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

    procedure ruletext(i,full:longint);

implementation

procedure ruletext(i,full:longint);
(****************************************************)
(*                                                  *)
(*    writes out the theorems for the trace.        *)
(*    full=1 implies the short form is printed.     *)
(*    full=0 implies the long form is printed.      *)
(*    full>1 implies the long form with all but     *)
(*           the first line indented.               *)
(*                                                  *)
(****************************************************)



procedure text500(i,full:longint);
   begin
     case i of

       401: write('if maximal planar and maxdeg <= mindeg+1 then
nconn = mindeg');
  402: write('bwidth <= nodes-(mindeg+1)*(Ncomp-1)-1-FL((nind-
Ncomp+1)/2)');
  403: if full = 1 then write('if regular and mindeg > nconn then
eind > ...more...')
      else
        begin
          writeln('if regular and mindeg > nconn then eind >=
(nodes-t)/2');
          if full <> 0 then write('         ');
          writeln('              where:  t is same parity as
nodes');
          if full <> 0 then write('         ');
          writeln('                       nodes <= (t+3)*(2
*CL(mindeg/2)+1)+x');
          if full <> 0 then write('         ');
          writeln('                       x = 0, mindeg, or 2
*mindeg-2');
          if full <> 0 then write('         ');
          write('                            when  nconn = 1, 2,
or >= 3, respectively');
        end;
  404: write('if mindeg > econn=nconn then nodes >=
mindeg+maxdeg');
  405: write('if mindeg > econn=nconn and diam = 3 then dom <=
econn+1');
  406: if full = 1 then write('if (nodes-1)**2 div 4 < edges and
edges <=...more...')
      else
       begin
         writeln('if (nodes-1)**2 div 4 < edges <=
(nodes-1)*(nodes-2)/2 then');
         if full <> 0 then write('         ');
         write('           mindeg <= econn-1+(nodes-sqrt(4*edges+
2*nodes-nodes**2))/2');
       end;
  407: write('dom <= (nodes-maxdeg-1)*(nodes-mindeg-2)/(nodes-1)+
2');
  408: if full = 1 then write('if diam = 3 and s = FL((m-
sqrt(m*m-4*nodes))/2)...more...')
    else
      begin
        writeln('if diam = 3, s = FL((m-sqrt(m*m-4*nodes))/2)
and');
        writeln('            s <= FL((nodes/2)**(1/3)),');
        if full <> 0 then write('         ');
        writeln('                   where  m = maxdeg+2, ');
        if full <> 0 then write('         ');
        write('            then  edges >= nodes+s*(s-1)/2-1');
      end;
  409: if full = 1 then write(' if nodes >= 4, hamilitonian and
clique = 2 then...more...')
     else
       begin
         writeln('if nodes >= 4, hamiltonian and clique = 2
then');
         if full <> 0 then write('         ');
         write('           edges <= m*m-2m+nodes, where m =
FL(nodes/2)');
       end;
  410: write('spectr >= mindeg');
  411: if full = 1 then write('if connected and nodes >= 2
*mindeg+2 then...more...')
     else
       begin
         writeln('if connected and nodes >= 2*mindeg+2 then ');
         if full <> 0 then write('         ');
         writeln('          diam <= 3*FL(nodes/(mindeg+1))-3+k');
         if full <> 0 then write('         ');
         writeln('           where k = 0, if nodes = s*(mindeg+
1)');
         if full <> 0 then write('         ');
         writeln('                   = 1, if nodes = s*(mindeg+
1)+1');
         if full <> 0 then write('         ');
         writeln('                   = 2, otherwise');
         if full <> 0 then write('         ');
         write('                         ( s a positive
integer )');
       end;
  412: if full = 1 then write('if 3 <= mindeg <= maxdeg-3 and not
a forest then ...more...')
      else
        begin
          writeln('if 3 <= mindeg <= maxdeg-3 and not a forest
then');
          if full <> 0 then write('         ');
          writeln('        eind >= (nodes*(mindeg-1)+(maxdeg-
mindeg+1)*T)/(maxdeg+mindeg-1)');
          if full <> 0 then write('         ');
          writeln('             where:  t = FL((girth-1)/2)');
          if full <> 0 then write('         ');
          writeln('                     k = (1-(-1)**(t-1))/2');
          if full <> 0 then write('         ');
          writeln('                     s =
((mindeg-1)**(t-1)-1)/(mindeg*(mindeg-2))');
          if full <> 0 then write('         ');
          writeln('                     T = (maxdeg*s-(maxdeg-
mindeg)*k)/mindeg');
          if full <> 0 then write('         ');
          writeln('                             when girth is
odd');
          if full <> 0 then write('         ');
          writeln('                       = ((maxdeg+mindeg-2)*s-
(maxdeg-2)*k)/mindeg');
          if full <> 0 then write('         ');
          write('                             when girth is
even');
        end;
  413: if full = 1 then write('edges >= m*(nodes-
chr)+chr*(chr-1)/2-...more...')
      else
        begin
          writeln('edges >= m*(nodes-chr)+chr*(chr-1)/2-
(nind-1)*m*(m+1)/2');
          if full <> 0 then write('         ');
          write('          where m = (nodes-chr) div (nind-1)');
        end;
  414: if full = 1 then write('if diam <= 4 then edges
<=...more...')
      else
        begin
          writeln('if diam <= 4 then edges <= ((p-2)*(p-3)-2
*(p-2)*(d-4)*k');
          if full <> 0 then write('         ');
          writeln('                    -4*k*(k-1)+(d-2)*(d-3)*k**
2)/2');
```

```
          if full <> 0 then write('         ');
          writeln('             where p = nodes');
          if full <> 0 then write('         ');
          writeln('                   d = diam');
          if full <> 0 then write('         ');
          write('                   k = nconn');
        end;
  415: write('dom <= (nodes+2-mindeg) div 2');
  416: if full = 1 then write('if nodes is even and maxdeg =
nodes-2 then...more...')
   else
     begin
       writeln('if nodes is even and maxdeg = nodes-2 then echr =
maxdeg+1 if');
       if full <> 0 then write('         ');
       write('              and only if  edges >= ((nodes-2)**
2)/2+1+mindeg');
    end;
  417: write('if clique = 2 and maxdeg <= 3 then edges >= 13
*nodes/2-14*nind');
  418: write('if clique = 2 and maxdeg <= 2 then edges >= 7
*nodes-15*nind');
  419: if full = 1 then write('if clique = 2 and ncov <= 3
*nodes/5 then ...more...')
    else
      begin
        writeln('if clique = 2 and ncov <= 3*nodes/5 then ncov
<=');
        if full <> 0 then write('         ');
        write('            (3*nodes-sqrt(5*edges-nodes**2))/5');
      end;
  420: if full = 1 then write('if clique = 2 and nind >= 2
*nodes/5 then ...more...')
    else
      begin
        writeln('if clique = 2 and nind >= 2*nodes/5 then
nind >=');
        if full <> 0 then write('         ');
        write('            (2*nodes+sqrt(5*edges-nodes**2))/5');
      end;
  421: if full = 1 then write('if nind = 2 and clique >= 2
*nodes/5 then ...more...')
    else
      begin
        writeln('if nind = 2 and clique >= 2*nodes/5 then
clique >=');
        if full <> 0 then write('         ');
        write('            (2*nodes+sqrt(nodes*(3*nodes-5)/2-5
*edges))/5');
      end;
   422: if full = 1 then write('if not a forest then
bwidth >= ...more...')
```

```
      else
        begin
          writeln('if not a forest then bwidth >= (2
*ncov*(girth-2)-nodes*(girth-3))/');
          if full <> 0 then write('         ');
          write('               (2*nodes-2*ncov)');
        end;
   423: write('ncov <= nodes-maxdeg/(chr-1)');
   424: if full = 1 then write('if connected, clique = 2,
and..more..')
     else
       begin
         writeln('if connected, clique = 2, and not odd cycle or
even path then');
         if full <> 0 then write('         ');
         writeln('            ncov <= nodes*(x-1)/x-nodes**
2/(nodes+2*edges)');
         if full <> 0 then write('         ');
         write('               where x = maxdeg**2+maxdeg');
       end;
   425: write('if planar then mindeg <= nodes-ncov+2');
   426: if full = 1 then write('edges <= MAX{(nodes-ecov)*(2
*nodes-2*ecov+1),...more..')
     else
       begin
         writeln('edges <= MAX{(nodes-ecov)*(2*nodes-2*ecov+
1),');
         if full <> 0 then write('         ');
         write('                  (nodes-
ecov)*(nodes+ecov-1)/2}');
       end;
   427: 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('                     (nodes+2*FL((nodes+1)/2x))/2
if nodes even');
       if full <> 0 then write('         ');
       writeln('             ecov <=');
       if full <> 0 then write('         ');
       writeln('                     (nodes+MAX{2*FL((nodes+1
+x)/2x)-1,1})/2 otherwise');
       if full <> 0 then write('         ');
       write('                        where x =
mindeg*FL((mindeg+3)/2)-1');
     end;
    428: write('if cubic then ecov <= nodes div 2+(nodes+3) div
18+(Ncomp+4) div 6');
    429: write('if clique = 2 and maxdeg <= 4 then edges >= 13
*ncov-7*nodes');
```

```
    430: if full = 1 then write('if nconn >= 2 and ncov <=
nodes-2 then...more..')
   else
     begin
       writeln('if nconn >= 2 and ncov <= nodes-2 then');
       if full <> 0 then write('         ');
       write('             circ >= (4*nodes-2*ncov-4)/(nodes-
ncov)');
     end;
   431: write('if girth >= 6 then ncov <= (nodes*maxdeg**
2)/(maxdeg**2+2*maxdeg-1)');
   432: if full = 1 then write('if cubic then ncov <= 33
*nodes/52, when ...more..')
   else
     begin
       writeln('if cubic then ncov <= 33*nodes/52, when girth >=
6');
       if full <> 0 then write('         ');
       write('                   <= 33*nodes/53, when girth >=
8');
     end;
   433: write('if regular and ecov > nodes/2 then echr = maxdeg+
1');
   434: write('if regular then ecov <= (nodes*(maxdeg+2))/(2
*maxdeg+2)');
   435: write('if regular and nodes = 2*maxdeg+1 then ncov >=
nodes-nconn');
   436: if full = 1 then write(' if girth >= 4 and mindeg = 2
then ncov <= ...more...')
  else
    begin
      writeln('if girth >= 4 and mindeg = 2 then ncov <= nodes-
maxdeg*FL(t+2)/2)-x');
      if full <> 0 then write('         ');
      writeln('                where t = FL((girth-2)/2)');
      if full <> 0 then write('         ');
      write('                       x = 1 when t is even,
otherwise = 0');
    end;
   437: if full = 1 then write('if girth >= 4 and mindeg >= 3
then ncov <= ...more...')
   else
     begin
       writeln('if girth >= 4 and mindeg >= 3 then ncov <= nodes-
x');
       if full <> 0 then write('         ');
       writeln('              where t = Fl((girth-2)/2) and');
       if full <> 0 then write('         ');
       writeln('                x = maxdeg*((mindeg-1)**(t+
1)-1)/(mindeg*(mindeg-2))');
       if full <> 0 then write('         ');
       writeln('                         when t is odd, and');
       if full <> 0 then write('         ');
       writeln('                x =
maxdeg*((mindeg-1)**t-1)/(mindeg*(mindeg-2))-1');
       if full <> 0 then write('         ');
       write('                         when t is even');
     end;
   438: write('nind >= (2*nodes-edges+eind)/4');
   439: write('ncov <= (nodes+edges+ecov)/4');
   440: write('nind >= (3*nodes-edges-ecov)/4');
   441: if full = 1 then write('if maxdeg >= 6 and clique
< maxdeg then...more..')
  else
    begin
      writeln('if maxdeg >= 6 and clique < maxdeg then ');
      if full <> 0 then write('         ');
      write('               ncov <=
(nodes*(maxdeg-1)-1)/maxdeg');
    end;
   442: if full = 1 then write('if clique = 2 then ncov <= ...
more...')
  else
    begin
      writeln('if clique = 2 then ncov <= nodes-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;
   443: write('bwidth <= nodes-(mindeg+1)*(ncomp-1)-1-FL((nodes-
ncov-ncomp+1)/2)');
  444: if full = 1 then write('if regular and mindeg > nconn then
ecov <=...more...')
      else
        begin
          writeln('if regular and mindeg > nconn then ecov <=
(nodes+t)/2');
          if full <> 0 then write('         ');
          writeln('             where:  t is an integer with the
same parity as nodes');
          if full <> 0 then write('         ');
          writeln('                 nodes <= (t+3)*(2
*CL(mindeg/2)+1)+x');
          if full <> 0 then write('         ');
          writeln('                 x = 0, mindeg, or 2
*mindeg-2');
          if full <> 0 then write('         ');
          write('                   when  nconn = 1, 2, or >= 3,
respectively');
        end;
  445: if full = 1 then write('edges >= m*(nodes-
chr)+chr*(chr-1)/2-...more...')
   else
```

```
     begin
       writeln('edges >= m*(nodes-chr)+chr*(chr-1)/2-(nodes-
ncov-1)*m*(m+1)/2');
       if full <> 0 then write('         ');
       write('                 where m = (nodes-chr) div (nodes-
ncov-1)');
     end;
  446: write('if clique = 2 and maxdeg <= 3 then edges >= 14
*ncov-(15*nodes) div 2');
  447: write('if clique = 2 and maxdeg <= 2 then edges >= 15
*ncov-8*nodes');
  448: write('edges <= ((nodes-
nccov)*(nccov+maxdeg-1)+mindeg)/2');
  449: if full = 1 then write('if nind = 2 and mindeg >= nodes-5
then edges <=...more...')
  else
    begin
      writeln('if nind = 2 and mindeg >= nodes-5 then');
      if full <> 0 then write('         ');
      write('                 edges <= nodes*(nodes-13)/2+13
*clique');
    end;
  450: if full = 1 then write('if nind < nccov = nodes-mindeg-1
then ..more..')
   else
     begin
       writeln('if nind < nccov = nodes-mindeg-1 then');
       if full <> 0 then write('         ');
       write('                 nodes <= 2*mindeg+3 ( -1 when
nodes >= mindeg+10)');
     end;
   451: if full = 1 then write('if mindeg <= MIN{nodes-7, nodes-
nind-2} then ..more...')
  else
    begin
      writeln('if mindeg <= MIN{nodes-7, nodes-nind-2} then ');
      if full <> 0 then write('         ');
      write('                  mindeg  <= ((nodes-1)*(clique-1)+
2) div clique');
    end;
  452: if full = 1 then write('if nind = 2 then clique >= ...
more...')
     else
       begin
         writeln('if nind = 2 then clique >= nodes*(d*ln(d)-d+
1)/(d-1)**2');
         if full <> 0 then write('         ');
         write('               where d = nodes-1-2*edges/nodes,
and ln is the natural log.');
       end;
  453: if full = 1 then write('edges <= nodes*(nodes-1)/2-
m*(nodes-nccov)...more...')
```

```
      else
        begin
          writeln('edges <= nodes*(nodes-1)/2-m*(nodes-nccov)-
nccov*(nccov-1)/2');
          if full <> 0 then write('         ');
          writeln('                       +(clique-1)*m*(m+
1)/2');
          if full <> 0 then write('         ');
          write('               where m = (nodes-nccov) div
(clique-1)');
        end;
  454: if full = 1 then write('if nind <= 2 and mindeg >= nodes-4
then ...more..')
  else
    begin
      writeln('if nind <= 2 and mindeg >= nodes-4 then edges
<=');
      if full <> 0 then write('         ');
      write('                   nodes*(nodes-14)/2+14*clique');
    end;
  455: if full = 1 then write('if nind <= 2 and mindeg >= nodes-3
then...more...')
   else
     begin
       writeln('if nind <= 2 and mindeg >= nodes-3 then edges
<=');
       if full <> 0 then write('         ');
       write('                   nodes*(nodes-15)/2+15*clique');
     end;
  456:write('if clique = 2 then chr <= (3*ncov+52)/16');
  457:write('if clique = 2 then chr <= (3*(nodes-nind)+52)/16');
  458:write('if nind = 2 then nccov <= (3*(nodes-clique)+
52)/16');
end;
end;

begin
  if (0 < i) and (i <= actualnumrules) then
     begin
       if i <= 150 then text150(i,full)
          else if i <= 300 then text300(i,full)
            else if i <= 400 then text400(i,full)
                  else  text500(i,full);
     end
   else writeln(' is an invalid theorem number.');
end;

end.
```