```
begin comment ALGOL editor. The data tape must contain the
  following control numbers: 1. numofpr: the number of programs to
  be edited, 2. width: the maximal number of positions on a line,
  3. page: the number of lines per type area, 4. rest: the number
  of blank lines between two pages, 5. n: the upper index of the
  integer array pos (see 6.), 6. pos[1:n]: n numbers which
  regulate the indentations;
  integer symbol, i, s, ih, sh, breaki, breaks, tabstop, level,
  arlevel, stringlevel, line, comm, decl, lab, proc, pointer, a, b,
  c, h, k, w, zz, nop, numofpr, width, page, rest, n;
  boolean booll, boolb, boolc;
  integer array buffer[1:2000], stock[1:40], proclevel[0:10];
  numofpr:= read; width:= read; page:= read; rest:= read; n:= read;
  begin integer array pos[0:n];

      procedure sym(n); value n; integer n;
      begin PUSYM(n); PRSYM(n) end;


      procedure space(n); value n; integer n;
      begin integer i;
         for i:= 1 step 1 until n do sym(93)
      end;


      procedure tabspace(n); value n; integer n;
      begin integer p, q;
         p:= n : 8; q:= n - p × 8;
         for a:= 1 step 1 until p do sym(118); space(q)
      end;


      procedure punchline(border); value border; integer border;
      begin tabspace(tabstop - (if boolb ∧ booll then (pos[b] + (if
         level ≤ n then pos[b - 1] else pos[n])) else if booll ∨
         boolb then pos[b] else 0)); booll:= boolb:= false;
         for a:= 1 step 1 until border do sym(buffer[a]); newline
      end;


      procedure newline;
      begin sym(119); line:= line + 1; if line ≥ page then
         begin RUNOUT;
            for a:= 1 step 1 until rest do sym(119); line:= 0
         end
      end;


      procedure punchbuffer;
      begin punchline(i - 1); i:= 1; s:= tabstop; breaki:= breaks:= 0
      end;


      procedure restbuffer(n); value n; integer n;
```

```
begin i:= i - n - 1;
   for a:= 1 step 1 until i do buffer[a]:= buffer[n + a];
   i:= i + 1
end;


procedure label;
begin if i < pos[b] then
   begin for a:= i step 1 until pos[b] do buffer[a]:= 93;
      i:= pos[b] + 1
   end
   else
   begin buffer[i]:= 93; i:= i + 1 end;
   lab:= 0
end;


procedure breakbuffer;
begin integer n;
   if breaki > 0 ∧ zz = 0 then
   begin punchline(breaki); restbuffer(breaki);
      s:= s - breaks + tabstop; if lab = 1 then
      begin label; s:= i + tabstop end
   end
   else
   begin for a:= i - 2 step - 1 until 1 do if buffer[a] = 93
      then
      begin n:= a; goto break1 end;
      goto break2;
   break1: punchline(n); restbuffer(n);
   break2: if lab = 1 then label; s:= i + tabstop;
      if s ≥ width then emergency
   end;
   breaki:= breaks:= 0
end;


procedure stockbuffer(spacesbefore, spacesafter, c);
value spacesbefore, spacesafter, c;
integer spacesbefore, spacesafter, c;
begin for a:= 1 step 1 until spacesbefore do buffer[i + a -
   1]:= 93; i:= i + spacesbefore; s:= s + spacesbefore;
   for a:= 1 step 1 until c do buffer[i + a - 1]:= stock[a];
   i:= i + c; s:= s + c ; 2;
   for a:= 1 step 1 until spacesafter do buffer[i + a - 1]:= 93;
   i:= i + spacesafter; s:= s + spacesafter;
   if comm = 0 then goto start1
end;


procedure semicolon;
begin buffer[i]:= 91; i:= i + 1; zz:= 0; punchbuffer;
   if level = proclevel[pointer] then
   begin RUNOUT; newline; pointer:= pointer - 1;
      proc:= if pointer = 0 then - 1 else 1
```

```
        end;
        goto start
      end;


      integer procedure undsym;
      begin
      undl: symbol:= RESYM; if symbol = 126 then goto undl;
        undsym:= symbol
      end;


      procedure emergency;
      begin ih:= i; sh:= s;
        for a:= ih step - 1 until 1 do
        begin ih:= ih - 1; sh:= sh - 1;
          if buffer[a] = 87 ∧ sh < width then
          begin punchline(ih + 1); restbuffer(ih + 1);
            s:= i + tabstop; breaki:= breaks:= 0; goto endem
          end;
          k:= a
        end;
        if k = 1 then
        begin sym(119); PRINTTEXT(
*
the chosen indentations are too large with respect to
the maximal number of positions on the line*
          ); EXIT
        end;
      endem:
      end;

  begin of program: for a:= 1 step 1 until n do pos[a]:= read;
      nop:= 0;
  repeat: i:= s:= line:= 1; booll:= boolb:= boolc:= false;
      breaki:= breaks:= tabstop:= level:= arlevel:= stringlevel:=
      comm:= decl:= lab:= b:= pointer:= zz:= proclevel[0]:=
      pos[0]:= 0; proc:= - 1; RUNOUT; sym(119);
  start: symbol:= RESYM;
      if i > 1 ∧ (symbol = 93 ∨ symbol = 119) then
      begin if buffer[i - 1] > 63 then goto start else
        begin for symbol:= RESYM while symbol = 93 ∨ symbol = 119
          do ; if symbol < 63 then
          begin buffer[i]:= 93; buffer[i + 1]:= symbol; i:= i + 2;
            s:= s + 2; goto start
          end
        end
      end;
  start1: if symbol = 93 ∨ symbol = 118 ∨ symbol = 119 then goto
      start; if s ≥ width then
      begin breakbuffer; goto start1 end;
      if symbol < 64 ∨ symbol = 76 ∨ symbol = 88 ∨ symbol = 89 ∨
      symbol = 98 ∨ symbol = 99 then
      begin comment digits, letters, ¬, ., ₁₀, (, );
        buffer[i]:= symbol; i:= i + 1; s:= s + 1; goto start
```

```
end
else if symbol = 64 ∨ symbol = 65 ∨ symbol = 66 ∨ symbol = 67
∨ symbol = 70 ∨ symbol = 72 ∨ symbol = 74 ∨ symbol = 79 ∨
symbol = 80 then
begin comment +, −, ×, /, =, <, >, ∧, ∨;
   if buffer[i − 1] ≠ 93 then
   begin buffer[i]:= 93; i:= i + 1; s:= s + 1 end;
   buffer[i]:= symbol; buffer[i + 1]:= 93; i:= i + 2; s:= s + 2;
   goto start
end
else if symbol = 100 ∨ symbol = 101 then
begin comment [, ];
   buffer[i]:= symbol; i:= i + 1; s:= s + 1;
   arlevel:= (if symbol = 100 then 1 else − 1) + arlevel;
   goto start
end
else if symbol = 87 then
begin comment ,;
   buffer[i]:= symbol; if arlevel = 0 then
   begin buffer[i + 1]:= 93; i:= i + 2; s:= s + 2 end
   else
   begin i:= i + 1; s:= s + 1 end;
   goto start
end
else if symbol = 91 then
begin comment semicolon;
   buffer[i]:= symbol; buffer[i + 1]:= 93; if s < width then
   begin breaki:= i + 1; breaks:= s + 1 end;
   i:= i + 2; s:= s + 2; if decl = 1 then
   begin if proc ≠ 0 then punchbuffer; decl:= 0; goto start
   end;
   if proc = 0 then
   begin punchbuffer; RUNOUT; newline;
      proc:= if pointer = 0 then − 1 else 1
   end;
   goto start
end
else if symbol = 90 then
begin comment colon;
   if arlevel ≠ 0 then
   begin buffer[i]:= 90; i:= i + 1; s:= s + 1; goto start end
   else
   begin symbol:= RESYM; if symbol = 70 then
      begin buffer[i]:= 90; buffer[i + 1]:= 70;
         buffer[i + 2]:= 93; i:= i + 3; s:= s + 3; goto start
      end
      else
   pardel: if symbol = 93 then
      begin symbol:= RESYM; goto pardel end;
      if symbol = 98 then
      begin buffer[i]:= 90; buffer[i + 1]:= 93;
         buffer[i + 2]:= 98; i:= i + 3; s:= s + 3; goto start
      end
      else
      begin buffer[i]:= 90; i:= i + 1; if level = 0 then
```

```
            begin punchbuffer; goto start1 end;
            lab:= 1;
            for a:= i - 2 step - 1 until 1 do if buffer[a] = 126
            ∨ buffer[a] = 91 ∨ buffer[a] = 90 then goto labinst;
            label; booll:= true; goto start1;
         labinst: if buffer[a] = 126 then breaki:= a + 2 else
            if buffer[a] = 90 then
            begin if a < pos[b] then breaki:= pos[b] else
               breaki:= a + 1
            end;
            breakbuffer; booll:= true; goto start1
         end
      end
   end
else if symbol = 127 then
begin comment |;
bar: symbol:= RESYM; if symbol = 127 then goto bar;
   if symbol = 80 ∨ symbol = 70 then
   begin comment ∧, ≠ ;
      buffer[i]:= 93; buffer[i + 1]:= 127;
      buffer[i + 2]:= symbol; buffer[i + 3]:= 93; i:= i + 4;
      s:= s + 3; goto start
   end
   else if symbol = 72 then
   begin comment ≤;
      buffer[i]:= 127; buffer[i + 1]:= 72; breaki:= i - 1;
      breaks:= s - 1; i:= i + 2; s:= s + 1;
   string: symbol:= RESYM; buffer[i]:= symbol;
      if symbol ≠ 127 then
      begin i:= i + 1; if symbol ≠ 126 then
         begin if symbol = 118 then s:= s + 8 else s:= s + 1;
            if symbol = 119 then line:= line + 1
         end;
         if s ≥ width - 4 ∧ breaks > 0 then breakbuffer;
         goto string
      end
      else
      begin
      bar1: symbol:= RESYM; if symbol = 127 then goto bar1;
         if symbol = 74 then
         begin comment ≥;
            buffer[i + 1]:= symbol; i:= i + 2; s:= s + 1;
            if stringlevel ≠ 0 then
            begin stringlevel:= stringlevel - 1; goto string
            end
            else if s > width - 3 then
            begin for a:= 1 step 1 until i - 1 do
               sym(buffer[a]); newline; i:= 1; s:= tabstop
            end;
            goto start
         end
         else if symbol = 72 then
         begin comment nested stringquotes;
            stringlevel:= stringlevel + 1;
            buffer[i + 1]:= symbol; i:= i + 2; s:= s + 1;
```

```
            goto string
         end
         else
         begin buffer[i + 1]:= symbol; i:= i + 2; s:= s + 1;
            goto string
         end
      end
   end
   else
   begin buffer[i]:= 127; i:= i + 1; goto start1 end
end
else if symbol = 126 then
begin comment _;
   symbol:= undsym;
   if symbol = 70 ∨ symbol = 72 ∨ symbol = 74 ∨ symbol = 76 ∨
   symbol = 90 then
   begin comment =, ≤ , ≥ , ¬ , : ;
      buffer[i]:= 93; buffer[i + 1]:= 126;
      buffer[i + 2]:= symbol; buffer[i + 3]:= 93; i:= i + 4;
      s:= s + 3; goto start
   end
   else
   begin stock[1]:= 126; stock[2]:= symbol; stock[3]:= RESYM;
      for a:= 4 step 2 until 18 do
      begin stock[a]:= undsym; symbol:= stock[a + 1]:= RESYM;
         if symbol ≠ 126 then
         begin k:= a; goto again end
      end;
   again: if (stock[2] = 15 ∧ stock[4] = 10) ∨ (stock[2] = 29
      ∧ stock[4] = 27) then
      begin comment false, true;
         stockbuffer(0, 0, k)
      end
      else if stock[2] = 11 ∧ stock[4] = 14 then
      begin comment begin;
         if i > pos[b] then
         begin for a:= i - 1 step - 1 until i - pos[b] do if
            buffer[a] ≠ 93 then
            begin c:= a;
               if buffer[c] = 90 ∧ c < pos[b] then goto labeg
               else goto pun
            end
         end;
      pun: if i ≠ 1 then punchbuffer;
      labeg: comm:= 1; stockbuffer(0, 1, k); comm:= 0;
         if proc = 0 then
         begin pointer:= pointer + 1;
            proclevel[pointer]:= level; proc:= 1
         end;
         level:= level + 1; b:= if level < n then level else n;
         boolb:= true; tabstop:= tabstop + pos[b];
         w:= pos[b] - 6; if w > 0 then
         begin for a:= 0 step 1 until w - 1 do buffer[i + a]:=
            93; i:= i + w; s:= s + w
         end;
```

```
      goto start1
   end
   else if stock[2] = 13 ∨ (stock[2] = 28 ∧ stock[4] = 29 ∧
   stock[6] = 14) ∨ stock[2] = 32 ∨ stock[2] = 29 ∨
   stock[2] = 30 then
   begin comment do, step, while, then, until;
      stockbuffer(1, 1, k)
   end
   else
e: if stock[2] = 14 ∧ stock[4] = 21 then
   begin comment else;
      if zz = 0 then stockbuffer(1, 1, k) else
      begin zz:= 0; stockbuffer(0, 1, k) end
   end
   else if stock[2] = 14 then
   begin comment end;
      if buffer[1] = 126 ∧ buffer[2] = 11 ∧ buffer[4] = 14
      ∧ s < width - 5 ∧ zz = 0 then zz:= 1 else zz:= 0;
      if zz = 1 then
      begin boolb:= false; goto goon end;
      if i ≠ 1 then punchbuffer;
   goon: comm:= 1;
      if zz = 0 then stockbuffer(0, 0, k) else
      stockbuffer(1, 0, k); comm:= 0;
      tabstop:= tabstop - pos[b]; level:= level - 1;
      if level = 0 then
      begin punchbuffer; sym(119); RUNOUT; nop:= nop + 1;
         if nop = numofpr then goto end else
         begin k:= page + rest - 1;
            for a:= line step 1 until k do sym(119);
            goto repeat
         end
      end
      else b:= if level < n then level else n;
      if symbol = 93 ∨ symbol = 118 ∨ symbol = 119 then
      begin
      reject: symbol:= RESYM;
         if symbol = 93 ∨ symbol = 118 ∨ symbol = 119 then
         goto reject else if symbol = 126 then
         begin stock[1]:= 126;
            for a:= 2 step 2 until 18 do
            begin stock[a]:= undsym;
               symbol:= stock[a + 1]:= RESYM;
               if symbol ≠ 126 then
               begin k:= a; goto nextund end
            end;
         nextund: if stock[2] = 14 then
            begin if i ≠ 1 then punchbuffer; zz:= 1; goto e
            end
            else
            begin if zz = 1 then
               begin breakbuffer; zz:= 0 end;
               buffer[i]:= 93; i:= i + 1; s:= s + 1;
               boolc:= true; goto comm1
            end
```

```
        end
        else if symbol = 91 then semicolon else
        begin if zz = 1 then
           begin breakbuffer; zz:= 0 end;
           buffer[i]:= 93; buffer[i + 1]:= symbol;
           i:= i + 2; s:= s + 2; boolc:= true; goto comm2
        end
     end
     else if symbol = 91 then semicolon else
     begin if zz = 1 then
        begin breakbuffer; zz:= 0 end;
        buffer[i]:= symbol; i:= i + 1; s:= s + 1;
        goto comm2
     end
end
else if stock[2] = 12 then
begin comment comment;
comm1: comm:= 1; stockbuffer(0, 1, k); comm:= 0;
nospace: if symbol = 93 then
   begin symbol:= RESYM; goto nospace end;
   buffer[i]:= symbol; i:= i + 1; s:= s + 1;
comm2: symbol:= RESYM;
   if symbol ≠ 118 ∧ symbol ≠ 119 ∧ symbol ≠ 93 then
   buffer[i]:= symbol else if buffer[i - 1] ≠ 93 then
   buffer[i]:= 93 else goto comm2; if symbol ≠ 91 then
   begin i:= i + 1;
      if symbol ≠ 126 then s:= s + 1 else if boolc then
      begin buffer[i]:= symbol:= undsym; i:= i + 1;
         s:= s + 1; if symbol ≠ 14 then goto comm2 else
         begin stock[1]:= 126; stock[2]:= 14;
            symbol:= stock[3]:= RESYM;
            if symbol = 126 then
            begin for a:= 4 step 2 until 10 do
               begin stock[a]:= undsym;
                  symbol:= stock[a + 1]:= RESYM;
                  if symbol ≠ 126 then
                  begin k:= a; i:= i - 3; s:= s - 2;
                     if stock[4] = 21 ∨ stock[4] = 23
                     then
                     begin i:= i + 1; punchbuffer;
                        boolc:= false;
                        if stock[4] = 21 then zz:= 1;
                        goto e
                     end
                     else
                     begin comm:= 1; stockbuffer(0, 0, k);
                        comm:= 0; goto comm2
                     end
                  end
               end
            end
            else
            begin comm:= 1; stockbuffer(0, 0, 2);
               comm:= 0; buffer[i]:= symbol; goto comm2
            end
```

```
            end
         end;
         if s ≥ width then breakbuffer; goto comm2
      end
      else
      begin i:= i + 1; punchbuffer;
         if level = proclevel[pointer] then
         begin RUNOUT; newline; pointer:= pointer - 1;
            proc:= if pointer = 0 then - 1 else 1
         end;
         boolc:= false; goto start
      end
   end
   else if stock[2] = 27 ∨ (stock[2] = 18 ∧ stock[4] = 23)
   ∨ stock[2] = 11 ∨ stock[2] = 38 then
   begin comment real, integer, boolean, Boolean;
      if stock[k + 1] = 93 then
      begin for a:= k + 2 step 1 until 100 do
         begin symbol:= RESYM; if symbol ≠ 93 then goto dec
         end;
      dec: if symbol = 126 then
         begin stock[k + 2]:= 126;
            for a:= k + 4 step 2 until k + 20 do
            begin stock[a - 1]:= undsym;
               symbol:= stock[a]:= RESYM;
               if symbol ≠ 126 then
               begin h:= a; goto arproc end
            end;
         arproc: if stock[k + 3] = 10 then
            begin comment <type> array;
               decl:= 1; stock[h]:= 93; stockbuffer(0, 0, h)
            end
            else
            begin comment <type> procedure;
               if i ≠ 1 then punchbuffer; if proc ≠ 0 then
               begin RUNOUT; newline; proc:= 0 end;
               decl:= 1; stock[h]:= 93; stockbuffer(0, 0, h)
            end
         end
         else
         begin stock[k + 2]:= symbol;
            symbol:= stock[k + 3]:= RESYM; decl:= 1;
            stockbuffer(0, 0, k + 2)
         end
      end
      else
      begin decl:= 1; stockbuffer(0, 1, k) end
   end
   else if stock[2] = 25 then
   begin comment procedure;
      if proc ≠ 0 then
      begin if i ≠ 1 then punchbuffer; RUNOUT; newline;
         proc:= 0
      end;
      decl:= 1; stockbuffer(0, 1, k)
```

```
                end
                else if stock[2] = 15 then
                begin comment for;
                   if i > pos[b] then
                   begin for a:= i - 1 step - 1 until i - pos[b] do if
                      buffer[a] ≠ 93 then
                      begin c:= a; goto nonl2 end;
                   nonl2: if c > 10 then
                      begin if (buffer[c] = 23 ∧ buffer[c - 1] = 126 ∧
                         buffer[c - 2] = 18) ∨ buffer[c] = 90 then goto
                         lafor
                      end;
                      if buffer[c] = 90 then goto lafor
                   end;
                   if i ≠ 1 then punchbuffer;
                lafor: stockbuffer(0, 1, k)
                end
                else if stock[2] = 24 ∨ (stock[2] = 28 ∧ stock[4] = 29)
                ∨ stock[2] = 10 ∨ stock[2] = 28 ∨ stock[2] = 21 ∨
                stock[2] = 31 then
                begin comment own, string, array, switch, label, value;
                   decl:= 1; stockbuffer(0, 1, k)
                end
                else stockbuffer(0, 1, k)
             end
          end
          else
          begin buffer[i]:= symbol; i:= i + 1; s:= s + 1; goto start end;
    end: STOPCODE
    end
end
```