of persons. In the lines that follow there are
pairs of numbers i,j denoting that two people with these numbers on the list of participants are
acquainted with each other. The end of the input is marked by the pair 0 0.

Output
The output consists of three numbers: t, m and k.

EXAMPLE

Input:

8
1 2  2 3  2 5  2 6  2 7  2 8  3 4  3 5  3 6  3 7
3 8  5 6  6 8  7 8
0 0

Output:

t = 10
m =  7
k =  2
```

Solution

```
n people are represented by n vertices of a simple
undirected graph. There is an edge between two
vertices if corresponding people know each other.
The adjacency matrix A is used for representing the
graph, and L, the vector of n integers, is used for
counting the number of triangles that a vertex
belongs to. The procedure initially sets counter L and
the number of triangles t to zero. Then it checks for
each edge whether both vertices have a common
neighbor. If yes, then t is increased by one and L is
increased by one for these three vertices. Next m, the
maximum value in L, is determined and finally, the
number of vertices k that belong to exactly m
triangles is calculated.
```

```
Tests

TEST 1

input           8
1 2  2 3  2 5  2 6  2 7  2 8  3 4  3 5  3 6  3 7
3 8  5 6  6 8  7 8
0 0

output  t = 10,  m =  7,  k =  2

TEST 2

input           9
1 2  1 3  1 4  1 7  2 3  2 5  2 8  3 6  3 9  4 5
4 6  4 7  5 6  5 8  6 9  7 8  7 9  8 9
0 0

output  t =  6,  m =  2,  k =  9

TEST 3

input           9
1 2  1 3  1 4  1 6  1 7  2 3  2 4  3 4  3 5  3 6
3 7  3 9  4 5  4 6  4 7  4 8  5 6  6 7
0 0

output  t = 16,  m = 10,  k =  2

TEST 4

input           9
1 2  1 3  1 4  1 7  2 3  2 4  2 5  3 4  3 5  3 6
3 7  3 9  4 5  4 6  4 7  4 8  5 6  6 7
0 0

output  t = 15,  m = 10,  k =  2


TEST 5

input           5
1 2  1 3  1 4  1 5  2 3  2 4  2 5  3 4  3 5  4 5
0 0

output  t = 10,  m =  6,  k =  5
```

Listing

```
     program party(input, output); { KTB, 1996 }
     const nmax=100;
     type ind=1..nmax;
          t1=array[ind] of integer;
          t2=array[ind,ind] of Boolean;
          alfa=string[15];

     var A:t2; n,tr,m,k:integer;
         key:integer; dat,out:text; devd, dev:alfa;

     procedure czyt(var n:integer; var A:t2);
     var i,j:integer;
     begin
       write('input file:'); readln(devd);
       assign(dat, devd); reset(dat);
       readln(dat, n);
       for i:=1 to n do for j:=1 to n do A[i,j]:=false;
       repeat
```

```
    read(dat, i, j);
    if (i > 0) and (i <= n) and (j > 0) and (j <=n) then
      begin A[i,j]:=true; A[j,i]:=true end;
    until (i = 0) or (j = 0);
  close(dat)
end; { czyt }

function max(n:integer; var a:t1):integer;
var i,x:integer;
begin
  x:=a[1];
  for i:=2 to n do if a[i] > x then x:=a[i];
  max:=x;
end; { max }

procedure triangles(n:integer; var A:t2;
                                   var tr,m,k:integer);
var i,j,h:integer; x:integer; L:t1;
begin
  for i:=1 to n do L[i]:=0;
  x:=0;
  for i:=1 to n - 2 do
    for j:=i + 1 to n do
      if A[i,j] then
        for h:=j + 1 to n do if A[i,h] and A[j,h] then
          begin
            x:=x + 1;
            L[i]:=L[i] + 1; L[j]:=L[j] + 1; L[h]:=L[h] + 1
          end;
 { write('L: '); for i:=1 to n do write(L[i]);}
  tr:=x;
  m:=max(n, L);
  k:=0; for i:=1 to n do if L[i] = m then k:=k + 1
end; { triangles }

procedure druk(n,tr,m,k:integer; var A:t2);
var i,j,h:integer;
begin
  write('output file:'); readln(dev);
  assign(out, dev); rewrite(out);
  writeln(out);
{  writeln(out, 'n =', n:3);
  writeln(out, 'Edges:');
  h:=0;
  for i:=1 to n - 1 do
    for j:=i + 1 to n do
      if A[i,j] then
        begin
          h:=h + 1; write(out, i:4, j:3);
          if h mod 10 = 0 then writeln(out)
        end;}
  writeln(out);
  writeln(out, 't =', tr:3, ',  m =', m:3, ',  k =', k:3);
  writeln(out);
  close(out);
end; { druk }

begin
  repeat
    czyt(n, A);
    triangles(n, A, tr, m, k);
    druk(n, tr, m , k, A);
    writeln('end? 0/1'); readln(key);
  until key=1;
end.
```