```
x,clear,pinf<
x,clear,plot2<
algol<
Program PLOT-2
begin
   message compile PLOT-2;
   integer i, at, ty, code, l, xx, yy, xxx, yyy, oldty, pinf,
   free, ident, track, case, f, ff, calcno, drawno, size, oldrand, points, deg;
   real deltax, deltay, t, x0, y0, xol, yol, h, hæld, a, b, r2, A, B, C, D;
   integer array REF[0:39];
   array X, Y[1:20], COEF[0:19];
   boolean plot, high;
   core code plot, xx, yy;
   2, 46;
   1, 44;
   1, 44;
   gs     re5 , ps   (b2)  ;  s := display reference
   xrn      0 , ann   sa2  ;  M := 0, R := abs(xx)
   sn     sa3 , cl      1  ;  - abs(yy)
   arn    sa2 , cl      1  ;  xx
   arn    sa3 , cl      1  ;  yy
   cln     -8 , ga    re6  ;  store bit×4
   arn    re7 , pm    re8  ;  fetch parameters
e6:cl       0 , mb    re1  ;  move bits
   gaX    re3              ;  store straight step
   mb     re1 , ga    re4  ;  store diagonally step
   annX   sa2              ;  M := abs(xx)
   ann    sa3 , tl      1  ;  R := abs(yy), M := M×2, R := R×2
   bs   (re6) t +12        ;  shift M and R
   xr       0              ;  if abs(yy) > abs(xx)
   gr     sa3 , gm    sa2  ;  store yy and xx
   srn    sa2 , tl     -1  ;  test := - xx:2
e2:hv     re1 , ar    sa3  ;  test := test + yy
e3:syV      0          LT  ;  straight step
e4:sy       0 , sr    sa2  ;  diagonally step
e1:xr      15 , sr    c42  ;  xx := xx - 1
   hhX    re2          NT  ;  xx ≥ 0
e5:ps       0 , hr     s1  ;  exit
e7:qq  4.9+4.13+ 2.17+2.21+ 8.25+1.29+ 8.33+1.37  ;  parameter 1
e8:qq 12.9+5.13+10.17+3.21+12.25+5.29+10.33+3.37  ;  parameter 2
   e;

comment
```

```
;
   procedure PEN(type);
   value type;
   integer type;
   if type ≠ oldty then
   begin
      oldty := type;
      writechar(if type = 0 then 16 else 32)
   end PEN;

comment
```

```
;
   procedure PCURVE(x, y, t, t1, t2, n, type);
   value t1, t2, n, type;
   integer n, type;
   real x, y, t, t1, t2;
   begin
      integer i, ty, s;
      ty := 0;
      if n < 1 then n := 1;
      if kbon then
      begin
         s := select(17);
         writecr;
         writetext(‹<PCURVE n: ›);
         writeinteger(‹p›,n);
         select(s)
      end;
      for i := 0 step 1 until n do
      begin
         t := (t1×(n-i) + t2×i)/n;
         PLINE(x, y, ty);
         ty := type
      end
   end PCURVE;

comment
```

```
;
   procedure PDASH(x, y, on, off);
   value x, y, on, off;
   real x, y, on, off;
   begin
      integer i, n;
      real dist, x0, y0, dx, dy, r1, r2, r3, r4;
      x0 := deltax×xxx;
      y0 := deltay×yyy;
      dx := x - x0;
      dy := y - y0;
      dist := sqrt(dx↑2+dy↑2);
      n := (dist+off)/(on+off);
      r1 := on×dx/dist;
      r2 := on×dy/dist;
      r3 := off×dx/dist;
      r4 := off×dy/dist;
      for i := 1 step 1 until n do
      begin
         PLINE(x0+i×r1+(i-1)×r3, y0+i×r2+(i-1)×r4, 1);
         if i < n then
         PLINE(x0+i×(r1+r3), y0+i×(r2+r4), 0);
      end for i;
   end PDASH;

comment
```

```
;
   procedure PLINE(x, y, type);
   value x, y, type;
   integer type;
   real x, y;
   begin
      PEN(type);
      xx := x/deltax - xxx;
      yy := y/deltay - yyy;
      xxx := xxx + xx;
      yyy:= yyy + yy;
      gier(plot)
   end PLINE;

comment
```

```
;
   procedure PSYMB(symb, x0, y0, hcm, hæld);
   value symb, hcm, hæld;
   integer symb;
   real x0, y0, hcm, hæld;
   begin
      integer pos, bit, list, k, pt, ty, f, ff, i, j, bi, kr, cr;
      real x, y, xol, yol, a, b, r2, xo, yo, t1, t2;
      boolean aa;
      integer array REF[0:39];
      boolean array A[0:39];
      integer procedure next;
      begin
         pos := pos + bit;
         if pos > 40 then
         begin
            pos := bit;
            list := list + 1;
            if list > 39 then
            begin
               k := k + 1;
               get(A, pinf, k);
               list := 0
            end;
            aa := A[list]
         end;
         next := integer((aashift pos) ∧ boolean bi)
      end;
      if pinf = 0 then go to NOLIBRARY;
      if symb > 127 then symb := symb - 63;
      k := symb:40 + 2;
      pt := symb mod 40;
      get(A, pinf, k);
      aa := A[pt];
      list := integer((aa ∧ 10 0 20 m) shift - 10);
      bit := integer((aa ∧ 30 0 5 m) shift - 5);
      f := integer(aa ∧ 35 0 5 m);
      get(A, pinf, 1);
      hcm := hcm×real A[f];
      hæld := 3.14159265/180×hæld;
      hæld := cos(hæld)/sin(hæld);
      if hæld < 0.0017 then hæld := 0;

comment
```

```
;
      k := list:40 + integer A[0];
      get(A, pinf, k);
      list := list mod 40;
      bi := 0;
      j := 1;
      for i := 1 step 1 until bit do
      begin
         bi := bi + j;
         j := j×2
      end;
      pos := 0;
      aa := A[list];
      xol := xo := x := x0;
      yol := yo := y := y0;
      for pt := next while pt ≠ bi do
      begin
         if pt < 6 then
         begin
            x := next×hcm + xo;
            y := next×hcm;
            x := x + y×hæld;
            y := y + yo;
            if pt < 2 then PLINE(x, y, pt) else
            begin
               r2 := (x - (y-yo)×hæld-a)↑2 + (y-b)↑2;
               a := a + (b-yo)×hæld;
               f := if pt = 3 ∨ pt = 4 then -1 else 1;
               t1 := sqrt(abs(r2-(xol-a)↑2))×hæld;
               t2 := sqrt(abs(r2-(x-a)↑2))×hæld;
               if pt < 4 then
               PCURVE(t+sqrt(abs(r2-(t-a)↑2))×hæld×f,
               b+sqrt(abs(r2-(t-a)↑2))×f, t, xol - t1×f,
               x - t2×f, 100×abs(xol-x), 1) else
               begin
                  ff := if y > yol then -1 else 1;
                  PCURVE(t+sqrt(abs(r2-(t-a)↑2))×hæld×ff,
                  b+sqrt(abs(r2-(t-a)↑2))×ff, t, xol-t1×ff, a + sqrt(r2)×f,
                  100×abs(xol-a-sqrt(r2)×f), 1);
                  PCURVE(t-sqrt(abs(r2-(t-a)↑2))×hæld×ff,
                  b-sqrt(abs(r2-(t-a)↑2))×ff, t, a+sqrt(r2)×f,
                  x+t2×ff,

comment
```

```
;
                100×abs(a+sqrt(r2)×f-x), 1)
            end pt > 4;
            a := a - (b-yo)×hæld
         end pt > 2;
         xol := x;
         yol := y
      end pt < 6 else
      if pt = 6 then
      begin
         a := next×hcm + xo;
         b := next×hcm + yo
      end pt = 6 else
      begin
         get(REF, free, track);
         i := REF[0];
         j := ident:40 + track;
         get(REF, free, j);
         ident := ident mod 40;
         REF[ident] := i;
         put(REF, free, j);
         kr := i:40;
         cr := i mod 40;
         get(REF, free, kr + 5 + track);
         for i := next while i ≠ bi do
         begin
            i := (i×hcm + x0)×100;
            j := (next×hcm + y0)×100;
            REF[cr] := integer((boolean i shift 20) ∨ boolean j);
            cr := cr + 1;
            if cr > 39 then
            begin
               put(REF, free, kr + 5 + track);
               kr := kr + 1;
               cr := 0
            end
         end;
         put(REF, free, kr + 5 + track);
         get(REF, free, track);
         REF[0] := kr×40 + cr;
         put(REF, free, track);
         go to E

comment
```

```
;
          end pt = 10
       end for pt;
E:     x0 := x;
       y0 := y;
NOLIBRARY:
   end PSYMB;

comment
```

```
;
   procedure XY;
   begin
      integer xcode, ycode, ref, i, j;
      xcode := code:10;
      ycode := code mod 10;
      if xcode = 1 then x0 := x0 + read real else
      if xcode = 3 then x0 := read real else
      if xcode = 5 then
      begin
         j := read integer;
         ref := j:1000;
         i := j mod 1000;
         j := ref:40 + track;
         get(REF, free, j);
         ref := ref mod 40;
         j := REF[ref];
         ref := j mod 40;
         j := j:40;
         ref := ref + i - 1;
A:       if ref > 39 then
         begin
            ref := ref - 40;
            j := j + 1;
            go to A
         end;
         get(REF, free, j + 5 + track);
         i := integer((boolean REF[ref] ∧ 20 m) shift -20);
         x0 := i/100
      end else
      if xcode = 7 then
      x0 := RANDOM(read real, read real);
      if ycode = 2 then y0 := y0 + read real else
      if ycode = 4 then y0 := read real else
      if ycode = 6 then
      begin
         j := read integer;
         ref := j:1000;
         i := j mod 1000;
         j := ref:40 + track;
         get(REF, free, j);
         ref := ref mod 40;

comment
```

```
;
          j := REF[ref];
          ref := j mod 40;
          j := j:40;
          ref := ref + i - 1;
B:        if ref > 39 then
          begin
             ref := ref - 40;
             j := j + 1;
             go to B
          end;
          get(REF, free, j + 5 + track);
          i := integer(boolean REF[ref] ∧ 20 0 20 m);
          y0 := i/100
       end else
       if ycode = 8 then
       y0 := RANDOM(read real, read real);
    end XY;

comment
```

```
;
   real procedure oldRANDOM(from, to);
   value from, to;
   real from, to;
   begin
      integer new, mod;
      mod := 2796203;
      new := 125×oldrand;
      oldrand := new - mod×entier(new/mod);
      oldRANDOM := from + (to - from)×oldrand/mod;
   end oldRANDOM;

comment
```

```
;
   real procedure RANDOM(from, to);
   value from, to;
   real from, to;
   begin
      real rnd,r;
      integer s;
      code rnd;
      3, 45;
      zl 3,mb re1
      nkf 0, grf pa1
      hv r2
e1:   qq 511.9+1023.19+1023.29+1023.39
      e;
      r:=from+(to-from)×rnd;
      RANDOM:=r;
      s:=select(16);
      writecr;
      write(≮-d.dddddd10-dd≯,r);
      select(s);
   end;


comment
```

```
;
   procedure RANSPOT(xbase, ybase);
   real xbase, ybase;
   begin
      integer i, N1, N2;
      real x, y, t, S;
      array C1, C2[0:10];
      real procedure P1;
      begin
         S := 0;
         for i := N1 step -1 until 0 do
         S := S×t + C1[i];
         P1 := (1-t↑2)×S + x;
      end P1;
      real procedure P2;
      begin
         S := 0;
         for i := N2 step -1 until 0 do
         S := S×t + C2[i];
         P2 := (1-t↑2)×S + y;
      end P2;
      N1 := read real;
      for i := 0 step 1 until N1 do
      C1[i] := RANDOM(read real, read real);
      N2 := read real;
      for i := 0 step 1 until N2 do
      C2[i] := RANDOM(read real, read real);
      x := xbase;
      y := ybase;
      PCURVE(P1, P2, t, -1, 1, 1000, 1);
   end RANSPOT;

comment
```

```
;
   procedure setpen(colour,width);
   value colour,width;
   integer colour,width;
   begin
      integer parameter;
      parameter := integer (
         ((boolean colour) shift 20) ∨
         ((boolean width) shift 22));
   code parameter;
   3,44;
   arn pa1, zj
   e;
   end setpen;

comment
```

```
;
   procedure CHANGE(pen, color);
   value pen, color;
   real pen;
   integer color;
   begin
      integer s;
      PEN(0);
      s := select(17);
      writecr;
      writetext(ł<skift til ł);
      if pen < 0.1 then writetext(ł<kuglepenł) else
      begin
         write(ł<dd.dł, pen);
         writetext(ł< mm penł)
      end;
      writecr;
      writetext(case color + 1 of (ł<sortł, ł<rødł, ł<grønł, ł<blaał));
      writetext(ł< farveł);
      setpen(color,pen×10);
      select(s)
   end CHANGE;

comment
```

```
;
   procedure SQUARE(xbase, ybase, B, H);
   value xbase, ybase, B, H;
   real xbase, ybase, B, H;
   begin
      PLINE(xbase, ybase, 0);
      PLINE(xbase, ybase + H, 1);
      PLINE(xbase + B, ybase + H, 1);
      PLINE(xbase +B , ybase, 1);
      PLINE(xbase, ybase, 1);
   end SQUARE;

comment
```

```
;
   procedure ARROW(xbase, ybase, L, t, v);
   value xbase, ybase, L, t, v;
   real xbase, ybase, L, v;
   integer t;
   begin
      integer i;
      real DY, v1, v2, x1, x2, y1, y2;
      if t = 1 then
      begin
         DY := ybase := 0;
         for i := deg step -1 until 0 do
         begin
            ybase := ybase×xbase + COEF[i];
            if i > 0 then DY := DY×xbase + i×COEF[i]
         end for i;
         xbase := A + B×xbase;
         ybase := C + D×ybase;
         v := arctan(D/B×DY) + v×3.141593
      end if t = 1 else
      v := v×3.141593/180;
      v1 := v + 0.3;
      v2 := v - 0.3;
      x1 := xbase + L×cos(v1);
      y1 := ybase + L×sin(v1);
      x2 := xbase + L×cos(v2);
      y2 := ybase + L×sin(v2);
      PLINE(xbase, ybase, 0);
      PLINE(x1, y1, 1);
      PLINE(x2, y2, 0);
      PLINE(xbase, ybase, 1);
   end ARROW;

comment
```

```
;
   procedure CIRCLE(xbase, ybase, R, v1, v2);
   value R, v1, v2;
   real xbase, ybase, R, v1, v2;
   begin
      integer n;
      real v, factor, x, y;
      n := 4 + 2×R×abs(v2 −v1);
      x := xbase;
      y := ybase;
      factor := 3.14159265/180;
      v1 := v1×factor;
      v2 := v2×factor;
      PCURVE(x + R×cos(v), y + R×sin(v), v, v1, v2, n, 1);
      xbase := xbase + R×cos(v2);
      ybase := ybase + R×sin(v2);
   end CIRCLE;

comment
```

```
;
   procedure GRID(xbase, ybase, col, row, dx, dy);
   value xbase, ybase, col, row, dx, dy;
   real xbase, ybase, dx, dy;
   integer row, col;
   begin
      integer i;
      real x1, x2, y1, y2, B, H;
      B := col×dx;
      H := row×dy;
      SQUARE(xbase, ybase, B, H);
      for i := 1 step 1 until col do
      begin
         x1 := xbase + i×dx;
         y1 := y2 := ybase;
         if i mod 2 = 0 then y1 := ybase + H
         else y2 := ybase + H;
         PLINE(x1, y1, 0);
         PLINE(x1, y2, 1);
      end for i;
      PLINE(xbase, ybase, 0);
      for i := 1 step 1 until row do
      begin
         y1 := ybase + i×dy;
         x1 := x2 := xbase;
         if i mod 2 = 0 then x1 := xbase + B
         else x2 := xbase + B;
         PLINE(x1, y1, 0);
         PLINE(x2, y1, 1);
      end for i;
      PLINE(xbase, ybase, 0);
   end GRID;

comment
```

```
;
   procedure ELLIPSE(xbase, ybase, a, b, v0, v1, v2);
   value a, b, v0, v1, v2;
   real xbase, ybase, a, b, v0, v1, v2;
   begin
      integer n;
      real v, factor, dx, dy, cosv0, sinv0, y, x0, y0;
      real procedure x;
      begin
         dx := a×cos(v);
         dy := b×sin(v);
         x := x0 + dx×cosv0 - dy×sinv0;
         y := y0 + dx×sinv0 + dy×cosv0;
      end x;
      n := 4 + (a + b)×abs(v2 - v1);
      x0 := xbase;
      y0 := ybase;
      factor := 3.14159265/180;
      v0 := v0×factor;
      v1 := v1×factor;
      v2 := v2×factor;
      cosv0 := cos(v0);
      sinv0 := sin(v0);
      PCURVE(x, y, v, v1, v2, n, 1);
      v := v2;
      xbase := x;
      ybase := y;
   end ELLIPSE;

comment
```

```
;
   procedure SUPERELLIPSE(xbase, ybase, a, b, power, v0, v1, v2);
   value a, b, power, v0, v1, v2;
   real xbase, ybase, a, b, power, v0, v1, v2;
   begin
      integer n;
      real v, factor, dx, dy, cosv, sinv, cosv0, sinv0, y, x0, y0;
      real procedure x;
      begin
         cosv := cos(v);
         sinv := sin(v);
         dx := a×sign(cosv)×abs(cosv)↑(2/power);
         dy := b×sign(sinv)×abs(sinv)↑(2/power);
         x := x0 + dx×cosv0 - dy×sinv0;
         y := y0 + dx×sinv0 + dy×cosv0;
      end x;
      n := 4 + (a + b)×abs(v2 - v1);
      x0 := xbase;
      y0 := ybase;
      factor := 3.14159265/180;
      v0 := v0×factor;
      v1 := v1×factor;
      v2 := v2×factor;
      cosv0 := cos(v0);
      sinv0 := sin(v0);
      PCURVE(x, y, v, v1, v2, n, 1);
      v := v2;
      xbase := x;
      ybase := y;
   end SUPERELLIPSE;

comment
```

```
;
   procedure POLY1(N, P, x, y, w, a, W);
   value N, P, W;
   integer N, P;
   boolean W;
   array x, y, w, a;
   begin
      integer j, k, n;
      real alfa, beta, XPROD, YPROD, SQ, SQSUM, OLDSQSUM, R, olda;
      array error, orpol, oldorpol[1:N],
      cora[-1:P], oldcora[0:P];
      for n := 1 step 1 until N do
      begin
         error[n] := y[n];
         orpol[n] := 0;
         oldorpol[n] := 1;
      end for n;
      alfa := olda := cora[-1] := 0;
      beta := OLDSQSUM := 1;
      for k := 0 step 1 until P do
      begin
         XPROD := YPROD := SQSUM := 0;
         for n := 1 step 1 until N do
         begin
            error[n] := error[n] - olda×orpol[n];
            R := oldorpol[n]×beta;
            oldorpol[n] := orpol[n];
            R := orpol[n] := R + orpol[n]×(x[n] + alfa);
            if W then R := orpol[n]×w[n];
            SQ := R×orpol[n];
            SQSUM := SQSUM + SQ;
            YPROD := YPROD + R×error[n];
            XPROD := XPROD + SQ×x[n];
         end for n;
         a[k] := olda := YPROD/SQSUM;
         oldcora[k] := 0;
         cora[k] := 1;
         if k > 0 then
         for j := k - 1 step -1 until 0 do
         begin
            R := beta×oldcora[j];
            oldcora[j] := cora[j];

comment
```

```
;
            cora[j] := alfa×oldcora[j] + R + cora[j-1];
            a[j] := a[j] + olda×cora[j];
         end for j;
         beta := -SQSUM/OLDSQSUM;
         OLDSQSUM := SQSUM;
         alfa := - XPROD/SQSUM;
      end for k;
   end POLY1;

comment
```

```
;
   procedure NUMBER(N, digit, x0, y0, H, deci);
   value N, digit, H, deci;
   integer N, digit;
   real x0, y0, H, deci;
   begin
      boolean first;
      integer MOD, symb,i;
      MOD := 1;
      for i := 2 step 1 until digit do MOD := 10×MOD;
      first := true;
      PLINE(x0, y0, 0);
      for i := 1 step 1 until digit do
      begin
         symb := N:MOD;
         N := N mod MOD;
         if symb ≠ 0 then first := false
         else if -, first then symb := 16;
         PSYMB(symb, x0, y0, H, deci);
         MOD := MOD:10
      end for i;
   end NUMBER;
   if where(¢<pinf¢, pinf) ≠ 0 then pinf:=0;
   where(¢<free¢, free);
   REF[0] := 0;
   put(REF, free, 1);
   x0 := y0 := 0;
   deltax := deltay := 0.01;
   xxx := yyy := 0;
   oldrand := 100001;
   oldty := -1;
   track := 1;
   select(64);
   setpen(0,4);
   calcno := read integer;
   size := read integer;
   high := read integer = 1;
   drawno := read integer;
   begin comment frame;
      real B, H, b, h, xbase, ybase, DN;
      integer i, j;
      if high then

comment
```

```
;
      begin
         B := case size + 1 of ( 84.1, 59.4, 42.0, 29.7, 21.0);
         H := case size + 1 of (118.9, 84.1, 59.4, 42.0, 29.7);
      end else
      begin
         B := case size + 1 of (118.9, 84.1, 59.4, 42.0, 29.7);
         H := case size + 1 of ( 84.1, 59.4, 42.0, 29.7, 21.0);
      end not high;
      b := if size = 4 then 8 else if size = 3 then 12.6  else 17;
      h := if size = 4 then 3.5 else if size = 3 then 5.1 else 6.5;
      xbase := B - b;
      ybase := 2;
      select(17);
      writecr;
      writetext(¢<Flyt pennen til højre, saaledes at
der er plads til en tegning af bredden:¢);
      write(¢-ddd¢, H+1);
      writetext(¢< cm¢);
      lyn;
      select(64);
      comment if size ≠ 4 then SQUARE(0, 0, B + 2, H + 2);
      SQUARE(1, 1, B, H);
      SQUARE(2, 2, B - 2, H - 2);
      PLINE(xbase, ybase, 0);
      PLINE(xbase, ybase + h, 1);
      PLINE(xbase + b, ybase + h, 1);
      PLINE(xbase + b, ybase + 0.61×h, 0);
      PLINE(xbase, ybase + 0.61×h, 1);
      x0 := xbase + 0.11×b;
      y0 := ybase + 0.775×h;
      PLINE(x0, y0, 0);
      for i := 184, 177, 163, 180, 166, 169, 0, 147, 166, 167, 146,
      171, 181 do
      PSYMB(i, x0, y0, if size = 4 then 0.3 else 0.4, 90);
      PLINE(xbase + 0.61×b, ybase + h, 0);
      PLINE(xbase + 0.61×b, ybase, 1);
      x0 := xbase + 0.625×b;
      y0 := ybase + 0.538×h;
      PLINE(x0, y0, 0);
      for i := 179, 49, 35, 51, 59, 37, 38 do
      PSYMB(i, x0, y0, 0.2, 90);

comment
```

```
;
      x0 := xbase + 0.625×b;
      y0 := ybase + 0.45×h;
      NUMBER(calcno, 6, x0, y0, 0.2, 90);
      x0 := xbase + 0.625×b;
      y0 := ybase + 0.33×h;
      PLINE(x0, y0, 0);
      for i := 180, 169, 59, 165, 166, 59, 187 do
      PSYMB(i, x0, y0, 0.2, 90);
      x0 := xbase + 0.61×b;
      y0 := ybase + 0.05×h;
      PLINE(x0, y0, 0);
      DN := if size = 4 then 0.6 else 0.7;
      PSYMB(if size = 0 then 16 else size, x0, y0, DN, 70);
      NUMBER(drawno, 5, x0, y0, DN, 70);
   end frame;
AA:ident := read integer;
   if ident < 0 then go to Z;
   at := read integer;
   if at < 1001 then
   begin
      code := read integer;
      XY;
      h := read real;
      hæld := read real;
      if at < 1000 then
      PSYMB(at, x0, y0, h, hæld) else
      begin
         xol := x0;
         for at := lyn while at ≠ 6 do;
         case := 128;
E:       i := lyn;
         if i = 58 ∨ i = 60 then case := (i-58)×64;
         if i = 64 then
         begin
            x0 := xol;
            y0 := y0 - h×1.5;
            PLINE(x0, y0, 0);
            go to E
         end CR;
         i := i + case;
         if i ≠ 135 then

comment
```

```
;
         begin
            PSYMB(i, x0, y0, h, hæld);
            go to E
         end
      end text
   end else
   begin
      at := at - 1000;
      xol := x0;
      yol := y0;
      for l := 1 step 1 until at do
      begin
         ty := read integer;
         if ty < 21 ∨ ty = 27then
         begin
            code := read integer;
            XY;
         end if ty < 21;
         if ty < 6 then
         begin
            if ty < 2 then PLINE(x0, y0, ty) else
            begin
               r2 := (x0-a)↑2 + (y0-b)↑2;
               f := if ty = 3 ∨ ty = 4 then -1 else 1;
               if ty < 4 then
               PCURVE(t, b + sqrt(abs(r2 - (t-a)↑2))×f, t, xol, x0,
               50×abs(xol-x0), 1) else
               begin
                  ff := if y0 > yol then -1 else 1;
                  PCURVE(t, b + sqrt(abs(r2-(t-a)↑2))×ff, t, xol,
                  a + sqrt(r2)×f, 50×abs(xol-a-sqrt(r2)×f), 1);
                  PCURVE(t, b-sqrt(abs(r2-(t-a)↑2))×ff, t,
                  a + sqrt(r2)×f, x0, 50×abs(a + sqrt(r2)×f - x0), 1)
               end
            end;
            xol := x0;
            yol := y0
         end else
         if ty = 6 then
         begin
            a := x0;

comment
```

```
;
            b := y0
         end else
         if ty = 10 then
         begin
            get(REF, free, track);
            f := REF[0];
            ff := ident:40 + track;
            get(REF, free, ff);
            ty := ident mod 40;
            REF[ty] := f;
            put(REF, free, ff);
            ff := f:40;
            f := f mod 40;
            get(REF, free, ff+5+track);
            i := x0×100;
            ty := y0×100;
            REF[f] := integer((boolean i shift 20) ∨ boolean ty);
            f := f + 1;
            put(REF, free, ff+5+track);
            if f > 39 then
            begin
               f := 0;
               ff := ff + 1;
            end;
            get(REF, free, track);
            REF[0] := ff×40 + f;
            put(REF, free, track);
         end if ty = 10 else
         case ty - 10 of
         begin
            SQUARE(x0, y0, read real, read real);
            CIRCLE(x0, y0, read real, read real, read real);
            GRID(x0, y0, read integer, read integer, read real, read real);
            ELLIPSE(x0, y0, read real, read real, read real, read real,
            read real);
            PDASH(x0, y0, read real, read real);
            ARROW(x0, y0, read real, read integer, read real);
            RANSPOT(x0,y0);
            i := 18; i := 19; i := 20;
            begin comment ty = 21;
               A := read real;

comment
```

```
;
                B := read real;
                C := read real;
                D := read real;
             end ty = 21;
             begin comment ty = 22;
                points := read integer;
                for i := 1 step 1 until points do
                X[i] := read real;
             end ty = 22;
             begin comment ty = 23;
                points := read integer;
                for i := 1 step 1 until points do
                Y[i] := read real;
             end ty = 23;
             begin comment ty = 24;
                deg := read integer;
                POLY1(points, deg, X, Y, X, COEF, false);
             end ty = 24;
             begin comment ty = 25;
                integer step, j;
                real x, X1, X2, P, X;
                real procedure y;
                begin
                   P := 0;
                   for j := deg step -1 until 0 do
                   P := P×X + COEF[j];
                   y := C + D×P;
                end y;
                step := read integer;
                X1 := read real;
                X2 := read real;
                PCURVE(A + B×X, y, X, X1, X2, step, 1);
             end ty = 25;
             CHANGE(read real, read integer);
             SUPERELLIPSE(x0, y0, read real, read real, read real, read real,
             read real, read real);
          end case;
       end for l
    end;
    go to AA;
Z: PEN(0);

comment
```

```
;
end;
move,work,free<
res,s0.0.0.0,plot2<
setsum,plot2<
algol,s<
begin
   message compile read pinf;
   comment

   Input format:

   maxsymb

   symb, nbit, scalefactor
   ...
   -1
   ;
   integer free, maxsymb, bit, pos, bi, list, track, symb, f, pt;
   integer i, x;
   boolean aa;
   integer array REF[0:39];
   boolean array A[0:39];
   array A1[0:39];

comment
```

```
;
   procedure nextlist;
   begin
      integer i;
      pos := bit;
      list := list+1;
      i := list mod 40;
      if i = 0 then
      begin
         put(A, free, track);
         track := track+1
      end new track;
      A[i] := 40 0
   end nextlist;

comment
```

```
;
   procedure store(v);
   value v;
   integer v;
   begin
      integer i;
      pos := pos+bit;
      if pos > 40 then nextlist;
      i := list mod 40;
      A[i] := A[i] ∨ ((boolean v)shift -pos)
   end store;

comment
```

```
;
   procedure storeid(symb, aa);
   value symb, aa;
   integer symb;
   boolean aa;
   begin
      boolean array A[0:39];
      integer i,k;
      k := symb : 40+2;
      i := symb mod 40;
      get(A, free, k);
      A[i] := aa;
      put(A, free, k)
   end storeid;

   cancel(‡<pinf‡);
   where(‡<free‡, free);
   select(16);
   maxsymb := read integer;
   if maxsymb > 127 then maxsymb := maxsymb-63;
   list := 0;
   A[0] := 40 0;
   track := (maxsymb+39):40+2;
   for i:=1 step 1 until 39 do
   A1[i] := 0.0;
   A1[0] := real track;

   for symb := read integer while symb ≥ 0 do
   begin
      if symb > 127 then symb := symb-63;
      bit := read integer;
      f := read integer;
     writecr;
      write(‡dddd‡, symb);
      write(‡dddd‡, bit, f); *
      for i:=1 step 1 until 39 do
      begin
         if A1[i]=f then goto F1;
         if A1[i]=0 then
         begin
            A1[i]:=f;
            goto F1

comment
```

```
;
         end new factor
      end for i;
F1:   f := i;
      aa := (boolean list) shift 10;
      aa := aa ∨ ((boolean bit) shift 5);
      aa := aa ∨ boolean f;
      if symb=0 then
      begin
         for i := 0 step 1 until maxsymb do
         storeid(i, aa)
      end
      else
      storeid(symb, aa);
      bi := (integer (40 1 shift bit))-1;
      if bit=40 then bi := integer 40 m;
      pos := 0;
      for pt := read integer while pt≥0 do
      begin
         store(pt);
         if pt<10 then
         begin
            store(read integer);
            store(read integer)
         end
         else
         begin
            for x := read integer while x≥0 do
            begin
               store(x);
               store(read integer)
            end store reference;
            goto E
         end pt=10
      end for pt;
E:    store(bi);
      nextlist
   end symb;
   put(A, free, track);
   for i:=1 step 1 until 39 do
   if A1[i] ≠ 0 then A1[i] := 1/A1[i];
   put(A1, free, 1);

comment
```

```
;
   reserve(‡<pinf‡, track);
   code i;
   1, 44;
   hs 1
   hv re1
   tsetsum;
   tpinf;
   qqf,
e1: e
end;
run<
Ff
304
0, 10, 360
0, 240, 0
-1
167P 10, 360
0, 0, 120
1, 0, 480
1, 135, 480
6, 135, 405
5, 135, 330
1, 0, 330
0, 360, 0
-1
163L 10, 360
0, 0, 480
1, 0, 120
1, 160, 120
0, 305, 0
-1
147T 10, 360
0, 110, 120
1, 110, 480
0, 0, 480
1, 220, 480
0, 305, 0
-1
181E 10, 360
0, 0, 120
1, 0, 480

comment
```

```
;
1, 210, 480
0, 0, 300
1, 160, 300
0, 0, 120
1, 210, 120
0, 360, 0
-1
169R 10, 360
0, 0, 120
1, 0, 480
1, 135, 480
6, 135, 405
5, 135, 330
1, 0, 330
0, 110, 330
1, 235, 120
0, 330, 0
-1
184H 10, 360
0, 0, 120
1, 0, 480
0, 200, 480
1, 200, 120
0, 0, 300
1, 200, 300
0, 360, 0
-1
177A 10, 360
0, 0, 120
1, 125, 480
1, 250, 120
0, 40, 245
1, 210, 245
0, 400, 0
-1
14_20, 360
0, 0, 60
1, 225, 60
0, 0, 0
-1
171Ø 10, 360
```

comment

```
;
0, 0, 220
1, 0, 380
6, 100, 380
2, 200, 380
1, 200, 220
6, 100, 220
3, 0, 220
0, 0, 120
1, 200, 480
0, 360, 0
-1
149V 10, 360
0, 0, 480
1, 125, 120
1, 250, 480
0, 360, 0
-1
152Y 10, 360
0, 125, 120
1, 125, 225
1, 0, 480
0, 125, 225
1, 250, 480
0, 370, 0
-1
161J 10, 360
0, 0, 220
6, 100, 220
3, 100, 120
6, 100, 195
3, 175, 195
1, 175, 480
0, 310, 0
-1
180D 10, 360
0, 0, 120
1, 0, 480
1, 100, 480
6, 100, 380
2, 200, 380
1, 200, 220
```

comment

```
;
6, 100, 220
3, 100, 120
1, 0, 120
0, 360, 0
-1
165N 10, 360
0, 0, 120
1, 0, 480
1, 200, 120
1, 200, 480
0, 360, 0
-1
185I 10, 360
0, 0, 120
1, 0, 480
0, 150, 0
-1
166O 10, 360
0, 0, 220
1, 0, 380
6, 100, 380
2, 200, 380
1, 200, 220
6, 100, 220
3, 0, 220
0, 360, 0
-1
168Q 10, 360
0, 0, 220
1, 0, 380
6, 100, 380
2, 200, 380
1, 200, 220
6, 100, 220
3, 0, 220
0, 100, 220
1, 200, 120
0, 360, 0
-1
183G 10, 360
0, 200, 380
```

comment

```
;
6, 100, 380
2, 0, 380
1, 0, 220
6, 100, 220
3, 200, 220
1, 200, 300
1, 75, 300
0, 360, 0
-1
164M 10, 360
0, 0, 120
1, 0, 480
1, 125, 245
1, 250, 480
1, 250, 120
0, 410, 0
-1
178B 10, 360
0, 0, 120
1, 0, 480
1, 135, 480
6, 135, 405
5, 135, 330
1, 0, 330
0, 105, 330
6, 105, 225
5, 105, 120
1, 0, 120
0, 360, 0
-1
148U 10, 360
0, 0, 480
1, 0, 220
6, 100, 220
3, 200, 220
1, 200, 480
0, 300, 0
-1
182F 10, 360
0, 0, 120
1, 0, 480
```

comment

```
;
1, 210, 480
0, 0, 300
1, 160, 300
0, 360, 0
-1
146S 10, 360
0, 0, 195
6, 75, 195
3, 75, 120
1, 150, 120
6, 150, 220
5, 150, 320
1, 80, 320
6, 80, 400
4, 80, 480
1, 155, 480
6, 155, 400
5, 235, 400
0, 360, 0
-1
162K 10, 360
0, 0, 120
1, 0, 480
0, 0, 230
1, 250, 480
0, 70, 300
1, 250, 120
0, 360, 0
-1
179C 10, 360
0, 200, 220
6, 100, 220
3, 0, 220
1, 0, 380
6, 100, 380
2, 200, 380
0, 360, 0
-1
150W 10, 360
0, 0, 480
1, 90, 120
```

comment

```
;
1, 180, 480
1, 270, 120
1, 360, 480
0, 480, 0
-1
151X 10, 360
0, 0, 120
1, 250, 480
0, 0, 480
1, 250, 120
0, 360, 0
-1
49a 10, 360
0, 0, 190
6, 70, 190
3, 140, 190
1, 140, 290
6, 70, 290
2, 0, 290
1, 0, 190
0, 140, 360
1, 140, 225
1, 210, 120
0, 340, 0
-1
39p 10, 360
0, 0, 190
6, 70, 190
3, 140, 190
1, 140, 290
6, 70, 290
2, 0, 290
0, 0, 360
1, 0, 0
0, 340, 0
-1
41r 10, 360
0, 0, 120
1, 0, 360
0, 0, 290
6, 70, 290
```

comment

```
;
2, 140, 290
0, 260, 0
-1
37n 10, 360
0, 0, 120
1, 0, 360
0, 0, 290
6, 70, 290
2, 140, 290
1, 140, 120
0, 340, 0
-1
38o 10, 360
0, 0, 190
6, 70, 190
3, 140, 190
1, 140, 290
6, 70, 290
2, 0, 290
1, 0, 190
0, 340, 0
-1
51c 10, 360
0, 140, 190
6, 70, 190
3, 0, 190
1, 0, 290
6, 70, 290
2, 140, 290
0, 340, 0
-1
35l 10, 360
0, 0, 480
1, 0, 140
6, 20, 140
3, 40, 120
0, 160, 0
-1
57i 10, 360
0, 0, 470
6, 10, 470
```

comment

```
;
2, 20, 470
3, 0, 470
0, 0, 360
1, 0, 140
6, 20, 140
3, 40, 120
0, 160, 0
-1
16, 10, 360
0, 0, 195
1, 0, 405
6, 75, 405
2, 150, 405
1, 150, 195
6, 75, 195
3, 0, 195
0, 310, 0
-1
1, 10, 360
0, 0, 340
1, 50, 480
1, 50, 120
0, 190, 0
-1
2, 10, 360
0, 0, 405
6, 75, 405
2, 150, 405
1, 0, 120
1, 150, 120
0, 310, 0
-1
8, 10, 360
0, 100, 120
6, 100, 220
4, 100, 320
6, 100, 400
4, 100, 480
5, 100, 320
6, 100, 220
5, 100, 120
```

comment

```
;
0, 310, 0
-1
3, 10, 360
0, 0, 220
6, 100, 220
3, 200, 220
2, 100, 320
6, 100, 400
5, 100, 480
6, 100, 400
2, 20, 400
0, 310, 0
-1
4, 10, 360
0, 150, 120
1, 150, 300
0, 170, 220
1, 0, 220
1, 170, 480
0, 310, 0
-1
5, 10, 360
0, 0, 195
6, 75, 195
3, 150, 195
1, 150, 270
6, 75, 270
2, 0, 270
1, 0, 480
1, 150, 480
0, 310, 0
-1
6, 10, 360
0, 150, 405
6, 75, 405
2, 0, 405
1, 0, 195
6, 75, 195
3, 150, 195
1, 150, 270
6, 75, 270
```

comment

```
;
2, 0, 270
0, 310, 0
-1
9, 10, 360
0, 0, 195
6, 75, 195
3, 150, 195
1, 150, 405
6, 75, 405
2, 0, 405
1, 0, 330
6, 75, 330
3, 150, 330
0, 310, 0
-1
7, 10, 360
0, 0, 120
1, 150, 480
1, 0, 480
1, 0, 425
0, 310, 0
-1
59point 10, 360
0, 0, 130
6, 10, 130
3, 20, 130
2, 0, 130
0, 200, 0
-1
187colon 10, 360
0, 0, 130
6, 10, 130
3, 20, 130
2, 0, 130
0, 0, 330
6, 10, 330
3, 20, 330
2, 0, 330
0, 200, 0
-1
32minus 10, 360
```

comment

```
;
0, 0, 230
1, 230, 230
0, 360, 0
-1
160plus 10, 360
0, 0, 230
1, 230, 230
0, 115, 0
1, 115, 345
0, 360, 0
-1
58lowercase 10, 360
0, 0, 0
-1
186LOWERCASE 10, 360
0, 0, 0
-1
60uppercase 10, 360
0, 0, 0
-1
188UPPERCASE 10, 360
0, 0, 0
-1
240, 10, 360
0, 0, 0
1, 72, 0
1, 72, 20
1, 0, 20
1, 0, 0
0, 9, 20
1, 9, 340
0, 0, 340
1, 72, 340
1, 72, 360
1, 0, 360
1, 0, 340
0, 63, 340
1, 63, 20
0, 0, 0
10
36, 360
```

comment

```
;
72, 350
72, 10
36, 0
0, 10
0, 350
-1
251, 10, 360
0, 0, 0
1, 0, 215
1, 658, 215
1, 658, 0
1, 0, 0
0, 110, 0
1, 110, 260
1, 216, 324
1, 442, 324
1, 548, 260
1, 548, 0
0, 548, 43
1, 110, 43
0, 110, 86
1, 548, 86
0, 548, 129
1, 110, 129
0, 110, 172
1, 548, 172
0, 0, 0
10
329, 324
658, 108
494, 0
329, 0
165, 0
0, 108
-1
199, 10, 360
0, 87, 0
1, 87, 244
1, 0, 360
0, 87, 244
1, 174, 360
```

comment

```
;
0, 0, 0
10
87, 360
-1
200, 10, 360
0, 0, 8
1, 0, 352
6, 58, 180
2, 116, 352
1, 116, 8
3, 0, 8
0, 0, 0
10
58, 360
116, 236
116, 180
116, 124
58, 0
0, 124
0, 180
0, 236
-1
210, 10, 360
0, 40, 0
6, 397, 180
4, 40, 360
1, 754, 360
5, 754, 0
1, 40, 0
0, 0, 0
10
218, 360
575, 360
397, 0
-1
230, 10, 360
0, 0, 14
1, 0, 346
6, 71, 180
2, 142, 346
1, 142, 14
```

comment

```
;
3, 0, 14
0, 0, 46
1, 142, 314
1, 0, 314
1, 142, 46
1, 0, 46
0, 0, 0
10
71, 360
71, 0
-1
250, 10, 360
0, 0, 0
1, 0, 180
1, 180, 360
1, 360, 360
1, 540, 180
1, 540, 0
1, 0, 0
0, 0, 36
1, 540, 36
0, 540, 72
1, 0, 72
0, 0, 108
1, 540, 108
0, 540, 144
1, 0, 144
0, 0, 180
1, 540, 180
0, 0, 0
10
0, 90
540, 90
-1
260, 10, 360
0, 0, 280
1, 0, 360
1, 192, 360
1, 192, 280
1, 0, 280
0, 42, 280
```

comment

```
;
1, 42, 0
1, 150, 0
1, 150, 280
0, 0, 0
10
96, 360
192, 320
96, 0
0, 320
-1
270, 10, 360
0, 0, 0
1, 144, 0
1, 144, 360
1, 0, 360
1, 0, 0
10
72, 360
144, 270
144, 180
144, 90
72, 0
0, 90
0, 180
-1
275, 10, 360
0, 0, 0
1, 0, 360
1, 108, 360
1, 108, 0
1, 0, 0
0, 0, 76
1, 108, 284
1, 0, 284
1, 108, 76
1, 0, 76
0, 0, 0
10
54, 360
108, 180
54, 0
```

comment

```
;
0, 180
-1
280, 10, 360
0, 0, 0
1, 0, 180
1, 180, 180
1, 180, 360
1, 360, 360
1, 360, 180
1, 540, 180
1, 540, 0
1, 0, 0
10
360, 270
180, 270
-1
300, 10, 360
0, 0, 180
6, 180, 180
3, 360, 180
2, 0, 180
0, 0, 0
10
180, 360
360, 180
180, 0
0, 180
-1
301, 10, 360
0, 0, 180
6, 180, 180
3, 360, 180
2, 0, 180
1, 70, 180
1, 70, 290
1, 290, 70
1, 290, 180
1, 360, 180
0, 0, 0
10
180, 360
```

comment

```
;
360, 180
180, 0
0, 180
-1
302, 10, 360
0, 0, 180
6, 180, 180
3, 360, 180
2, 0, 180
1, 70, 180
1, 70, 70
1, 290, 290
1, 290, 180
1, 360, 180
0, 0, 0
10
180, 360
360, 180
180, 0
0, 180
-1
303, 10, 360
0, 0, 180
6, 180, 180
3, 360, 180
2, 0, 180
0, 180, 360
1, 180, 290
1, 70, 290
1, 290, 70
1, 180, 70
1, 180, 0
0, 0, 0
10
180, 360
360, 180
180, 0
0, 180
-1
304, 10, 360
0, 0, 180
```

comment

```
;
6, 180, 180
3, 360, 180
2, 0, 180
0, 180, 360
1, 180, 290
1, 290, 290
1, 70, 70
1, 180, 70
1, 180, 0
0, 0, 0
10
180, 360
360, 180
180, 0
0, 180
-1
-1

w,outparam,done,read,pinf<
compress<
algol<
Clean free and buffer, incl. marker bits.
begin
   begin
      integer FREE, track, ntrack, index;
      integer array A[0:39];
      boolean code1;
      core code code1,A,index;
      3, 46;
      3, 48;
      3, 44;
      grn c17          M       ; UV:=0 incl marks;
      arn pa2, ar pa3
      us  0  , hr  s1
      e;


      where(¢<free¢, FREE);

      ntrack := integer(((boolean FREE)shift -16)∧24 0 16 m);


comment
```

```
;
      select(17);
      writecr;
      writeinteger(‹p›,ntrack);

      for index:=0 step 1 until 39 do gier(code1);
      for track:=1 step 1 until ntrack do
      begin
         put(A, FREE, track)
      end track
   end clean free;
   begin
      integer array A[0:4095];
      integer index;
      boolean code2;
      core code code2,A,index;
      3, 46; code2
      3, 48; A
      3, 44; index;
      grn c17          M       ; UV:=0 incl marks;
      arn pa2, ar pa3
      us  0  , hr  s1
      e;
      for index:=0 step 1 until 4095 do gier(code2);
   end clean buffer;
end;
run<
w,outparam,done,clean<
```