```
unit txt400;
  interface
    uses   MemTypes, QuickDraw, OSIntf, ToolIntf, PackIntf,
PrintTraps, PasLibIntf,
      globals;

    procedure text400(i,full:longint);

implementation

procedure text350(i,full:longint);
   begin
     case i of
 301: if full = 1 then write('if cubic ...more...')
      else
       begin
       writeln('if cubic, nconn > 1, planar, not Hamiltonian, and
bipartite');
       if full <> 0 then write('         ');
       write('                 then nodes >= 26');
     end;
302: write('if cubic, nconn > 1, planar, not Hamiltonian then
nodes >= 14');
303: write('if cubic, nconn > 1, not Hamiltonian, and bipartite
then nodes >= 20 ') ;
304: if full = 1 then write('if regular and econn >= mindeg-2 >=
1 then ...more...')
    else
      begin
        writeln('if regular and econn >= mindeg-2 >= 1 then');
        if full <> 0 then write('         ');
        writeln('           eind >= nodes/2-FL[(nodes+1)/(2*x)],
when nodes even, else');
        if full <> 0 then write('         ');
        writeln('                >= (nodes-MAX{2*FL[(nodes+1
+x)/(2*x)]-1, 1})/2');
        if full <> 0 then write('         ');
        write('                  where x = mindeg*FL[(mindeg+
3)/2]-1');
      end;
305: write('if cubic then eind >= nodes/2-FL[(nodes+3)/18]-
FL[(ncomp+4)/6]');
306: write('if clique = 2 and maxdeg <= 4 then edges >= 6
*nodes-13*nind');
307: if full = 1 then write('if econn > 0 then econn >=
MIN{mindeg, ...more...')
    else
      begin
        writeln('if econn > 0 then econn >= MIN{mindeg,
nodes*(maxdeg-2)/x}');
```

```
        if full <> 0 then write('         ');
        write('               where x =
(maxdeg-1)**(diam-1)+maxdeg*(maxdeg-2)-1');
     end;
308: write('if nconn >= 2 and nind >= 2 then circ >= 2
*(nodes-2)/nind+2');
309: write('if cubic, nconn = 3, and planar then circ >=
MIN[nodes, 17]');
310: write('if cubic, nconn = 3, planar, and nodes <= 36 then
Hamiltonian');
311: if full = 1 then write('if clique < chr = maxdeg
then ...more...')
      else
       begin
         writeln('if clique < chr = maxdeg then ');
         if full <> 0 then write('         ');
         write('             nodes >= 2*maxdeg  {-1, if maxdeg <=
8}');
       end;
312: write('if cubic and econn >= 2 then eind = nodes/2');
313: write('if regular and nconn >= 3 then circ >= MIN[nodes, 3
*mindeg]');
314: if full = 1 then write('if regular ...more...')
      else
       begin
       writeln('if regular, nconn >= 2, and nodes <= 3*mindeg+
3');
       if full <> 0 then write('         ');
       write('                 then circ >= MIN[nodes, 3
*mindeg]');
     end;
315: if full = 1 then write('if regular and nconn >= 2
then ...more...')
      else
       begin
       writeln('if regular and nconn >= 2 then');
       if full <> 0 then write('         ');
       write('             circ >= MIN[nodes, 3*mindeg, 2*mindeg+
4]');
     end;
316: write('if regular, nodes even, maxdeg >= 6*nodes/7 then echr
= maxdeg');
317: if full = 1 then write('if maxdeg = nodes-1 ...more...')
      else
       begin
       writeln('if maxdeg = nodes-1 and ');
       if full <> 0 then write('         ');
       write('             ');
       writeln('edges <= 2*FL[(nodes-1)/2]**2  {+mindeg, if nodes
even}');
       if full <> 0 then write('         ');
       write('                then echr = maxdeg');
     end;
318: if full = 1 then write('if spectr > ...more...')
      else
       begin
       writeln('if spectr > SQRT[maxdeg]*(nodes*Ck/2)**(1/(2*k))
then ');
       if full <> 0 then write('         ');
       write('       ');
       write('        girth <= 2*k+1,    where Ck is the k\th
Catalan number.');
     end;
319: if full = 1 then write('if spectr >= ...more... then girth
<= 4')
      else
       begin
       writeln('if spectr >= (2*edges(2*maxdeg-1)-2*s*(maxdeg-
s))**(1/4)');
       if full <> 0 then write('         ');
       write('            ');
       writeln('where 2*edges = q*maxdeg+s, 0 <= s < maxdeg,');
       if full <> 0 then write('         ');
       write('            then girth <= 4');
     end;
320: write('if connct, regular, nodes odd and < 5*mindeg/2 then
girth = 3');
321: if full = 1 then write('if connct, regular ...more...')
      else
       begin
       writeln('if connct, regular, and not complete then
eccov');
       if full <> 0 then write('         ');
       writeln('              >= 3*nodes/4,
if mindeg <=  3');
       if full <> 0 then write('         ');
       write('              >=
mindeg*nodes/((mindeg-1)*(mindeg-2)), if mindeg >= 5');
     end;
322: if full = 1 then write('if connct, regular ...more...')
      else
       begin
       writeln('if connct, regular, mindeg <= 4 and not complete
then');
       if full <> 0 then write('         ');
       writeln('           eccov >= 3*nodes/5');
       if full <> 0 then write('         ');
       write('                   {+1, if nodes = 7, 3 MOD 5, but
<> 13 or 18}');
     end;
323: if full = 1 then write('if girth >= 6 then
nind >= ...more...')
       else
         begin
```

```
           writeln('if girth >= 6 then nind >= ');
           if full <> 0 then write('         ');
           write('              nodes*(2*maxdeg-1)/(maxdeg**2+2
*maxdeg-1)');
         end;
324: if full = 1 then write('if cubic then nind >= ...more...')
      else
       begin
       writeln('if cubic then nind >= 19*nodes/52, if girth >=
6');
       if full <> 0 then write('         ');
       write('                            >= 20*nodes/53, if
girth >= 8');
     end;
325: if full = 1 then write('if regular, ...more...')
      else
       begin
       writeln('if regular, girth even and >= 6, connct, and
nodes <=');
       if full <> 0 then write('         ');
       write('            ');
       writeln(' (mindeg*(mindeg-3)+2
*(mindeg-1)**(girth/2))/(mindeg-2)');
       if full <> 0 then write('         ');
       write('                 then bipartite and diam = girth/2+
1');
     end;
326: if full = 1 then write('if bipartite and nodes odd
then ...more...')
       else
         begin
           writeln('if bipartite and nodes odd then');
           if full <> 0 then write('         ');
           write('              thick <= CL[(nodes**2-1)/(8
*nodes-16)]');
         end;
327: write('if edges > 0 then maxdeg >= 2*thick-1');
328: write('nconn <= 6*thick-1');
329: write('if cubic and girth = 10 then nodes >= 70');
330: if full = 1 then write('if edges >= MAX[...more...] then
Hamiltonian')
   else
    begin
      writeln('if edges >= 1+MAX[(nodes-mindeg)*(nodes-
mindeg-1)/2+mindeg**2,');
      if full <> 0 then write('         ');
      writeln('           FL[(nodes+
2)/2]*(FL[nodes/2]/2-1)+FL[(nodes-1)/2]**2]');
      if full <> 0 then write('         ');
      write('                  then Hamiltonian');
    end;
331: write('edges <= nodes*(nodes-1)/2-(mindeg-nconn+1)*(nodes-
mindeg-1)');
332: write('if tree and maxdeg <= nodes-2 then bwidth <=
(nodes-1)/2');
333: write('edges >= 2*bwidth-1');
334: if full = 1 then write('if girth >= 5, mindeg >= 3 and
connct then dom <= ...more...')
     else
      begin
        writeln('if girth >= 5, mindeg >= 3, and connct then dom
<=');
        if full <> 0 then write('         ');
        write('            (nodes-1-FL[diam/3]*(mindeg-1)-
(mindeg-1)*(mindeg-2)/2)/2');
        if full <> 0 then write('         ');
      end;
335: if full = 1 then write('if nodes/2 <= bwidth < nodes
then ...more...')
      else
       begin
       writeln('if nodes/2 <= bwidth then edges >=');
       if full <> 0 then write('         ');
       write('            ');
       write(' (2*FL[nodes/2]-1)*(nodes/(nodes-2))**(bwidth-
FL[nodes/2])');
      end;
336: if full = 1 then write('if bwidth >= nodes/2
then ...more...')
       else
        begin
          writeln('if bwidth >= nodes/2 then');
          if full <> 0 then write('         ');
          write('              edges >= nodes*(nodes-1)/(2
*nodes-2*bwidth)');
        end;
337: if full = 1 then write('if girth >= 5 then dom
<= ...more...')
       else
         begin
           writeln('if girth >= 5 then dom <=');
           if full <> 0 then write('         ');
           write('            nodes/2-(mindeg-1)*(4*edges/(nodes-
mindeg-2))/4');
         end;
338: if full = 1 then write('if connected, girth >= 5
and ...more...')
      else
       begin
       writeln('if connected, girth >= 5 and mindeg >= 4 then');
       if full <> 0 then write('         ');
       write('             dom <=(nodes-maxdeg-
mindeg*(mindeg-3)/2)/2');
     end;
```

```
339: write('if girth >= 5 then dom >= mindeg*ncomp');
340: write('if girth >= 6 then dom >= 2*mindeg-2');
341: write('if mindeg >= 2 and girth >= 7 then dom >= maxdeg+1');
342: if full = 1 then write('if 5 <= girth <= nodes/2 then
edges...more...')
      else
       begin
         writeln('if 5 <= girth <= nodes/2 then');
         if full <> 0 then write('         ');
         write('              edges <= (nodes**2)/girth-nodes+
2');
       end;
343: write('if girth >= 5 then edges <= nodes*SQRT[nodes-1]/2');
344: if full = 1 then write('if not a forest and nodes >= (3
*girth-2)/2 then ...more...')
      else
       begin
         writeln('if not a forest and nodes >= (3*girth-2)/2
then');
         if full <> 0 then write('         ');
         write('             edges <= nodes*(nodes-1)/FL[(3
*girth-5)/2]-ncomp+1');
        end;
345: if full = 1 then write('if not a forest thenedges <=
nodes*m**(1/t)+nodes-1...more...')
    else
      begin
        writeln('if not a forest then edges <=
nodes*m**(1/t)+nodes-1');
        if full <> 0 then write('         ');
        writeln('                           , where t =
FL[(girth-1)/2] >= 2 and');
        if full <> 0 then write('         ');
        write('                                   m = MAX[1,
(nodes-2**(t+1)+8)/4]');
      end;
346: if full = 1 then write('if not a forest...more...')
      else
       begin
       writeln('if not a forest and (nconn > 0 or mindeg > 1)
then');
       if full <> 0 then write('         ');
       write('              genus >= edges*(1-2/girth-2/mindeg)/2
+ncomp');
     end;
347: write('if diam=2 then nodes <= nconn*maxdeg+1');
348: if full = 1 then write('if not a forest and edges >= nodes+
2-ncomp then...more..')
 else
   begin
     writeln('if not a forest and edges >= nodes+2-ncomp then ');
     if full <> 0 then write('         ');
     write('             nodes >= 3*girth/2-3+2*ncomp');
   end;
349: if full = 1 then write('if not a forest and edges >= nodes+
3-ncomp then...more...')
  else
    begin
      writeln('if not a forest and edges >= nodes+3-ncomp then');
      if full <> 0 then write('         ');
      write('            nodes >= 2*girth-4+2*ncomp');
    end;
350: if full = 1 then write('if not a forest and edges >= nodes+
4-ncomp then...more..')
  else
    begin
      writeln('if not a forest and edges >= nodes+4-ncomp then
');
      if full <> 0 then write('         ');
      write('            nodes >= 9*girth/4-5+2*ncomp');
    end;
     end;
     end;

procedure text400(i,full:longint);
   begin
     if i <= 350 then text350(i,full)
         else
     case i of
351: if full = 1 then write('if girth >= MAX[5, (nodes+1)/2]
and ...more...')
      else
        begin
          writeln('if girth >= MAX[5, (nodes+1)/2] and edges >=
nodes+3');
          if full <> 0 then write('         ');
          writeln('             then girth <= 8 { <> 7}, nodes =
2*girth-1, nonplanar,');
          if full <> 0 then write('         ');
          write('                  nconn = econn = mindeg = 2,
and edges = nodes+3');
        end;
352: if full = 1 then write('if girth >= 5 then
edges ...more...')
       else
        begin
         writeln('if girth >= 5 then ');
         if full <> 0 then write('         ');
         writeln('             edges <= nodes*(.5+SQRT[(nodes-
mindeg-1)/S+.25])/2');
         if full <> 0 then write('         ');
         writeln('               where t = FL[(girth-1)/2] and S
= t-1, if mindeg <= 2,');
         if full <> 0 then write('         ');
         write('                 else S =
((mindeg-1)**(t-1)-1)/(mindeg-2)');
       end;
353: if full = 1 then write('if girth >= 5 then ...more...')
      else
        begin
          writeln('if girth >= 5 then ');
          if full <> 0 then write('         ');
          writeln('             edges <= nodes*(.5+sqrt[(nodes-
k-2)/R+.25])/2');
          if full <> 0 then write('         ');
          writeln('               where t = FL[(girth-1)/2], k =
FL[edges/nodes]');
          if full <> 0 then write('         ');
          writeln('                and R = t-1, if k <= 1, else
');
          if full <> 0 then write('         ');
          write('                    R = (k**(t-1)-1)/(k-1)');
        end;
354: if full = 1 then write('if not a forest then edges <=
nodes+...more...')
      else
        begin
          writeln('if not a forest then edges <=');
          if full <> 0 then write('         ');
          writeln('             nodes+k*(2*nodes-girth*(k+1))/(2
*FL[(girth-1)/4])');
          if full <> 0 then write('         ');
          write('                 where k = FL[(nodes-
FL[(girth-1)/2])/girth]');
        end;
355: if full = 1 then write('nodes >= 1+maxdeg*S ...more...')
       else
         begin
           writeln('nodes >= 1+maxdeg*S {-
(maxdeg-1)*(mindeg-1)**(t-1), if girth even}');
           if full <> 0 then write('         ');
           writeln('                  where t = FL[girth/2],');
           if full <> 0 then write('         ');
           writeln('                     S = 1, if mindeg = 1,');
           if full <> 0 then write('         ');
           writeln('                       = t, if mindeg = 2,
and');
           if full <> 0 then write('         ');
           write('                        =
((mindeg-1)**t-1)/(mindeg-2), if mindeg > 2');
         end;
356: write('if regular and eind < nodes/2 then echr = maxdeg+1');
357: write('if regular then eind >= nodes*maxdeg/(2*maxdeg+2)');
358: write('if nconn >= 2 and 3*mindeg >= nodes+nconn then
Hamiltonian');
359: write('if nconn >= 3 then circ >= MIN{nodes, 3*mindeg-
nconn]');
360: write('if regular and nodes = 2maxdeg+1 then nconn >=
nind');
361: if full = 1 then write('if girth >= 4 and mindeg = 2
then ...more...')
      else
        begin
          writeln('if girth >= 4 and mindeg = 2 then');
          if full <> 0 then write('         ');
          writeln('              nind >= maxdeg*FL[(t+1)/2] {+1,
if t even}');
          if full <> 0 then write('         ');
          write('                      where t =
FL[(girth-2)/2]');
        end;
362: if full = 1 then write('if girth >= 4 and mindeg >= 3
then...more...')
      else
        begin
          writeln('if girth >= 4 and mindeg >= 3 then  nind >=
maxdeg*s+k');
          if full <> 0 then write('         ');
          writeln('             where t = FL[(girth-2)/2] and');
          if full <> 0 then write('         ');
          writeln('              s =
(mindeg-1)*((mindeg-1)**t-1)/(mindeg*(mindeg-2))');
          if full <> 0 then write('         ');
          write('              k = 1 or CL[maxdeg/mindeg], when t
is odd or even, resp.');
        end;
363: if full = 1 then write('if diam=2 and nconn >= 3 then
edges >= ...more..')
       else
        begin
          writeln('if diam=2 and nconn >= 3 then edges >=');
          if full <> 0 then write('         ');
          write('            (nodes-1)*(k+1)/2-k**2+2*k,     k =
nconn');
        end;
364: if full = 1 then write('if diam=2, nconn >= 3 and nodes >= 3
*nconn**2+6..')
       else
         begin
           writeln('if diam=2, nconn >= 3, and nodes >= 3*nconn**
2+6 then');
           if full <> 0 then write('         ');
           write('               edges >= (nodes-1)*(nconn+
1)/2');
         end;
365: if full = 1 then write('if bipart then xnum <= ...more...')
       else
         begin
```

```
           writeln('if bipart then xnum <= (nodes div 4)**2
*((nodes-2) div 4)**2');
           if full <> 0 then write('         ');
           writeln('            when nodes even, and when odd');
           if full <> 0 then write('         ');
           write('            xnum <= ((nodes+1) div
4)*((nodes-1) div 4)**2*(nodes-3) div 4)');
         end;
366: write('if connct then spectr >= 2*cos(PI/(nodes+1))');
367: if full = 1 then write('if reg, mindeg >= 7 and odd, not
bipart, and girth =4 then...')
    else
      begin
        writeln('if reg, mindeg >= 7(<> 9) and odd, not bipart,
and girth = 4 then');
        if full <> 0 then write('         ');
        write('               nodes >= 2*CL((5*mindeg+5)/4)');
      end;
 368: write('if bipartite then thick <= nodes/8+2');
 369: write('if clique <= 2 then thick <= genus+1');
 370: write('if genus <= 1 then thick = genus+1');
 371: write('arbor <= earbor');
 372: write('thick <= earbor');
 373: write('if genus > 0 then earbor <= 2+CL(sqrt(3*genus))');
 374: write('thick <= 5+sqrt(2*genus-2)');
 375: write('if conncted and regular, earbor <= 4+(6*genus+
2)/(nodes-1)');
 376: write('genus <= (thick-1)*(nodes-1)');
 377: write('earbor <= (maxdeg+2)/2');
 378: write('earbor >= (mindeg+1)/2');
 379: write('earbor >= edges/(nodes-ncomp)');
 380: write('earbor <= 3*thick');
 381: write('if planar and nodes >= 4 then edges <= 3*nodes-9
+min{3,econn}');
 382: if full = 1 then write('if planar and econn < mindeg then
edges <=...more...')
    else
      begin
        writeln('if planar and econn < mindeg and (nodes >= 5 or
mindeg >= 2) then');
        if full <> 0 then write('         ');
        writeln('              edges <= 3*nodes-11,
when mindeg = econn+1 = 1');
        if full <> 0 then write('         ');
        write('                       3*nodes-12+econn,
otherwise.');
      end;
  383: if full = 1 then write('if not a forest then nodes >=
maxdeg+ncomp...more...')
     else
       begin
         writeln('if not a forest then nodes >= maxdeg+ncomp-2
+');
         if full <> 0 then write('         ');
         write('               (circ*(girth-3)+2)/(girth-2)');
       end;
  384: write('ncov <= (2*nodes+edges-eind)/4');
  385: if full = 1 then write('if genus <= (nodes*(sqrt(2
*nodes)-7))/12+1 then...more...')
     else
       begin
         writeln('if genus <= (nodes*(sqrt(2*nodes)-7))/12+1
then');
         if full <> 0 then write('         ');
         write('              eccov <= nind*ncov');
       end;
  386: write('if mindeg >= 2 then dom >= CL(girth/3)*Ncomp');
  387: if full = 1 then write('if mindeg >= 2 and girth >= 5 then
dom <= ...more...')
      else
        begin
          writeln('if mindeg >= 2 and girth >= 5 then dom <=');
          if full <> 0 then write('         ');
          writeln('             (nodes-FL(girth/3)-
(girth-4)*(mindeg-2)*(mindeg-3)/2');
          if full <> 0 then write('         ');
          write('                     -2*(mindeg-2)+1)/2');
        end;
  388: if full = 1 then write('if mindeg >= 2 and girth >= 9 then
dom <= ...more...')
      else
        begin
          writeln('if mindeg >= 2 and girth >= 9 then dom <=');
          if full <> 0 then write('         ');
          write('             (nodes-FL(girth/3)-
girth*(mindeg-2)*(mindeg-3)/2+1)/2');
        end;
  389: write('if maxdeg >= 6 and clique < maxdeg then nodes
< maxdeg*nind');
  390: write('if not a 5-cycle, then max{clique,nind} >= .5
*log2(2*sqrt(PI)*nodes)');
  391: if full = 1 then write('if circ <= nodes-mindeg
then ...more...')
         else
           begin
             writeln('if circ <= nodes-mindeg then edges <=');
             if full <> 0 then write('         ');
             write('               nodes(nodes-1)/2-mindeg(nodes-
mindeg-1)');
           end;
  392: write('if girth = 5 then nodes >= 19,30,40 when mindeg =
4,5, >= 6');
  393: if full = 1 then write('if girth = 6 and mindeg >= 7
then ...more...')
```

```
        else
          begin
            writeln('if girth = 6 and mindeg >= 7 then nodes >=
93, if not regular');
             if full <> 0 then write('         ');
            write('
nodes >= 90, otherwise');
          end;
  394: if full = 1 then write('if clique = 2 then nind >= ...
more...')
     else
       begin
         writeln('if clique = 2 then nind >= nodes*(d*ln(d)-d+
1)/(d-1)**2');
         if full <> 0 then write('         ');
         write('          where d = 2*edges/nodes, and ln is the
natural log.');
       end;
  395: if full = 1 then write('if connected and not a tree then
nodes >=...more...')
        else
          begin
            writeln('if connected and not a tree then nodes >=
');
            writeln('                      CL((diam+
1)/(girth+k))*(1+mindeg*S-m*k)');
            if full <> 0 then write('         ');
            writeln('             where k = (1+(-1)**girth)/2');
            if full <> 0 then write('         ');
            writeln('                   t = girth div 2');
            if full <> 0 then write('         ');
            writeln('                   m = (mindeg-1)**t');
            if full <> 0 then write('         ');
            writeln('                   S = 1   if mindeg = 1');
            if full <> 0 then write('         ');
            writeln('                     = t   if mindeg = 2');
            if full <> 0 then write('         ');
            write('                     = (m-1)/(mindeg-2) if
mindeg >= 3');
          end;
  396: if full = 1 then write('if not a forest then edges
<= ..more..')
      else
        begin
          writeln('if not a forest then edges <=
nodes*(nodes-1)/(4m)+nodes/2');
          if full <> 0 then write('         ');
          writeln('           where  t = girth div 2');
          if full <> 0 then write('         ');
          writeln('                  s = 1    if mindeg = 1');
          if full <> 0 then write('         ');
          writeln('                    = t    if mindeg = 2');
          if full <> 0 then write('         ');
          writeln('                    =
((mindeg-1)**t-1)/(mindeg-2), otherwise.');
          if full <> 0 then write('         ');
          writeln('              and m = s-((mindeg-1)**(t-1))/2
if odd girth');
          if full <> 0 then write('         ');
          write('                    = s-(mindeg-1)**(t-1) if
even girth.');
        end;
  397: write('if girth >= 5+4*log3(max{1,genus}) then arbor <=
2');
  398: write('if mindeg >= 2 then nodes >=
girth*Ncomp+maxdeg-2');
  399: if full = 1 then write('if nconn > 0 then
nconn >= ...more...')
       else
        begin
          writeln('if nconn > 0 then nconn >=');
          if full <> 0 then write('         ');
          write('
nodes*(maxdeg-2))/((maxdeg-1)**diam+maxdeg-3)');
        end;
  400: if full = 1 then write('if diam = 2 then ...more...')
    else
      begin
        writeln('if diam = 2 then edges >=');
        if full <> 0 then write('         ');
        writeln('            2*nodes-4,           if (2
*nodes-2)/3 <= maxdeg < nodes-4');
        if full <> 0 then write('         ');
        writeln('                                        or
maxdeg = nodes-2');
        if full <> 0 then write('         ');
        writeln('            3*nodes-maxdeg-6,    if (3
*nodes-3)/5 <= maxdeg < (2*nodes-2)/3');
        if full <> 0 then write('         ');
        writeln('            5*nodes-4*maxdeg-10, if (5
*nodes-3)/9 <= maxdeg < (3*nodes-3)/5');
        if full <> 0 then write('         ');
        write('            4*nodes-2*maxdeg-13, if (nodes+1)/2
<= maxdeg < (5*nodes-3)/9');
     end;

end;
end;


end.
```