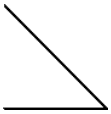
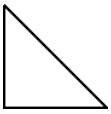
Métapost : exemples

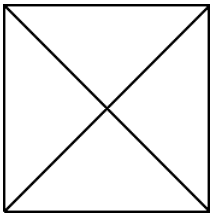
```
input macros;
verbatimtex
\documentclass[12pt]{article}
\usepackage[T1]{fontenc}
\begin{document}
etex
```



```
beginfig(1)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A--B--C;

endfig;
```

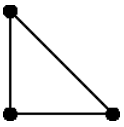


```
beginfig(2)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A--B--C--cycle;

endfig;
```

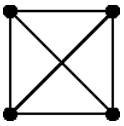


```
beginfig(3)
  pair A[];
  A[0]:=(-1cm, -1cm);
  A[1]:=( 1cm, -1cm);
  A[2]:=( 1cm,  1cm);
  A[3]:=(-1cm,  1cm);
  draw A[0]--A[1]--A[2]--A[3]--cycle;
  draw A[0]--A[2];
  draw A[1]--A[3];
endfig;
```

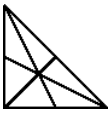
```
beginfig(4)
    pair A;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A withpen pencircle scaled 4bp;

endfig;
```



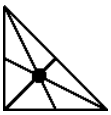
```
beginfig(5)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  draw A withpen pencircle scaled 4bp;
  draw B withpen pencircle scaled 4bp;
  draw C withpen pencircle scaled 4bp;
endfig;
```



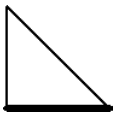
```
beginfig(6)
  pair A, B, C, D;
  A:=(0,0); B:=(1cm,0);
  C:=(1cm,1cm); D:=(0,1cm);
  draw A--B--C--D--cycle;
  draw A--C;
  draw B--D;
  draw A withpen pencircle scaled 4bp;
  draw B withpen pencircle scaled 4bp;
  draw C withpen pencircle scaled 4bp;
  draw D withpen pencircle scaled 4bp;
endfig;
```



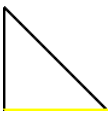
```
beginfig(7)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  draw 1/2[A,B] -- C;
  draw 1/2[B,C] -- A;
  draw 1/2[C,A] -- B;
endfig;
```



```
beginfig(8)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  draw 1/2[A,B] -- C;
  draw 1/2[B,C] -- A;
  draw 1/2[C,A] -- B;
  draw 1/3 A + 1/3 B + 1/3 C
       withpen pencircle scaled 4bp;
endfig;
```



```
beginfig(9)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A--B--C--cycle;
    draw A--B withpen pencircle scaled 2bp;

endfig;
```

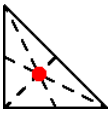


```
beginfig(10)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A--B--C--cycle;
    draw A--B withcolor (green + red);

endfig;
```

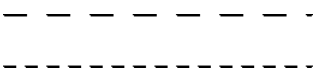
```
beginfig(11)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
```

```
    draw A--B;
    draw B--C dashed evenly;
    draw C--A dashed withdots;

endfig;
```

```
beginfig(12)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    draw A--B withpen pencircle scaled 2bp withcolor .8white;
    draw B--C withpen pencircle scaled 2bp withcolor .6white;
    draw C--A withpen pencircle scaled 2bp withcolor .4white;

endfig;
```



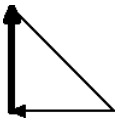
```
beginfig(13)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  draw 1/2[A,B] -- C dashed evenly;
  draw 1/2[B,C] -- A dashed evenly;
  draw 1/2[C,A] -- B dashed evenly;
  draw 1/3 A + 1/3 B + 1/3 C
       withpen pencircle scaled 4bp
       withcolor red;
endfig;
```



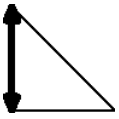
```
beginfig(14)
    draw (0,0)--(3cm,0) dashed evenly scaled 2;
    draw (0,-5mm)--(3cm,-5mm) dashed evenly;

endfig;
```

```
beginfig(15)
    draw (0,0)--(3cm,0)
      dashed dashpattern(on 2bp off 3bp);

endfig;
```

```
beginfig(16)
  draw (0,0)--(3cm,0) dashed dashpattern(on 1bp off 2bp on 10bp off 2bp);
endfig;
```

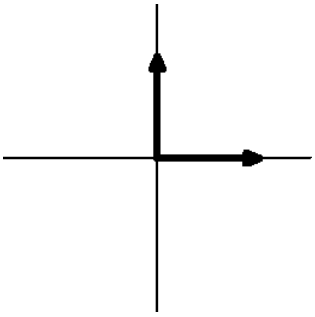


```
beginfig(17)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    drawarrow C--B--A;
    drawarrow A--C withpen pencircle scaled 2bp;

endfig;
```



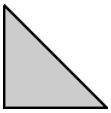
```
beginfig(18)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
```

```
    draw C--B--A--cycle;
    drawdblarrow A--C withpen pencircle scaled 2bp;

endfig;
```

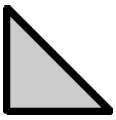


```
beginfig(19)
  draw (-1.5cm,0)--(1.5cm,0);
  draw (0,-1.5cm)--(0,1.5cm);
  drawarrow (0,0)--(1cm,0)
            withpen pencircle scaled 2bp;
  drawarrow (0,0)--(0,1cm)
            withpen pencircle scaled 2bp;
endfig;
```

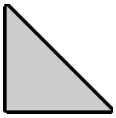
```
beginfig(20)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    fill A--B--C--cycle withcolor .8 white;

endfig;
```



```
beginfig(21)
    pair A, B, C;
    A:=(0,0); B:=(1cm,0); C:=(0,1cm);
    fill A--B--C--cycle withcolor .8 white;
    draw A--B--C--cycle;

endfig;
```

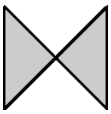


```
beginfig(22)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  fill A--B--C--cycle withcolor .8 white;
  draw A--B--C--cycle withpen pencircle scaled 2bp;
endfig;
```

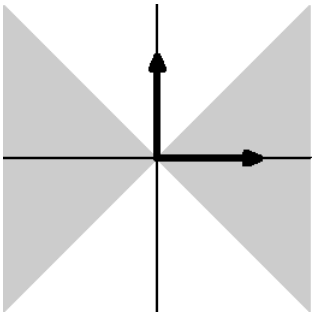


```
beginfig(23)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle withpen pencircle scaled 2bp;
  fill A--B--C--cycle withcolor .8 white;
endfig;
```

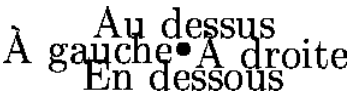
```
beginfig(24)
  pair A, B, C, D;
  A:=(0,0); B:=(1cm,0);
  C:=(1cm,1cm); D:=(0,1cm);
  fill A--C--B--D--cycle withcolor .8white;
endfig;
```



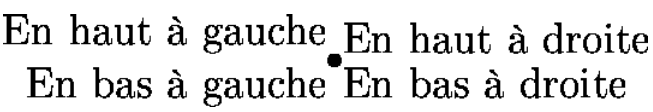
```
beginfig(25)
  pair A, B, C, D;
  A:=(0,0); B:=(1cm,0);
  C:=(1cm,1cm); D:=(0,1cm);
  path p;
  p := A--C--B--D--cycle;
  fill p withcolor .8white;
  draw p;
endfig;
```



```
beginfig(26)
  pair A, B, C, D;
  A:=(-1.5cm,-1.5cm); B:=(1.5cm,-1.5cm);
  C:=(1.5cm,1.5cm);   D:=(-1.5cm,1.5cm);
  fill A--C--B--D--cycle withcolor .8white;
  draw (-1.5cm,0)--(1.5cm,0);
  draw (0,-1.5cm)--(0,1.5cm);
  drawarrow (0,0)--(1cm,0)
            withpen pencircle scaled 2bp;
  drawarrow (0,0)--(0,1cm)
            withpen pencircle scaled 2bp;
endfig;
```

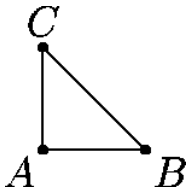



```
beginfig(27)
    pair A;
    A := (0,0);
    draw A withpen pencircle scaled 4bp;
    label.top(btex Au dessus  etex, A);
    label.bot(btex En dessous etex, A);
    label.rt (btex ^^c0 droite   etex, A);
    label.lft(btex ^^c0 gauche   etex, A);

endfig;
```

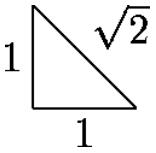



```
beginfig(28)
    pair A;
    A := (0,0);
    draw A withpen pencircle scaled 4bp;
    label.ulft(btex En haut ^^e0 gauche etex, A);
    label.urt (btex En haut ^^e0 droite etex, A);
    label.llft(btex En bas ^^e0 gauche  etex, A);
    label.lrt (btex En bas ^^e0 droite  etex, A);

endfig;
```

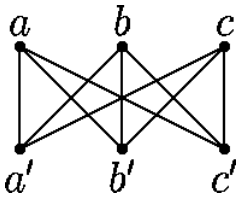

```
beginfig(29)
    pair A;
    A := (0,0);
    dotlabel.urt(btex $A$ etex, A);

endfig;
```




```
beginfig(30)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  dotlabel.llft(btex $A$ etex, A);
  dotlabel.lrt(btex $B$ etex, B);
  dotlabel.top(btex $C$ etex, C);
endfig;
```

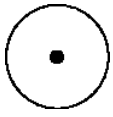



```
beginfig(31)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  label.bot(btex $1$ etex, 1/2[A,B]);
  label.lft(btex $1$ etex, 1/2[A,C]);
  label.urt(btex $\sqrt 2$ etex, 1/2[B,C]);
endfig;
```



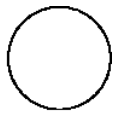

```
beginfig(32)
  u:=1cm;
  pair A,B,C,D,E,F,G;
  A := (-u,u);
  B := (0,u);
  C := (u,u);
  D := (-u,0);
  E := (0,0);
  F := (u,0);
  draw A--D; draw A--E; draw A--F;
  draw B--D; draw B--E; draw B--F;
  draw C--D; draw C--E; draw C--F;
  dotlabel.top(btex $a$ etex, A);
  dotlabel.top(btex $b$ etex, B);
  dotlabel.top(btex $c$ etex, C);
  dotlabel.bot(btex $a'$ etex, D);
  dotlabel.bot(btex $b'$ etex, E);
  dotlabel.bot(btex $c'$ etex, F);
endfig;
```

```
beginfig(33)
    draw fullcircle;

endfig;
```

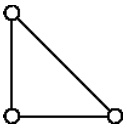


```
beginfig(34)
```

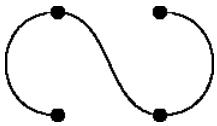
```
    draw (0,0) withpen pencircle scaled 4bp;
    draw fullcircle scaled 1cm;

endfig;
```




```
beginfig(35)
    draw (0,0) withpen pencircle scaled 4bp;
    draw fullcircle scaled 1cm shifted (1cm,1cm);

endfig;
```

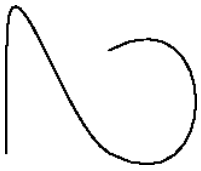


```
beginfig(36)
  pair A, B, C;
  A:=(0,0); B:=(1cm,0); C:=(0,1cm);
  draw A--B--C--cycle;
  fill fullcircle scaled 4bp shifted A withcolor white;
  fill fullcircle scaled 4bp shifted B withcolor white;
  fill fullcircle scaled 4bp shifted C withcolor white;
  draw fullcircle scaled 4bp shifted A;
  draw fullcircle scaled 4bp shifted B;
  draw fullcircle scaled 4bp shifted C;
endfig;
```

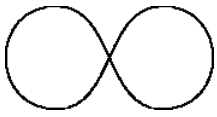


```
beginfig(37)
    draw (0,0) .. (0,1cm) .. (1cm,0) .. (1cm,1cm);
    draw (0,0)     withpen pencircle scaled 4bp;
    draw (0,1cm)   withpen pencircle scaled 4bp;
    draw (1cm,0)   withpen pencircle scaled 4bp;
    draw (1cm,1cm) withpen pencircle scaled 4bp;

endfig;
```

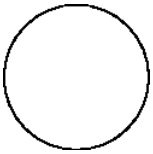
```
beginfig(38)
    draw (0,0) -- (0,1cm) .. (1cm,0) .. (1cm,1cm);

endfig;
```



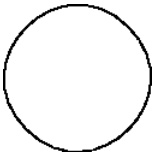
```
beginfig(39)
    draw (0,0) --- (0,1cm) .. (1cm,0) .. (1cm,1cm);

endfig;
```



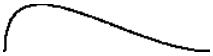
```
beginfig(40)
    draw (0,0) .. (0,1cm) .. (1cm,0) .. (1cm,1cm)
         .. cycle;

endfig;
```



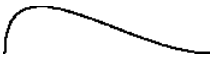
```
beginfig(41)
  pair A, B, C, D;
  A:=(0,0); B:=(1cm,0);
  C:=(1cm,1cm); D:=(0,1cm);
  draw A..B..C..D..cycle;
endfig;
```



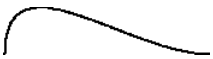
```
beginfig(42)
  draw (0,0) .. (1cm,1cm) .. cycle;
endfig;
```



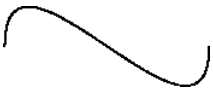
```
beginfig(43)
    draw (0,0){(0,1)} .. (2cm,0){(1,0)};

endfig;
```



```
beginfig(44)
    draw (0,0){dir 90} .. (2cm,0){dir 0};

endfig;
```

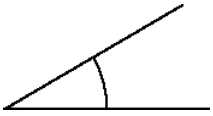


```
beginfig(45)
    draw (0,0){up} .. (2cm,0){right};

endfig;
```



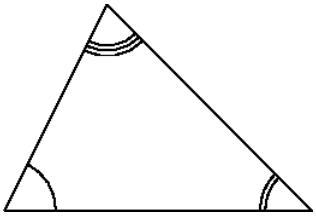
```
beginfig(46)
  draw (0,0){up} .. (2cm,0){up};
endfig;
```

```
beginfig(47)
  draw (0,0){up} .. (2cm,0){up} .. cycle;
endfig;
```



```
beginfig(48)
  alpha := 30;
  draw (0,0) -- 2cm*dir 0;
  draw (0,0) -- 2cm*dir alpha;
  draw 1cm * dir 0 {dir 90} ..
       1cm * dir alpha {dir(90+alpha)};
endfig;
```

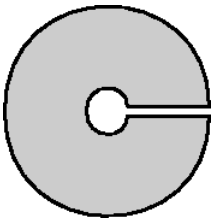
```
beginfig(49)
  alpha := 30;
  draw (0,0) -- 2cm*dir 0;
  draw (0,0) -- 2cm*dir alpha;
  draw 1cm * dir 0 {dir 90} ..
       1cm * dir alpha {dir(90+alpha)};
  draw 1.1cm * dir 0 {dir 90} ..
       1.1cm * dir alpha {dir(90+alpha)};
endfig;
```



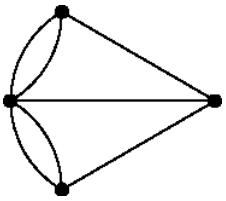
```
beginfig(50)
      def draw_angle(expr A, O, B, n) =
    draw_angle_(A,O,B,5mm);
    if n>1 : draw_angle_(A,O,B,4.5mm); fi;
    if n>2 : draw_angle_(A,O,B,4mm);   fi;
    if n>3 : draw_angle_(A,O,B,5.5mm); fi;
  enddef;
  def draw_angle_(expr A,O,B,d) =
    draw (O + d*unitvector(A-O))
         { d*unitvector(A-O) rotated 90 }
         ..
         { d*unitvector(B-O) rotated 90 }
         (O + d*unitvector(B-O));
  enddef;
  pair A, B, C;
  A := (0,0);
  B := (3cm,0);
  C := (1cm,2cm);
  draw A--B--C--cycle;
  draw_angle(B,A,C,1);
  draw_angle(C,B,A,2);
  draw_angle(A,C,B,3);
endfig;
```

```
beginfig(51)
  u:=1cm;
  path p;
  p := (u,0){up}..(-u,0){down}--cycle;
  fill p withcolor .8*white;
  draw p withpen pencircle scaled 1bp;
endfig;
```

```
beginfig(52)
  u:=1cm;
  path p;
  p := (u,0){up} ..
       (-u,0){down} --
       (-.2u,0){up} ..
       (.2u,0){down} --
       cycle;
  fill p withcolor .8*white;
  draw p withpen pencircle scaled 1bp;
endfig;
```



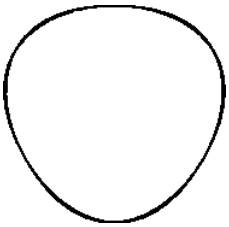
```
beginfig(53)
  u:=1cm;
  pair A,AA,B,BB,C,D;
  A  := ( .2u,  .05u);
  AA := ( .2u, -.05u);
  B  := (   u,  .05u);
  BB := (   u, -.05u);
  C  := (-.2u,     0);
  D  := (  -u,     0);
  path p;
  p:= B{up}    .. D{down} .. BB{up}  --
      AA{down} .. C{up}   .. A{down} -- cycle;
  fill p withcolor .8*white;
  draw p withpen pencircle scaled 1bp;
endfig;
```



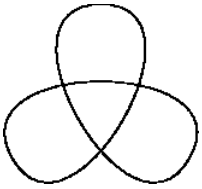
```
beginfig(54)
  u:=1cm;
  pair A,B,C,D;
  A:=(0,0);
  B:=u*dir 60;
  C:=u*dir -60;
  D:=(2u,0);
  draw A withpen pencircle scaled 4bp;
  draw B withpen pencircle scaled 4bp;
  draw C withpen pencircle scaled 4bp;
  draw D withpen pencircle scaled 4bp;
  draw A--D--B--D--C;
  draw A{up}..B;
  draw A{down}..C;
  draw A{dir 30}..B;
  draw A{dir -30}..C;
endfig;
```

```
beginfig(55)
    draw (0,0) .. (1cm,1cm) .. (2cm,0);

endfig;
```

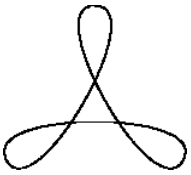
```
beginfig(56)
    draw (0,0) .. tension 2 ..
         (1cm,1cm) .. (2cm,0);

endfig;
```



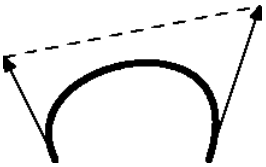
```
beginfig(57)
  draw (0,u)         {right}    ..
       ( u*dir -150 ){dir 120}  ..
       ( u*dir -30  ){dir -120} ..
  cycle;
endfig;
```



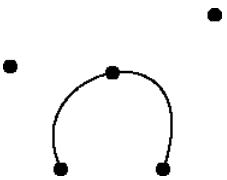
```
beginfig(58)
  draw (0,u)         {right}    .. tension 2 ..
       ( u*dir -150 ){dir 120}  .. tension 2 ..
       ( u*dir -30  ){dir -120} .. tension 2 ..
  cycle;
endfig;
```



```
beginfig(59)
  draw (0,u)         {right}    .. tension 4 ..
       ( u*dir -150 ){dir 120}  .. tension 4 ..
       ( u*dir -30  ){dir -120} .. tension 4 ..
  cycle;
endfig;
```

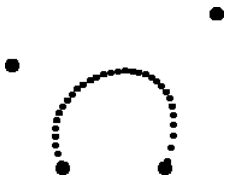


```
beginfig(60)
    u:=.5cm;
    pair A,B,C,D;
    A:=(0,0);   B:=(-u,2u);
    C:=(4u,3u); D:=(3u,0);
    draw A.. controls B and C .. D
         withpen pencircle scaled 2pt;
    draw B--C dashed evenly;
    drawarrow A--B;
    drawarrow D--C;

endfig;
```

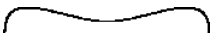


```
beginfig(61)
  pair A, B, C, D;
  A = (0,0);
  D = (2u,0);
  B = (-u,2u);
  C = (3u,3u);

  draw A withpen pencircle scaled 4bp;
  draw B withpen pencircle scaled 4bp;
  draw C withpen pencircle scaled 4bp;
  draw D withpen pencircle scaled 4bp;
  draw A .. controls B and C .. D;

  pair a[];
  a[1] := A;
  a[2] := B;
  a[3] := C;
  a[4] := D;

  a[12]   := 1/2 [ a[1],   a[2]   ];
  a[23]   := 1/2 [ a[2],   a[3]   ];
  a[34]   := 1/2 [ a[3],   a[4]   ];
```

```
  a[123]  := 1/2 [ a[12],  a[23]  ];
  a[234]  := 1/2 [ a[23],  a[34]  ];
  a[1234] := 1/2 [ a[123], a[234] ];

  draw a[1234] withpen pencircle scaled 4bp;
endfig;
```



```
beginfig(62)
vardef bezier(expr n,A,B,C,D) =
  save a;
  pair a[];
  a[1] := A;
  a[2] := B;
  a[3] := C;
  a[4] := D;

  a[12]   := 1/2 [ a[1],   a[2]   ];
  a[23]   := 1/2 [ a[2],   a[3]   ];
  a[34]   := 1/2 [ a[3],   a[4]   ];
  a[123]  := 1/2 [ a[12],  a[23]  ];
  a[234]  := 1/2 [ a[23],  a[34]  ];
  a[1234] := 1/2 [ a[123], a[234] ];

  if n>0:
    bezier(n-1, a[1], a[12], a[123], a[1234]);
    bezier(n-1, a[1234], a[234], a[34], a[4]);
  else:
    draw a[1234] withpen pencircle scaled 2bp;
  fi;
enddef;

  pair A, B, C, D;
  A = (0,0);
  D = (2u,0);
  B = (-u,2u);
  C = (3u,3u);

  draw A withpen pencircle scaled 4bp;
  draw B withpen pencircle scaled 4bp;
  draw C withpen pencircle scaled 4bp;
  draw D withpen pencircle scaled 4bp;

  bezier(5,A,B,C,D);
endfig;
```

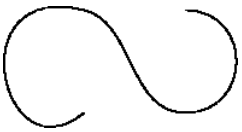


```
beginfig(63)
    draw (0,0){up} .. (1cm, 1mm) .. (2cm,0){down};

endfig;
```

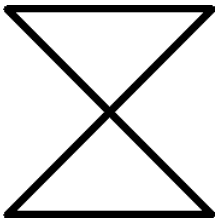
```
beginfig(64)
    draw (0,0){up} ... (1cm, 1mm) ... (2cm,0){down};

endfig;
```

```
beginfig(65)
    draw (0,0){curl 0} .. (0,1cm)..(1cm,0)..(1cm,1cm);

endfig;
```

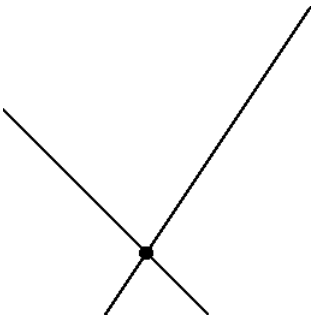
```
beginfig(66)
    draw (0,0){curl 1} .. (0,1cm)..(1cm,0)..(1cm,1cm);

endfig;
```



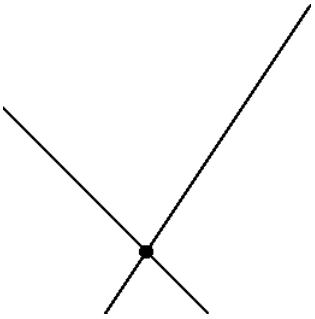
```
beginfig(67)
    draw (0,0){curl 2} .. (0,1cm)..(1cm,0)..(1cm,1cm);

endfig;
```



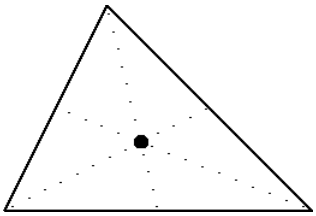
```
beginfig(68)
  pair A,B,C,D;
  xpart A = ypart A = ypart B = xpart C = 0;
  xpart B = ypart C = xpart D = ypart D = 2cm;
  draw A--B--C--D--cycle withpen pencircle scaled 2bp;
endfig;
```



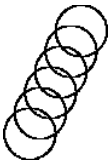
```
beginfig(69)
    pair A,B,C,D,M;
    numeric lambda, mu;
    A=(0,0); B=(2cm,3cm);
    C=(1cm,0); D=(-1cm,2cm);
    M = lambda [A,B];
    M = mu     [C,D];
    draw A--B;
    draw C--D;
    draw M withpen pencircle scaled 4bp;

endfig;
```



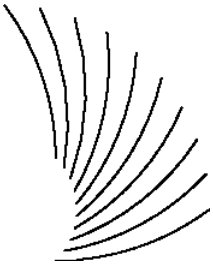
```
beginfig(70)
    pair A,B,C,D,M;
    A=(0,0); B=(2cm,3cm);
    C=(1cm,0); D=(-1cm,2cm);
    M = whatever [A,B];
    M = whatever [C,D];
    draw A--B;
    draw C--D;
    draw M withpen pencircle scaled 4bp;

endfig;
```



```
beginfig(71)
  pair A,B,C,AA,BB,CC,G;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  AA = 1/2 [B,C];
  BB = 1/2 [C,A];
  CC = 1/2 [A,B];
  G = whatever [A,AA];
  G = whatever [B,BB];
  draw A--B--C--cycle;
  draw A--AA dashed withdots;
  draw B--BB dashed withdots;
  draw C--CC dashed withdots;
  draw G withpen pencircle scaled 4bp;
endfig;
```

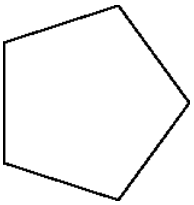


```
beginfig(72)
    path p;
    p := fullcircle scaled 5mm;
    draw p;
    draw p shifted  (1mm,2mm);
    draw p shifted 2(1mm,2mm);
    draw p shifted 3(1mm,2mm);
    draw p shifted 4(1mm,2mm);
    draw p shifted 5(1mm,2mm);

endfig;
```

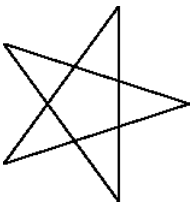


```
beginfig(73)
    path p;
    p := (5mm,-5mm) {right} .. (2cm,0);
    draw p;
    draw p rotated 10;
    draw p rotated 20;
    draw p rotated 30;
    draw p rotated 40;
    draw p rotated 50;
    draw p rotated 60;
    draw p rotated 70;
    draw p rotated 80;
    draw p rotated 90;

endfig;
```

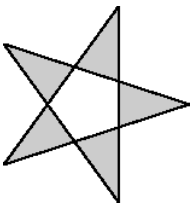
```
beginfig(74)
    path p;
    p := fullcircle scaled 5mm;
    draw p;
    draw p xscaled 2;
    draw p xscaled 3;
    draw p xscaled 4;

endfig;
```



```
beginfig(75)
  pair A,B,C,D,E;
  A := (1cm,0);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  draw A--B--C--D--E--cycle;
endfig;
```



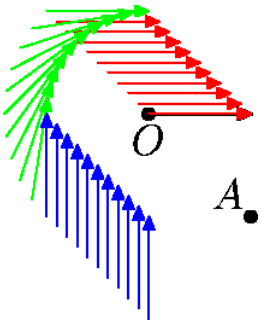
```
beginfig(76)
  pair A,B,C,D,E;
  A := (1cm,0);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  draw A--C--E--B--D--cycle;
endfig;
```



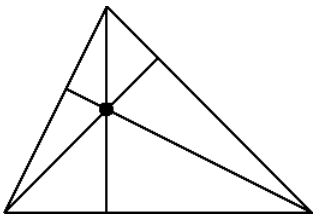
```
beginfig(77)
  pair A,B,C,D,E;
  A := (1cm,0);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  pair AA, BB, CC, DD, EE;
  AA = whatever [A, C];
  AA = whatever [B, D];
  BB = whatever [B, D];
  BB = whatever [C, E];
  CC = whatever [C, E];
  CC = whatever [D, A];
  DD = whatever [D, A];
  DD = whatever [E, B];
  EE = whatever [E, B];
  EE = whatever [A, C];
  fill A--C--E--B--D--cycle withcolor .8white;
  fill AA--BB--CC--DD--EE--cycle withcolor white;
  draw A--C--E--B--D--cycle;
endfig;
```

```
beginfig(78)
    path p;
    p := (0,0) -- (1cm,0);
    drawarrow p withpen pencircle scaled 2bp;
    drawarrow p zscaled (1,2);

endfig;
```

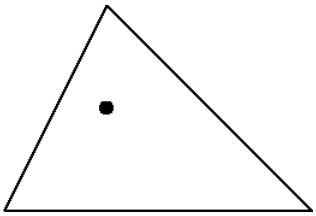
```
beginfig(79)
  u:=1cm;
  path p;     p := (0,0) -- (u,0);
  pair A;     A := (u,-u);
  numeric a;  a := 90;
  drawarrow p withpen pencircle scaled 1bp;
  drawarrow p rotatedaround( A, a );
  drawarrow p shifted -A rotated a shifted A
    withpen pencircle scaled 1bp dashed withdots;
endfig;
```



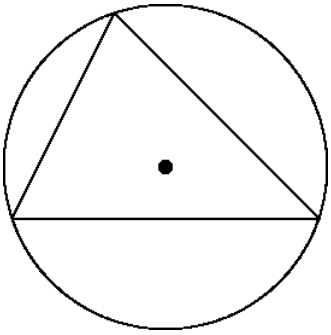

```
beginfig(80)
  path p;     p := (0,0) -- (u,0);
  pair A;     A := (u,-u);
  numeric a;  a := 90;
  drawarrow p withpen pencircle scaled 1bp;
  draw A withpen pencircle scaled 4bp;
  label ulft ( btex $A$ etex, A );
  draw (0,0) withpen pencircle scaled 4bp;
  label bot ( btex $O$ etex, (0,0) );
  for i=0 upto 10:
    drawarrow p shifted -(i*A/10) withcolor red;
  endfor;
  for i=0 upto 10:
    drawarrow p shifted -A rotated (i*a/10) withcolor green;
  endfor;
  for i=0 upto 10:
    drawarrow p shifted -A rotated a shifted (i*A/10) withcolor blue;
  endfor;
endfig;
```



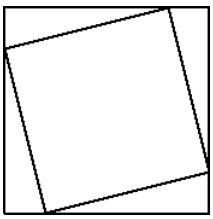
```
beginfig(81)
  pair A,B,C,AA,BB,CC,H;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  AA - A = whatever * (B-C) rotated 90;
  AA = whatever [B,C];
  BB - B = whatever * (A-C) rotated 90;
  BB = whatever [A,C];
  CC - C = whatever * (A-B) rotated 90;
  CC = whatever [A,B];
  H = whatever [A,AA];
  H = whatever [B,BB];
  draw A--B--C--cycle;
  draw A--AA;
  draw B--BB;
  draw C--CC;
  draw H withpen pencircle scaled 4bp;
endfig;
```



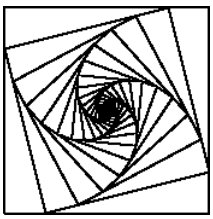
```
beginfig(82)
  pair A,B,C,H;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  H - A = whatever * (B-C) rotated 90;
  H - B = whatever * (A-C) rotated 90;
  draw A--B--C--cycle;
  draw H withpen pencircle scaled 4bp;
endfig;
```



```
beginfig(83)
  pair A,B,C,O;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  O - 1/2[B,C] = whatever * (B-C) rotated 90;
  O - 1/2[A,B] = whatever * (A-B) rotated 90;
  draw A--B--C--cycle;
  draw O withpen pencircle scaled 4bp;
  draw fullcircle scaled 2abs(O-A) shifted O;
endfig;
```

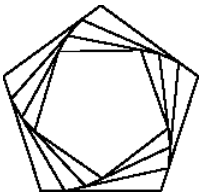


```
beginfig(84)
    pair A,B,C,D;
    u:=2cm;
    A=(0,0); B=(u,0); C=(u,u); D=(0,u);

    transform T;
    A transformed T = 1/5[A,B];
    B transformed T = 1/5[B,C];
    C transformed T = 1/5[C,D];

    path p;
    p = A--B--C--D--cycle;
    draw p;
    draw p transformed T;

endfig;
```



```
beginfig(85)
    pair A,B,C,D;
    u:=2cm;
    A=(0,0); B=(u,0); C=(u,u); D=(0,u);

    transform T;
    A transformed T = 1/5[A,B];
```

```
    B transformed T = 1/5[B,C];
    C transformed T = 1/5[C,D];

    path p;
    p = A--B--C--D--cycle;
    for i=0 upto 100:
      draw p;
      p:= p transformed T;
    endfor;

endfig;
```



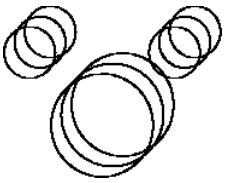
```
beginfig(86)
  u:=1cm;
  pair A,B,C,D,E;
  A := (0,u);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  transform T;
  A transformed T = 1/5[A,B];
  B transformed T = 1/5[B,C];
  C transformed T = 1/5[C,D];
  path p;
  p := A--B--C--D--E--cycle;
  draw p;
  p := p transformed T; draw p;
  p := p transformed T; draw p;
  p := p transformed T; draw p;
endfig;
```

```
beginfig(87)
  u:=3mm;
  fill fullcircle scaled 2u withcolor .8white;
  fill fullcircle scaled u shifted (u*dir30)
       withcolor .8white;
  fill fullcircle scaled u shifted (u*dir150)
       withcolor .8white;
endfig;
```

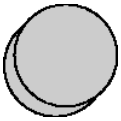
```
beginfig(88)
  u:=3mm;
  transform T;
  (0,0) transformed T = (0,0);
  (1,0) transformed T = (1,1);
  (0,1) transformed T = (1,0);

  fill fullcircle scaled 2u transformed T withcolor .8white;
  fill fullcircle scaled u shifted (u*dir30) transformed T
       withcolor .8white;
  fill fullcircle scaled u shifted (u*dir150) transformed T
       withcolor .8white;
endfig;
```

```
beginfig(89)
```

```
  u:=3mm;
  transform T;
  xpart T = ypart T = 0;
  xxpart T = 1;
  yxpart T = 2;
  xypart T = 3;
  yypart T = 4;

  fill fullcircle scaled 2u transformed T withcolor .8white;
  fill fullcircle scaled u shifted (u*dir30) transformed T
       withcolor .8white;
  fill fullcircle scaled u shifted (u*dir150) transformed T
       withcolor .8white;
endfig;
```



```
beginfig(90)
    draw fullcircle scaled 1cm;
    draw fullcircle scaled 5mm shifted (1cm*dir45);
    draw fullcircle scaled 5mm shifted (1cm*dir135);
    picture mypicture;
    mypicture := currentpicture;
    currentpicture := nullpicture;
    draw mypicture;
    draw mypicture shifted (1mm,1mm);
    draw mypicture shifted (2mm,2mm);

endfig;
```



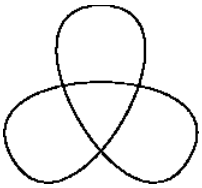
```
beginfig(91)
    picture pic;
    pic := nullpicture;
    addto pic contour fullcircle scaled 1cm
      withcolor .8white;
    addto pic doublepath fullcircle scaled 1cm
      withpen pencircle scaled .5bp;
    addto pic also pic shifted (1mm,1mm);
    draw pic;

endfig;
```

```
beginfig(92)
  u:=3mm;
  picture mickey;
  mickey := nullpicture;
  addto mickey contour fullcircle scaled 2u
        withcolor .8white;
  addto mickey contour fullcircle scaled u
        shifted (u*dir30) withcolor .8white;
  addto mickey contour fullcircle scaled u
        shifted (u*dir150)
        withcolor .8white;
  draw mickey;
endfig;
```

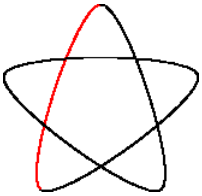
```
beginfig(93)
  u:=3mm;
  picture mickey;
  mickey := nullpicture;
  addto mickey contour fullcircle scaled 2u
        withcolor .8white;
  addto mickey contour fullcircle scaled u
        shifted (u*dir30) withcolor .8white;
  addto mickey contour fullcircle scaled u
        shifted (u*dir150) withcolor .8white;

  transform T;
  (0,0) transformed T = (0,0);
  (1,0) transformed T = (1,1);
  (0,1) transformed T = (1,0);
```

```
  draw mickey transformed T;
endfig;
```

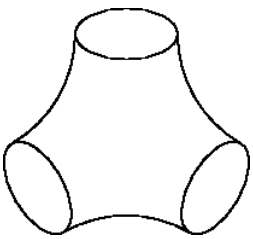
```
beginfig(94)
  u:=3mm;
  picture mickey;
  fill fullcircle scaled 2u withcolor .8white;
  fill fullcircle scaled u shifted (u*dir30)
       withcolor .8white;
  fill fullcircle scaled u shifted (u*dir150)
       withcolor .8white;
  mickey := currentpicture;
  currentpicture := nullpicture;

  transform T;
  (0,0) transformed T = (0,0);
  (1,0) transformed T = (1,1);
  (0,1) transformed T = (1,0);

  draw mickey transformed T;
endfig;
```



```
beginfig(95)
  pair A,B;
  path p;
  A = (0,1cm);
  B = A rotated 120;
  p = A{dir 0} .. tension 2 .. B{dir 120};
  draw p;
  draw p rotated 120;
  draw p rotated -120;
endfig;
```

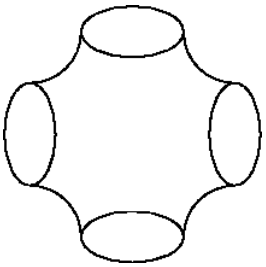


```
beginfig(96)
  pair A,B;
  path p;
  numeric n;
  n:=5;
  A = (0,1cm);
  B = A rotated (2*360/n);
  p = A{dir 180} .. tension 4 .. B{dir (180+2*360/n)};
  draw p withcolor red;
  draw p rotated (1*360/n);
  draw p rotated (2*360/n);
  draw p rotated (3*360/n);
  draw p rotated (4*360/n);
endfig;
```

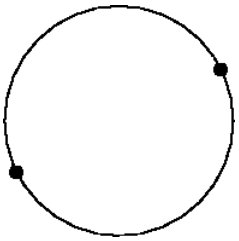


```
beginfig(97)
  path c[], l[];
  c[0] = fullcircle xscaled 1cm yscaled .5cm
         shifted (0,1cm);
  c[1] = c[0] rotated 120;
  c[2] = c[1] rotated 120;
```

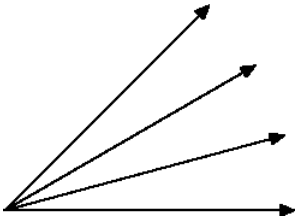
```
  pair A;
  A = (-.5cm,1cm);
  l[0] = A{down} ..
         (A xscaled -1 rotated 120){-down rotated 120};
  l[1] = l[0] rotated 120;
  l[2] = l[1] rotated 120;
  draw c[0]; draw c[1]; draw c[2];
  draw l[0]; draw l[1]; draw l[2];
endfig;
```



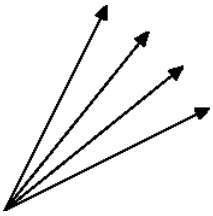
```
beginfig(98)
  path c[], l[];
  c[0] = fullcircle xscaled 1cm yscaled .5cm
                    shifted (0,1cm);
  c[1] = c[0] rotated (360/4);
  c[2] = c[1] rotated (360/4);
  c[3] = c[2] rotated (360/4);
  pair A;
  A = (-.5cm,1cm);
  l[0] = A{down} ..
         (A xscaled -1 rotated (360/4))
         {-down rotated (360/4)};
  l[1] = l[0] rotated (360/4);
  l[2] = l[1] rotated (360/4);
  l[3] = l[2] rotated (360/4);
  draw c[0]; draw c[1]; draw c[2]; draw c[3];
  draw l[0]; draw l[1]; draw l[2]; draw l[3];
endfig;
```



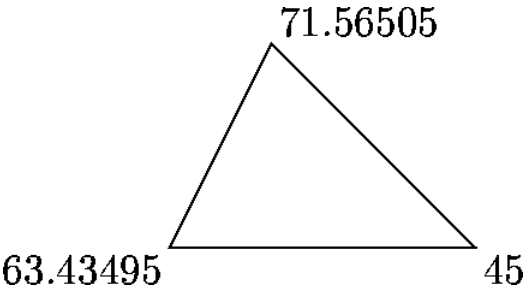
```
beginfig(99)
    pair A, B;
    A := (0,0); B := (2cm,1cm);
    draw A withpen pencircle scaled 4bp;
    draw B withpen pencircle scaled 4bp;
    draw fullcircle scaled abs(B-A) shifted 1/2[A,B];

endfig;
```



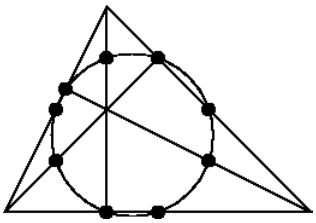
```
beginfig(100)
    pair A;
    A:=(2cm,2cm);
    drawarrow origin--A;
    drawarrow (origin--A) rotated -1/3 angle(A);
    drawarrow (origin--A) rotated -2/3 angle(A);
    drawarrow (origin--A) rotated -angle(A);

endfig;
```



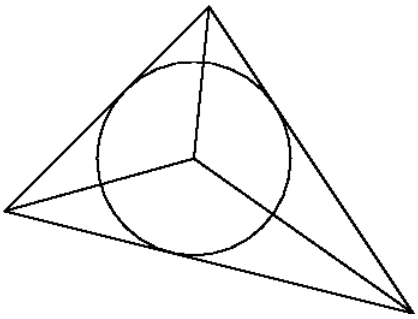
```
beginfig(101)
  pair A,B;
  A:=(1cm,2cm); B:=(2cm,1cm);
  numeric alpha;
  alpha = angle(A) - angle(B);
  drawarrow origin--A;
  drawarrow origin--B;
  drawarrow (origin--A) rotated -1/3 alpha;
  drawarrow (origin--A) rotated -2/3 alpha;
  drawarrow (origin--A) rotated -alpha;
endfig;
```




```
beginfig(102)
  pair A,B,C;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  draw A--B--C--cycle;
  label.llft(TEX decimal(angle(C-A)-angle(B-A)), A);
  label.lrt(TEX decimal(angle(A-B)-angle(C-B)), B);
  label.urt(TEX decimal(angle(B-C)-angle(A-C)), C);
endfig;
```

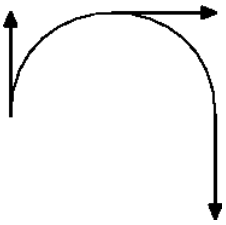


```
beginfig(103)
  pair A,AA,B,BB,C,CC,O,H;
  A=(0,0); B=(3cm,0); C=(1cm,2cm);
  AA = 1/2[B,C];
  BB = 1/2[A,C];
  CC = 1/2[A,B];
  O - 1/2[BB,CC] = whatever * (BB-CC) rotated 90;
  O - 1/2[AA,BB] = whatever * (AA-BB) rotated 90;
  draw A--B--C--cycle;
  draw AA withpen pencircle scaled 4bp;
  draw BB withpen pencircle scaled 4bp;
  draw CC withpen pencircle scaled 4bp;
  draw fullcircle scaled 2abs(O-AA) shifted O;
  % Il faut aussi tracer les hauteurs
  pair AA,BB,CC;
  AA - A = whatever * (B-C) rotated 90;
  AA = whatever [B,C];
  BB - B = whatever * (A-C) rotated 90;
  BB = whatever [A,C];
  CC - C = whatever * (A-B) rotated 90;
  CC = whatever [A,B];
  draw A--AA; draw B--BB; draw C--CC;
  draw AA withpen pencircle scaled 4bp;
  draw BB withpen pencircle scaled 4bp;
  draw CC withpen pencircle scaled 4bp;
  % Il passe aussi par le milieu de HA, HB, HC
  H = whatever [A,AA];
  H = whatever [B,BB];
  draw 1/2 [A,H] withpen pencircle scaled 4bp;
  draw 1/2 [B,H] withpen pencircle scaled 4bp;
  draw 1/2 [C,H] withpen pencircle scaled 4bp;
endfig;
```

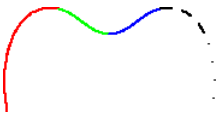


```
beginfig(104)
  pair A,B,C,M,h;
  u:=2cm;
  A=(0,0); B=(2u,-.5u); C=(u,u);
  draw A--B--C--cycle;
  (M-A) = whatever * (
    (A-C) rotated 1/2( angle(B-A) - angle(C-A)) );
  (M-B) = whatever * (
    (B-A) rotated 1/2( angle(C-B) - angle(A-B)) );
  draw M--A; draw M--B; draw M--C;
  M-h = whatever * (B-C) rotated 90;
  h = whatever[B,C];
  draw fullcircle scaled 2 abs(M-h) shifted M;
endfig;
```

```
beginfig(105)
    path p;
    p = (0,0){up} .. (2cm,0){down};
    draw p;
    draw point 0  of p withpen pencircle scaled 4bp;
    draw point .5 of p withpen pencircle scaled 4bp;
    draw point 1  of p withpen pencircle scaled 4bp;

endfig;
```

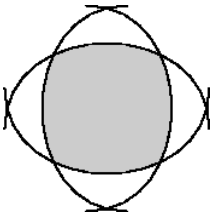


```
beginfig(106)
    path p;
    p = (0,0){up} .. (2cm,0){down};
    draw p;
    pair A;
    A := point 0  of p;
    B := A + 1cm*unitvector(direction 0  of p);
    drawarrow A--B withpen pencircle scaled 1bp;
    A := point .5 of p;
    B := A + 1cm*unitvector(direction .5 of p);
    drawarrow A--B withpen pencircle scaled 1bp;
    A := point 1  of p;
    B := A + 1cm*unitvector(direction 1  of p);
    drawarrow A--B withpen pencircle scaled 1bp;

endfig;
```



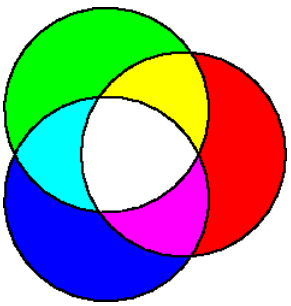
```
beginfig(107)
  u:=5mm;
  path p;
  p = (0,0) {curl 0} .. (u,2u) .. (2u,1.5u)
      .. (3u,2u) .. {curl 0} (4u,0);
  draw subpath(0,1) of p withcolor red;
  draw subpath(1,2) of p withcolor green;
  draw subpath(2,3) of p withcolor blue;
  draw subpath(3,3.5) of p dashed evenly;
  draw subpath(3.5,4) of p dashed withdots;

endfig;
```

```
beginfig(108)
  u:=1cm;
  path p,q;
  p := (0,0){up} .. (u,2u){up};
  q := (u,0){up} .. (0,2u){up};
  draw p;
  draw subpath(0,.4) of q withpen pencircle scaled 1bp;
  draw subpath(.6,1) of q withpen pencircle scaled 1bp;
endfig;
```



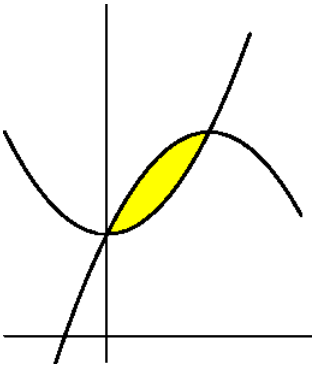
```
beginfig(109)
  u:=1cm;
  path a,b,c,d;
  a = (-u,-.2u){up} .. tension 1.2 .. (u,-.2u){down};
  b = a rotated 90;
  c = b rotated 90;
  d = c rotated 90;
  fill buildcycle(a,b,c,d) withcolor .8white;
  draw a; draw b; draw c; draw d;

endfig;
```

```
beginfig(110)
  u:=1cm;
  path c[];
  c[1] := fullcircle scaled u;
  c[2] := c[1] shifted (0,.5u);
  draw c[1] dashed evenly;
  draw c[2] dashed evenly;
  draw buildcycle(c[1],c[2]) withpen pencircle scaled 1bp;
endfig;
```

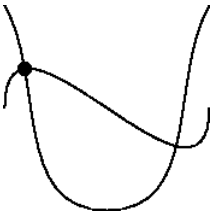


```
beginfig(111)
  path a,b,c;
  a = fullcircle scaled 2u shifted (.5u,0);
  b = a rotated (360/3);
  c = b rotated (360/3);
  fill a withcolor red;
  fill b withcolor green;
  fill c withcolor blue;
  fill buildcycle(a,b) withcolor red + green;
  fill buildcycle(b,c) withcolor green + blue;
  fill buildcycle(c,a) withcolor blue + red;
  fill buildcycle(a,b,c) withcolor white;
  draw a; draw b; draw c;
```

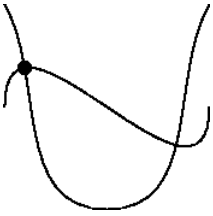
```
endfig;
```



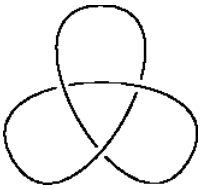
```
beginfig(112)
    def compute_curve(suffix f)(expr xmin, xmax, xinc) =
    ( (xmin,f(xmin))
    for x=xmin+xinc step xinc until xmax:
      .. (x,f(x))
    endfor )
  enddef;
  vardef f(expr x) = x**2 + 1 enddef;
  vardef g(expr x) = 2 - (x-1)**2 enddef;
  path p, q;
  p := compute_curve(f, -1, 1.5, .1) scaled 1cm;
  q := compute_curve(g, -.5, 2, .1) scaled 1cm;
  fill buildcycle(p,reverse q) withcolor red+green;
  draw p withpen pencircle scaled 1bp;
  draw q withpen pencircle scaled 1bp;
  draw (-1cm,0) -- (2cm,0);
  draw (0,g(-.5)*1cm) -- (0,f(1.5)*1cm);
endfig;
```



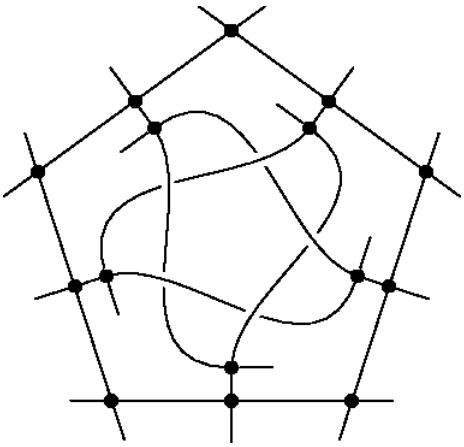
```
beginfig(113)
    path p, q;
    p = (0,0){up} .. (2cm,0){up};
    q = (0,1cm){dir -60}..(1cm,-1cm)..{dir 60}(2cm,1cm);
    draw p; draw q;
    draw p intersectionpoint q withpen pencircle scaled 4bp;

endfig;
```



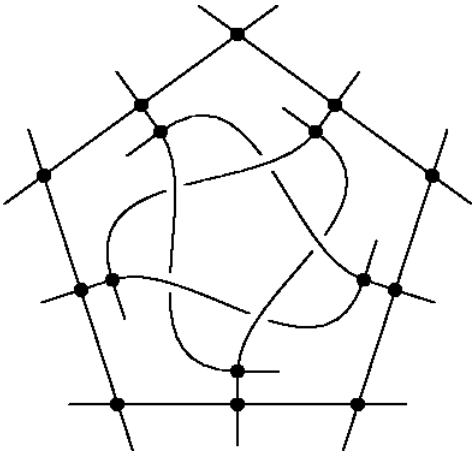
```
beginfig(114)
    path p, q;
    p = (0,0){up} .. (2cm,0){up};
    q = (0,1cm){dir -60}..(1cm,-1cm)..{dir 60}(2cm,1cm);
    draw p; draw q;
    numeric a,b;
    (a,whatever) = p intersectiontimes q;
    draw point a of p withpen pencircle scaled 4bp;

endfig;
```



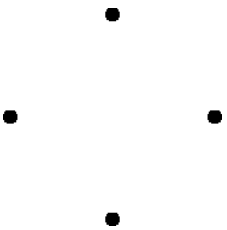
```
beginfig(115)
  pair A,B;
  path p;
  A = (0,1cm);
  B = A rotated 120;
  p = A{dir 0} .. tension 2 .. B{dir 120};
  numeric a;
  (a,whatever) = p intersectiontimes (p rotated 120);
  draw subpath(0,a-.02) of p;
  draw subpath(a+.02,1) of p;
  draw subpath(0,a-.02) of p rotated 120;
  draw subpath(a+.02,1) of p rotated 120;
  draw subpath(0,a-.02) of p rotated -120;
  draw subpath(a+.02,1) of p rotated -120;
endfig;
```



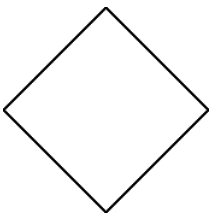
```
beginfig(116)
  u:=2cm;
  pair A[], B[], C[], D[], E[];
  path p[];

  A[0] = u*up;
  for i=1 upto 10:
    A[i] := A[i-1] rotated 72;
  endfor;
  for i=0 upto 4:
    p[i] := A[i]--A[i+1];
    draw p[i];
    draw (point 1 of p[i]) --
    ( (point 1 of p[i]) + 4mm*unitvector(direction 1 of p[i]));
    draw (point 0 of p[i]) --
    ( (point 0 of p[i]) - 4mm*unitvector(direction 0 of p[i]));
  endfor

  for i=0 upto 9:
    B[i] := 1/2[ A[i], A[i+1] ];
  endfor;
  B[5]:=B[0];

  for i=0 upto 9:
    C[i] := .8*B[i];
  endfor;

  for i=0 upto 4:
    p[i] := B[i] --- C[i] .. C[i+2]{dir 72i};
    draw p[i];
    draw (point 2 of p[i]) --
    ( (point 2 of p[i]) + 4mm*unitvector(direction 2 of p[i]));
    draw (point 0 of p[i]) --
    ( (point 0 of p[i]) - 4mm*unitvector(direction 0 of p[i]));
  endfor;
  for i=0 upto 4:
    draw subpath(1,1.4) of p[i] withpen pencircle scaled 4bp withcolor white;
    draw subpath(0,1.5) of p[i];
  endfor;

  for i=0 upto 4:
    draw A[i] withpen pencircle scaled 4bp;
    draw B[i] withpen pencircle scaled 4bp;
    draw C[i] withpen pencircle scaled 4bp;
  endfor;
endfig;
```



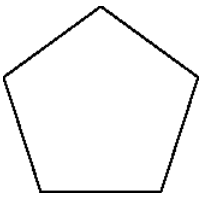
```
beginfig(117)
  u:=2cm;
  pair A, B, C, D, E;
  path p, q, r;

  A = u*up;
  p := (-.2)[ A, A rotated 72 ] -- (1.2)[ A, A rotated 72 ];
  for i=0 upto 5:
    draw p rotated 72i;
  endfor;
  B := 1/2[ A, A rotated 72 ];
  C := .8*B;

  p := B --- C .. (C rotated (2*72)){right};
  % On allonge le chemin p
  p := ( (point 0 of p) - 4mm*unitvector(direction 0 of p))
       --
       (point 0 of p)
       & p &
       (point 2 of p)
       --
       ( (point 2 of p) + 4mm*unitvector(direction 2 of p));

  E = p intersectionpoint (p rotated 72);
  q := p cutbefore fullcircle scaled -2mm shifted E;
  r := p cutafter fullcircle scaled  2mm shifted E;

  for i=0 upto 4:
    draw q rotated 72i;
    draw r rotated 72i;
    draw A rotated 72i withpen pencircle scaled 4bp;
    draw B rotated 72i withpen pencircle scaled 4bp;
    draw C rotated 72i withpen pencircle scaled 4bp;
  endfor;
endfig;
```



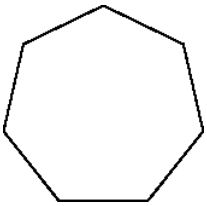
```
beginfig(118)
    for i=0 step 1 until 3:
      draw 1cm*right rotated (i*90)
           withpen pencircle scaled 4bp;
    endfor;

endfig;
```



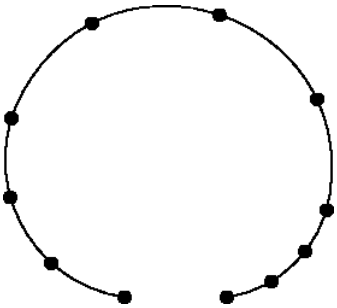
```
beginfig(119)
    draw for i=0 step 1 until 3:
        1cm*right rotated (i*90) --
      endfor cycle;

endfig;
```



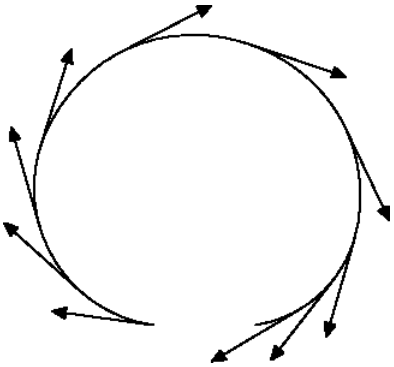
```
beginfig(120)
  n:=5;
  draw for i=0 step 1 until n-1:
      1cm*up rotated (i*360/n) --
    endfor cycle;
endfig;
```



```
beginfig(121)
  n:=7;
  draw for i=0 step 1 until n-1:
      1cm*up rotated (i*360/n) --
    endfor cycle;
endfig;
```

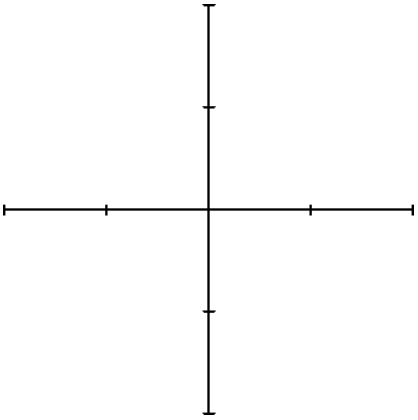


```
beginfig(122)
  path p;
  p = (0,0) .. (-1cm,2cm) .. (2cm,1cm) .. (1cm,0);
  draw p;
  n:=10;
  for i=0 step 1 until n:
    draw point (i/n*length(p)) of p
         withpen pencircle scaled 4bp;
  endfor;
endfig;
```

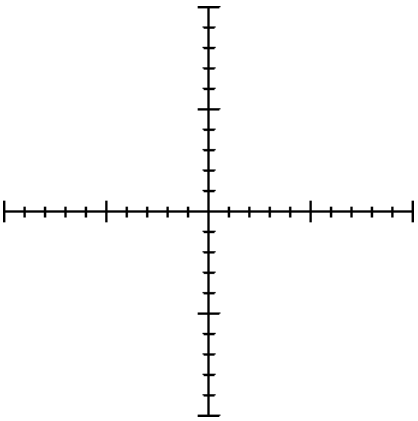


```
beginfig(123)
  path p;
  p = (0,0) .. (-1cm,2cm) .. (2cm,1cm) .. (1cm,0);
  draw p;
  n:=10;
  for i=0 step length(p)/n until length(p):
    drawarrow (point i of p) --
              1cm * unitvector(direction i of p)
              shifted point i of p;
  endfor;
```

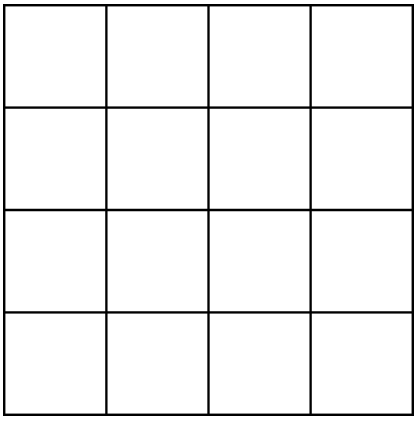
```
endfig;
```



```
beginfig(124)
  u:=1cm;
  draw (-2u,0)--(2u,0);
  draw (0,-2u)--(0,2u);
  for i=-2u step u until 2u:
    draw (i,u/20)--(i,-u/20);
    draw (u/20,i)--(-u/20,i);
  endfor;
endfig;
```

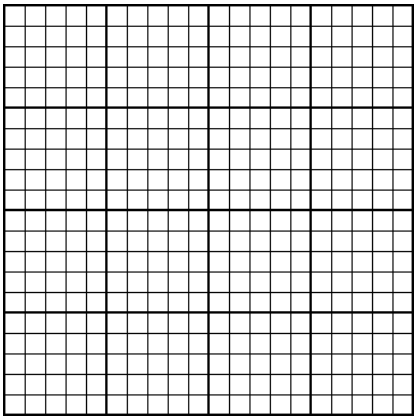


```
beginfig(125)
  u:=1cm;
  draw (-2u,0)--(2u,0);
  draw (0,-2u)--(0,2u);
  for i=-2u step u until 2u:
    draw (i,u/10)--(i,-u/10);
    draw (u/10,i)--(-u/10,i);
  endfor;
  for i=-2u step u/5 until 2u:
    draw (i,u/20)--(i,-u/20);
    draw (u/20,i)--(-u/20,i);
  endfor;
endfig;
```



```
beginfig(126)
  u:=1cm;
```

```
  draw (-2u,0)--(2u,0);
  draw (0,-2u)--(0,2u);
  for i=-2u step u until 2u:
    draw (i,2u)--(i,-2u);
    draw (2u,i)--(-2u,i);
  endfor;
endfig;
```



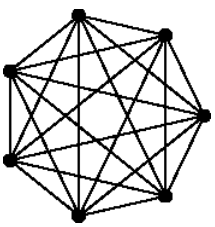
```
beginfig(127)
  u:=1cm;
  draw (-2u,0)--(2u,0);
  draw (0,-2u)--(0,2u);
  for i=-2u step u until 2u:
    draw (i,2u)--(i,-2u);
    draw (2u,i)--(-2u,i);
  endfor;
  for i=-2u step u/5 until 2u:
    draw (i,2u)--(i,-2u) withpen pencircle scaled .2bp;
    draw (2u,i)--(-2u,i) withpen pencircle scaled .2bp;
  endfor;
endfig;
```

```
beginfig(128)
  u := 5mm;
  % vardef est n^^e9cessaire pour pouvoir passer f en argument
  vardef f(expr x) = x**2 -.1 enddef;
  def axes(expr xmin,xmax,ymin,ymax) =
    draw ( (xmin,0) -- (xmax,0) ) scaled u;
    draw ( (0,ymin) -- (0,ymax) ) scaled u;
  enddef;
  def courbe(suffix f)(expr xmin, xmax, M) =
    draw ( ( xmin, f(xmin) )
    for i=1 upto M:
    -- ( xmin + (i/M)*(xmax - xmin), f( xmin + (i/M)*(xmax - xmin) ))
    endfor ) scaled u;
  enddef;
  vardef newton(suffix f)(expr y, h, M) =
    save x,t;
    numeric x,t; x:=y;
    for i=1 upto M:
      t := x - f(x)/(  (f(x+h) - f(x))/h );
      draw ( (x,f(x)) -- (t,0) -- (t,f(t)) ) scaled u;
      x := t;
    endfor;
  enddef;
  axes(-.5,2,-.5,4);
  courbe(f,-.5,2, 100);
  newton(f, 2, .01, 10);
endfig;
```

```
beginfig(129)
  transform T;
  u:=1cm;
  z0=(0,0); z1=(2u,0); z3 = z1 rotated 90; z2 = z1+z3;
  z0 transformed T = .1[z0,z1];
  z1 transformed T = .1[z1,z2];
  z2 transformed T = .1[z2,z3];
  path p;
  p = z0--z1--z2--z3--cycle;
  fill p withcolor .8*white;
  fill p transformed T withcolor white;
endfig;
```

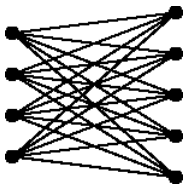
```
beginfig(130)
  transform T;
  u:=1cm;
  z0=(0,0); z1=(2u,0); z3 = z1 rotated 90; z2 = z1+z3;
  z0 transformed T = .1[z0,z1];
  z1 transformed T = .1[z1,z2];
  z2 transformed T = .1[z2,z3];
  path p;
  p = z0--z1--z2--z3--cycle;
  for i=0 upto 100:
    fill p withcolor .8*white;
    p := p transformed T;
    fill p withcolor white;
    p := p transformed T;
  endfor;
endfig;
```

```
beginfig(131)
  transform T;
  u:=1cm;
  z1=(0,2u);
  n := 5;
  for i=1 upto n-1:
    z[i+1] = z1 rotated (360*i/n);
  endfor;
  z1 transformed T = .1[z1,z2];
  z2 transformed T = .1[z2,z3];
  z3 transformed T = .1[z3,z4];
  path p;
  p = for i=1 upto n: z[i] -- endfor cycle;
  for i=0 upto 100:
    fill p withcolor .8*white;
    p := p transformed T;
    fill p withcolor white;
    p := p transformed T;
  endfor;
endfig;
```

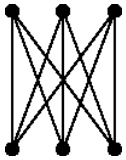


```
beginfig(132)
  u:=1cm;
```

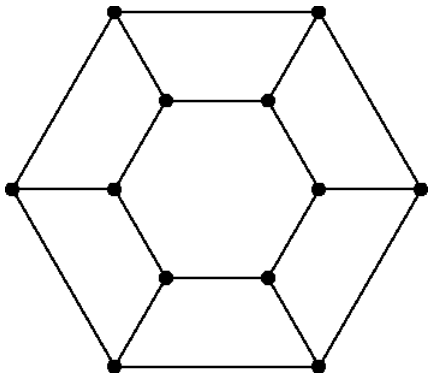
```
  pair A[];
  numeric n; n:=7;
  A[0] = (u,0);
  for i=1 upto n-1:
    A[i] = A[0] rotated (360/n*i);
  endfor;
  for i=0 upto n-1:
    draw A[i] withpen pencircle scaled 4bp;
    for j=0 upto n-1:
      if i<>j: draw A[i]--A[j] fi;
    endfor;
  endfor;
endfig;
```



```
beginfig(133)
  u:=2cm;
  numeric n,m; n:=4; m:=5;
  pair A[], B[];
  for i=1 upto n-1: A[i+1]-A[i] = (0,.2u); endfor;
  for j=1 upto m-1: B[j+1]-B[j] = (0,.2u); endfor;
  (0,0) for i=1 upto n: + A[i] endfor  = (0,0);
  (0,0) for j=1 upto m: + B[j] endfor  = (4u,0);
  for i=1 upto n:
    draw A[i] withpen pencircle scaled 4bp;
  endfor;
  for j=1 upto m:
    draw B[j] withpen pencircle scaled 4bp;
  endfor;
  for i=1 upto n:
    for j=1 upto m:
      draw A[i]--B[j];
    endfor;
  endfor;
endfig;
```

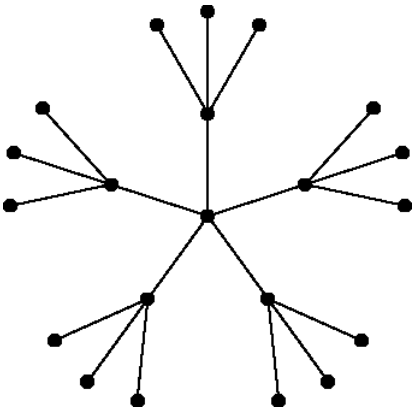


```
beginfig(134)
  u:=1cm;
  numeric n,m; n:=3; m:=3;
  pair A[], B[];
  for i=1 upto n-1: A[i+1]-A[i] = (.5u,0); endfor;
  for j=1 upto m-1: B[j+1]-B[j] = (.5u,0); endfor;
  (0,0) for i=1 upto n: + A[i] endfor  = (0,0);
  (0,0) for j=1 upto m: + B[j] endfor  = (0,4u);
  for i=1 upto n:
    draw A[i] withpen pencircle scaled 4bp;
  endfor;
  for j=1 upto m:
    draw B[j] withpen pencircle scaled 4bp;
  endfor;
  for i=1 upto n:
    for j=1 upto m:
      draw A[i]--B[j];
    endfor;
  endfor;
endfig;
```

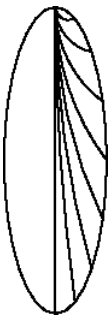


```
beginfig(135)
  pair A[], B[];
```

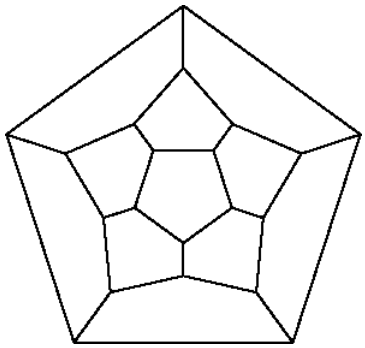
```
  numeric n; n:=6;
  for i=0 upto n-1:
    A[i] = 1cm * right rotated (i*360/n);
    B[i] = 2cm * right rotated (i*360/n);
  endfor;
  A[n] = A[0]; B[n] = B[0];
  for i=0 upto n-1:
    draw A[i] -- A[i+1] -- B[i+1] -- B[i];
    draw A[i] withpen pencircle scaled 4bp;
    draw B[i] withpen pencircle scaled 4bp;
  endfor;
endfig;
```



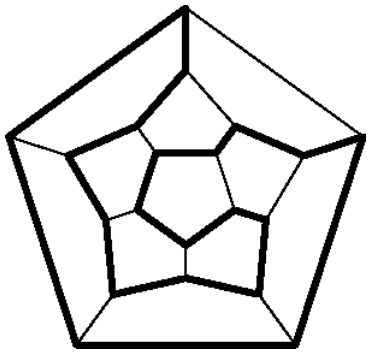
```
beginfig(136)
  numeric n; n:=5;
  pair O,A,B,C,D;
  O = (0,0);
  A = 1cm*up;
  B = 2cm*up rotatedabout(A,30);
  C = 2cm*up;
  D = 2cm*up rotatedabout(A,-30);
  for i=0 upto n-1:
    draw (O--A--C) rotated (i*360/n);
    draw (B--A--D) rotated (i*360/n);
    draw A rotated (i*360/n) withpen pencircle scaled 4bp;
    draw B rotated (i*360/n) withpen pencircle scaled 4bp;
    draw C rotated (i*360/n) withpen pencircle scaled 4bp;
    draw D rotated (i*360/n) withpen pencircle scaled 4bp;
  endfor;
  draw O withpen pencircle scaled 4bp;
endfig;
```



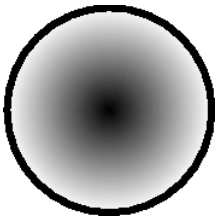
```
beginfig(137)
  path p; u:=1cm;
  p = fullcircle xscaled -u yscaled 3u;
  draw p;
  for i=2 step .5 until 6:
    draw (point 2 of p){down} .. (point i of p);
  endfor;
endfig;
```



```
beginfig(138)
  pair P[], Q[], R[], S[];
  u:=.5cm;
  for i=0 upto 4:
    P[i] = u* down rotated (i*360/5);
  endfor;
  P[5] = P[0];
  for i=0 upto 4:
    Q[i] = 3*( 1/2[ P[i], P[i+1] ] );
  endfor;
  Q[5] = Q[0];
  for i=0 upto 4:
    R[i] = 1/3( Q[i] + Q[i+1] + P[i+1] );
  endfor;
  R[5] = R[0];
  for i=0 upto 5:
    S[i] = 1.5*Q[i];
  endfor;
  for i=0 upto 4:
    draw P[i]   -- P[i+1];
    draw P[i+1] -- R[i];
    draw Q[i]   -- R[i];
    draw R[i]   -- Q[i+1];
    draw Q[i]   -- S[i];
    draw S[i]   -- S[i+1];
  endfor;
endfig;
```

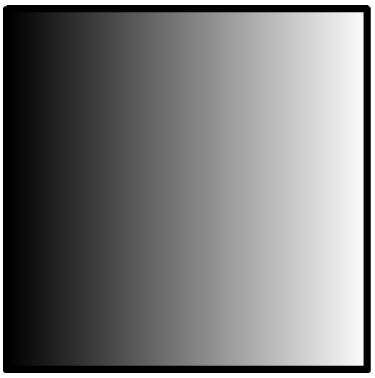


```
beginfig(139)
  pair P[], Q[], R[], S[];
  u:=.5cm;
  for i=0 upto 4:
    P[i] = u* down rotated (i*360/5);
  endfor;
  P[5] = P[0];
  for i=0 upto 4:
    Q[i] = 3*( 1/2[ P[i], P[i+1] ] );
  endfor;
  Q[5] = Q[0];
  for i=0 upto 4:
    R[i] = 1/3( Q[i] + Q[i+1] + P[i+1] );
  endfor;
  R[5] = R[0];
  for i=0 upto 5:
    S[i] = 1.5*Q[i];
  endfor;
  for i=0 upto 4:
    draw P[i]   -- P[i+1];
    draw P[i+1] -- R[i];
    draw Q[i]   -- R[i];
    draw R[i]   -- Q[i+1];
    draw Q[i]   -- S[i];
    draw S[i]   -- S[i+1];
  endfor;
  draw P[2] -- P[3] -- P[4] -- P[0] -- P[1] --
       R[0] -- Q[0] -- R[4] -- Q[4] -- R[3]
            -- Q[3] -- R[2] -- Q[2] --
       S[2] -- S[3] -- S[4] -- S[0] -- S[1] --
       Q[1] -- R[1] -- cycle
       withpen pencircle scaled 2bp;
endfig;
```

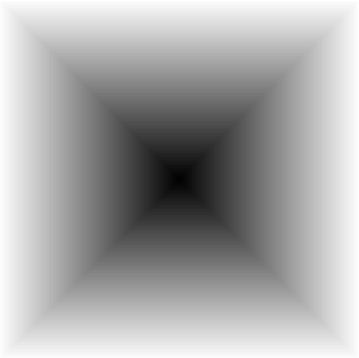


```
beginfig(140)
  for i=1 step -.01 until 0:
    fill fullcircle scaled (i*2cm) withcolor i*white;
  endfor;
  draw fullcircle scaled 2cm withpen pencircle scaled 2bp;
endfig;
```

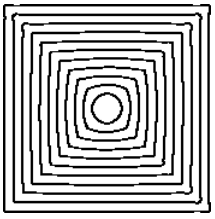
```
beginfig(141)
  u:=5mm;
  path p;
  p = (0,0) .. (-1,1) .. (2,0) .. (0,-3) .. cycle;
  p := p shifted (-1,0);
  for i=1 step -.01 until 0:
    fill p scaled (i*u) withcolor i*white;
  endfor;
  draw p scaled u withpen pencircle scaled 2bp;
endfig;
```



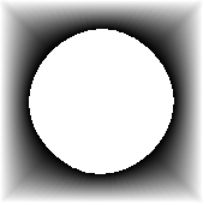
```
beginfig(142)
  z0 = (50,50);
  z1 = z0 rotated 90;
  z2 = z1 rotated 90;
  z3 = z2 rotated 90;
  path carre;
  carre = z0--z1--z2--z3--cycle;
  s := .01;
  path rect;
  z4 = s [z2,z3]; z5 = s [z1,z0];
  rect = z1--z2--z4--z5--cycle;
  for i=0 step s until 1:
    fill rect shifted (i*(z0-z1)) withcolor i*white;
  endfor;
  draw carre withpen pencircle scaled 2bp;
endfig;
```



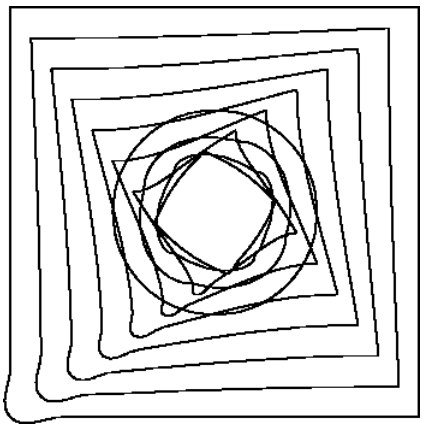
```
beginfig(143)
  z0 = (50,50);
  z1 = z0 rotated 90;
  z2 = z1 rotated 90;
  z3 = z2 rotated 90;
  path carre;
  carre = z0--z1--z2--z3--cycle;
  s := .01;
  for i=1 step -s until s:
    fill carre scaled i withcolor i*white;
  endfor;
endfig;
```



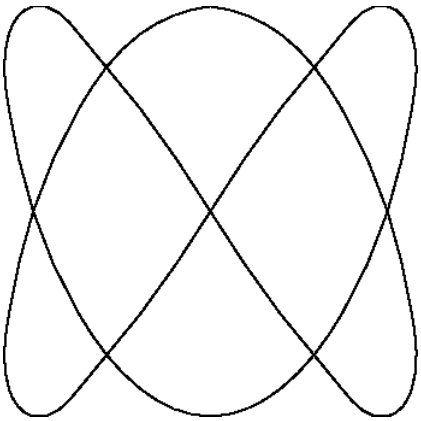
```
beginfig(144)
  u:=1cm;
  vardef degrade(expr p,q,M,N) =
    save a,b;
    numeric a,b;
    a := length(p);
    b := length(q);
    for i=0 upto M:
      draw (i/M) [ point 0 of p, point 0 of q ]
      for j=1 upto N:
        .. (i/M) [ point (j/N*a) of p, point (j/N*b) of q ]
      endfor;
    endfor;
  enddef;
  degrade(
    (-.1u,-.1u) .. (.1u,.1u) .. cycle,
    (-u,-u) -- (u,-u) -- (u,u) -- (-u,u) -- cycle,
    10,
    100);
endfig;
```



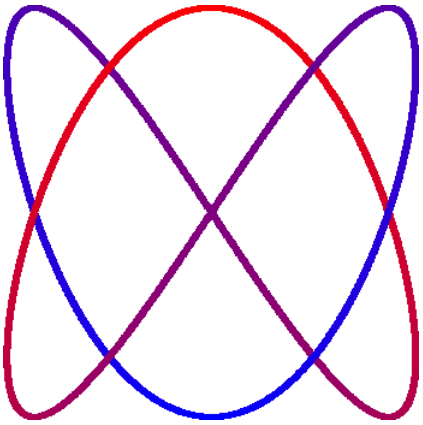
```
beginfig(145)
  u:=1cm;
  def couleur(expr c) = c*white enddef;
  vardef degrade(expr p,q,M,N) =
    save a,b;
    numeric a,b;
    a := length(p);
    b := length(q);
    for i=0 upto M:
      draw (i/M) [ point 0 of p, point 0 of q ]
      for j=1 upto N:
        .. (i/M) [ point (j/N*a) of p, point (j/N*b) of q ]
      endfor
      withcolor couleur(i/M);
    endfor;
  enddef;
  degrade(
    (-.5u,-.5u) .. (.5u,.5u) .. cycle,
    (-u,-u) -- (u,-u) -- (u,u) -- (-u,u) -- cycle,
    255,
    100);
endfig;
```



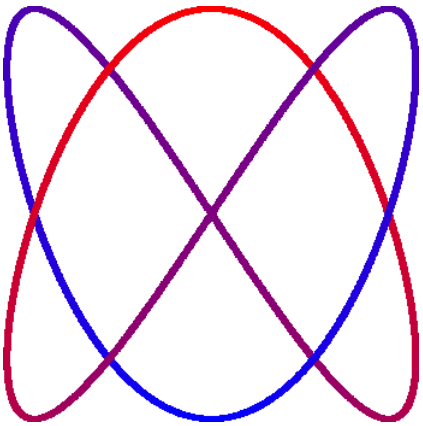
```
beginfig(146)
  def degrade(expr p, q, N) =
    begingroup
      save n, m, M;
      numeric n, m, M;
      n := length(p);
      m := length(q);
      M := n*m; % Il faudrait prendre le ppcm
      for i=0 upto N:
        draw
        (i/N)[ point 0 of p, point 0 of q ]
        { (i/N)[ direction 0 of p, direction 0 of q ] }
        for j=1 upto M-1:
          ..
          { (i/N) [ direction 1 of subpath((j-1)*n/M,j*n/M) of p,
                    direction 1 of subpath((j-1)*m/M,j*m/M) of q  ] }
          (i/N)[ point (j*n/M) of p, point (j*m/M) of q ]
          { (i/N) [ direction 0 of subpath(j*n/M,(j+1)*n/M) of p,
                    direction 0 of subpath(j*m/M,(j+1)*m/M) of q  ] }
        endfor
        ..
        { (i/N)[ direction n of p, direction m of q ] }
        (i/N)[ point n of p, point m of q ]
        ;
      endfor;
    endgroup;
  enddef;
  numeric u;
  u := 2cm;
  path p, q;
  p := fullcircle scaled u;
  q := (-u,-u)--(u,-u)--(u,u)--(-u,u)--cycle;
  degrade(p,q,10);
endfig;
```



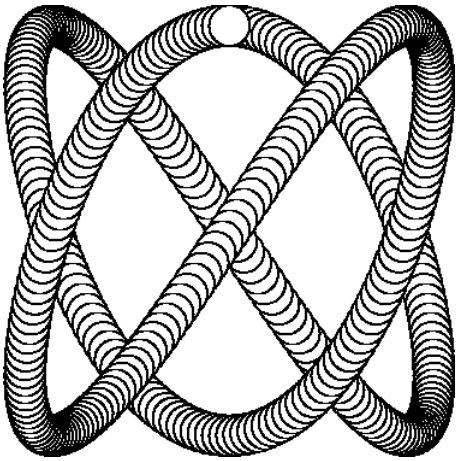
```
beginfig(147)
  draw for i=0 step 10 until 360:
    2cm* (sind(2*i), cosd(3*i)) ..
  endfor cycle;
endfig;
```



```
beginfig(148)
  for i=0 step .1 until 360:
    col := i/360;
    draw 2cm* (sind(2*i), cosd(3*i))
    withpen pencircle scaled 2bp
    withcolor
      if col>.5: (2*(1-col)) [red, blue]
      else:    (1-2col) [blue, red]
      fi;
  endfor;
endfig;
```

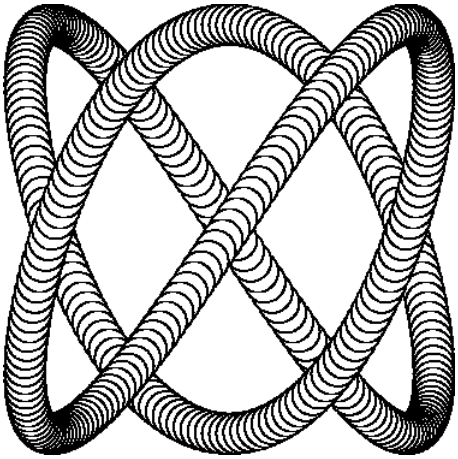


```
beginfig(149)
  def couleur(expr x) =
    if x>.5: (2*(1-x)) [red, blue];
    else:    (1-2x) [blue, red]
    fi
  enddef;
  for i=0 step .1 until 360:
    draw 2cm* (sind(2*i), cosd(3*i))
    withpen pencircle scaled 2bp
    withcolor couleur(i/360);
  endfor;
endfig;
```

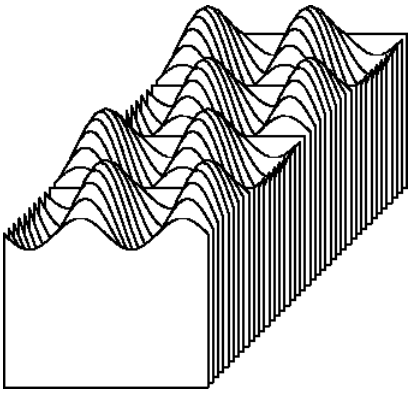


```
beginfig(150)
  for i=0 step 1 until 360:
    pair P;
```

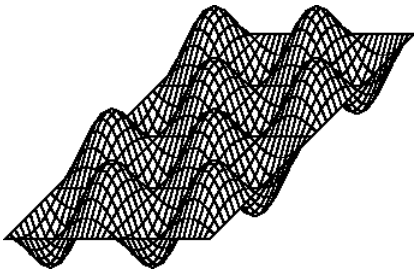
```
    P = 2cm* (sind(2*i), cosd(3*i));
    fill fullcircle scaled 4mm shifted P withcolor white;
    draw fullcircle scaled 4mm shifted P;
  endfor;
endfig;
```



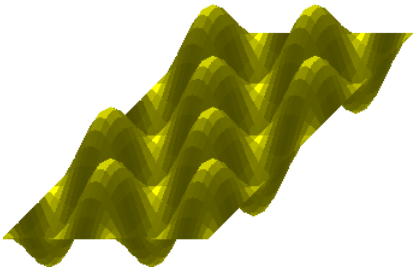
```
beginfig(151)
  for i=0 step 1 until 360:
    pair P;
    P = 2cm* (sind(2*i), cosd(3*i));
    fill fullcircle scaled 4mm shifted P withcolor white;
    draw fullcircle scaled 4mm shifted P;
  endfor;
  picture p;
  p:=nullpicture;
  for i=-180 step 1 until 180:
    pair P;
    P = 2cm* (sind(2*i), cosd(3*i));
    addto p contour fullcircle scaled 4mm shifted P withcolor white;
    addto p doublepath fullcircle scaled 4mm shifted P
      withpen pencircle scaled .5bp;
  endfor;
  clip p to (.5cm,2.5cm) -- (.5cm,1.5cm)--
            (-.5cm,1.5cm) -- (-.5cm,2.5cm)--cycle;
  draw p;
endfig;
```



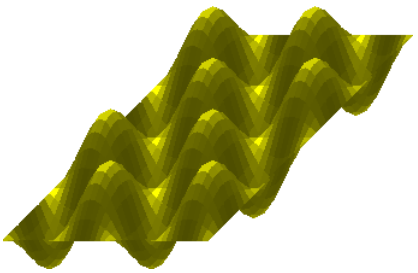
```
beginfig(152)
  u:=5mm;
  vardef project(expr x,y,z) =
    x*(-1,-1) + y*(1,0) + z*(0,1)
  enddef;
  vardef f(expr x,y) = sind(x/u*180)*sind(y/u*180)*u enddef;
  numeric m,M;
  m:=-2u; M:=2u;
  for i=m step .1u until M:
    path p;
    p =
      for j=m step .1u until M:
        project(i,j,f(i,j)) --
      endfor
      project(i,M,f(i,M));
    fill (project(i,m,f(i,m)) - (0,3u)) -- p --
         (project(i,M,f(i,M)) - (0,3u)) -- cycle
         withcolor white;
    draw (project(i,m,f(i,m)) - (0,3u)) -- p --
         (project(i,M,f(i,M)) - (0,3u)) -- cycle;
    draw p;
  endfor;
endfig;
```



```
beginfig(153)
  vardef project(expr x,y,z) =
    x*(-1,-1) + y*(1,0) + z*(0,1)
  enddef;
  vardef f(expr x,y) = sind(x/u*180)*sind(y/u*180)*u enddef;
  m:=-2u; M:=2u; inc:=.1u;
  for i=m step inc until M:
    for j=m step inc until M:
      path p;
      p = project(i,j,f(i,j)) --
          project(i,j+inc,f(i,j+inc)) --
          project(i+inc,j+inc,f(i+inc,j+inc)) --
          project(i+inc,j,f(i+inc,j)) --
          cycle;
      fill p withcolor white;
      draw p;
    endfor
  endfor;
endfig;
```

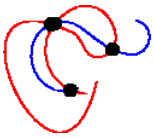


```
beginfig(154)
  vardef project(expr x,y,z) =
    x*(-1,-1) + y*(1,0) + z*(0,1)
  enddef;
  vardef f(expr x,y) = sind(x/u*180)*sind(y/u*180)*u enddef;
  numeric m,M,inc,couleur;
  m:=-2u; M:=2u; inc:=.1u;
  for i=m step inc until M:
    for j=m step inc until M:
      path p;
      p = project(i,j,f(i,j)) --
          project(i,j+inc,f(i,j+inc)) --
          project(i+inc,j+inc,f(i+inc,j+inc)) --
          project(i+inc,j,f(i+inc,j)) --
          cycle;
      dfdx := (f(i,j) - f(i+inc,j))/inc;
      dfdy := (f(i,j) - f(i,j+inc))/inc;
      couleur := 1/sqrt( dfdx**2 + dfdy**2 + 1);
      fill p withcolor couleur*(red+green);
    endfor
  endfor;
endfig;
```

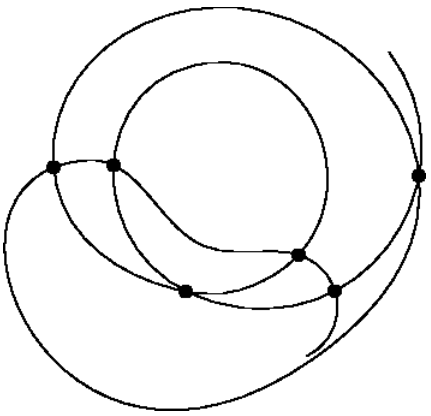


```
beginfig(155)
vardef f(expr x,y) = sind(x/u*180)*sind(y/u*180)*u enddef;

boolean dessine_fil_de_fer;  dessine_fil_de_fer := false;
color   dessine_couleur;     dessine_couleur    := red+green;

vardef dessine(suffix f)(expr xmin, xmax, xinc, ymin, ymax, yinc) =
  save i,j,p,dfdx,dfdy,project;
  vardef project(expr x,y,z) =
    x*(-1,-1) + y*(1,0) + z*(0,1)
  enddef;
  for i=m step inc until M:
```

```
    for j=m step inc until M:
      path p;
      p = project(i,j,f(i,j)) --
          project(i,j+inc,f(i,j+inc)) --
          project(i+inc,j+inc,f(i+inc,j+inc)) --
          project(i+inc,j,f(i+inc,j)) --
          cycle;
      dfdx := (f(i,j) - f(i+inc,j))/inc;
      dfdy := (f(i,j) - f(i,j+inc))/inc;
      couleur := 1/sqrt( dfdx**2 + dfdy**2 + 1);
      fill p withcolor couleur*dessine_couleur;
      if dessine_fil_de_fer: draw p fi;
    endfor
  endfor;
enddef;

  % Il faudrait pouvoir choisir diff^^e9rents
  % types d'^^ab ^^e9clairage ^^bb.
  vardef f(expr x,y) = sind(x/u*180)*sind(y/u*180)*u enddef;
  dessine(f,-2u,2u,.1u, -2u,2u,.1u);
endfig;
```

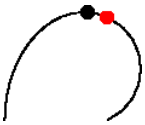


```
beginfig(156)
  vardef random_path (expr n) =
    save i, A ; numeric i; pair A[];
    for i=0 upto n:
      A[i] = (uniformdeviate(2u), uniformdeviate(2u));
    endfor;
    A[0]
    for i=1 upto n:
      .. A[i]
    endfor
  enddef;

  vardef intersections(expr p,q) =
    save a,b,N,i,j;
    numeric N,i,j;
    N:=10;
    for i=0 step length(p)/N until length(p):
      for j=0 step length(q)/N until length(p):
        numeric a,b;
        pair A;
        (a,b) = (subpath(i,i+length(p)/N) of p)
                intersectiontimes
                (subpath(j,j+length(q)/N) of q);
        if a <> -1:
          A = point a of subpath(i,i+length(p)/N) of p;
          show A;
          draw A withpen pencircle scaled 4bp;
        fi;
      endfor;
    endfor;
  enddef;

  path p,q;
  p:=random_path(4);
  q:=random_path(4);
  draw p withcolor red;
  draw q withcolor blue;
  intersections(p,q);
endfig;
```



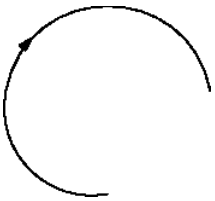
```
beginfig(157)
  vardef auto_intersections(expr p) =
    save a,b,N,i,j;
    numeric N,i,j;
    N:=100;
    for i=0 step length(p)/N until length(p):
      for j=i+2*length(p)/N
          step length(p)/N
          until length(p):
```

```
        numeric a,b;
        pair A;
        (a,b) = (subpath(i,i+length(p)/N) of p)
                intersectiontimes
                (subpath(j,j+length(p)/N) of p);
        if a <> -1:
          A = point a of subpath(i,i+length(p)/N) of p;
          show A;
          draw A withpen pencircle scaled 4bp;
        fi;
      endfor;
    endfor;
  enddef;

  u:=2cm;
  path p,q;
  p:=random_path(12);
  draw p;
  auto_intersections(p);
endfig;
```



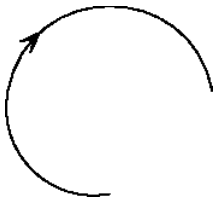
```
beginfig(158)
  path p;
  p := (0,0){up} .. (1cm,1cm) .. (1cm,0);
  draw p;
  draw point 1/2length(p) of p
    withpen pencircle scaled 4bp withcolor red;
  draw point (arctime (1/2 arclength(p)) of p) of p
    withpen pencircle scaled 4bp;

endfig;
```

```
beginfig(159)
  vardef milieu(expr p) =
    save l,i,tot,A,B;
    numeric l,tot,i;
    pair A,B;
    tot := longueur(p);
    l:=0;
    B := point 0 of p;
    for i:=0 step .01 until length(p):
      A := B;
      B := point i of p;
      l := l+abs(B-A);
      exitif l > 1/2 tot;
    endfor;
    1/2[A,B]
  enddef;

  path p;
  p := (0,0){up} .. (1cm,1cm) .. (1cm,0);
  draw p;
  draw point 1/2length(p) of p withpen pencircle scaled 4bp withcolor red;
  draw milieu(p) withpen pencircle scaled 4bp;
endfig;
```

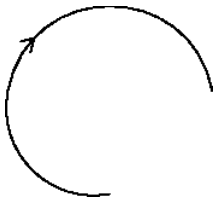


```
beginfig(160)
  vardef milieu_time(expr p) =
    save l,i,tot,A,B,t;
    numeric l,tot,i,t;
    pair A,B;
    tot := longueur(p);
    l:=0;
    B := point 0 of p;
    for i:=0 step .01 until length(p):
      t:=i;
      A := B;
      B := point i of p;
      l := l+abs(B-A);
      exitif l > 1/2 tot;
    endfor;
    t % Pas de point-virgule
  enddef;
```

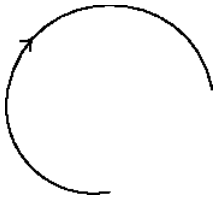
```
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := milieu(p);
    u := unitvector(direction milieu_time(p) of p);
    A -- (A - ahlength*u rotated 15) --
    (A - ahlength*u rotated -15) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



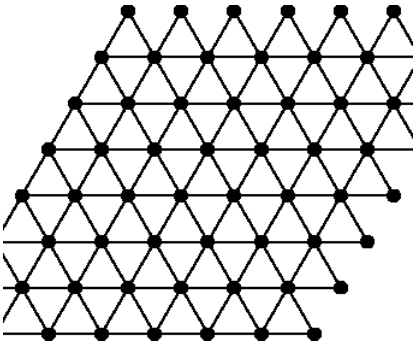
```
beginfig(161)
  save arrowhead;
  vardef arrowhead expr p =
    save A,B,u; pair A,B,u;
    A := milieu(p);
    B := p intersectionpoint
         (fullcircle scaled ahlength shifted A);
    u := unitvector(direction milieu_time(p) of p);
    A -- (A - ahlength*u rotated 30) -- B --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



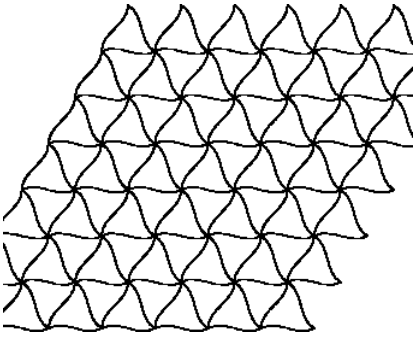
```
beginfig(162)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := milieu(p);
    u := unitvector(direction milieu_time(p) of p);
    A -- (A - ahlength*u rotated 30) -- A --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



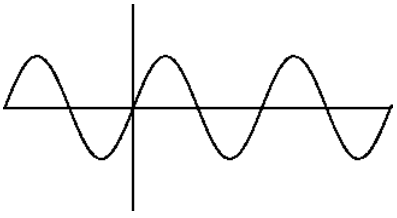
```
beginfig(163)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u,a,b; pair A,u; path a,b;
    A := milieu(p);
    u := unitvector(direction milieu_time(p) of p);
    a := A{-u} .. (A - ahlength*u rotated 30);
    b := A{-u} .. (A - ahlength*u rotated -30);
    ( a & reverse(a) & b & reverse(b) ) --cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



```
beginfig(164)
  pair A,B,C;
  C = 3mm*up;
  A = C rotated 120;
  B = C rotated -120;
  picture pic ;
  pic:=nullpicture;
  addto pic doublepath A--B--C--cycle withpen currentpen;
  addto pic doublepath A withpen pencircle scaled 4bp;
  addto pic doublepath B withpen pencircle scaled 4bp;
  addto pic doublepath C withpen pencircle scaled 4bp;
  for i=-3 upto 3:
    for j=-3 upto 3:
      draw pic shifted( i*(B-A) + j*(C-A) );
    endfor;
  endfor;
  clip currentpicture to (-2cm,-2cm)--(2cm,-2cm)--(2cm,2cm)--(-2cm,2cm)--cycle;
endfig;
```

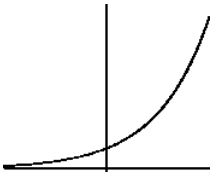


```
beginfig(165)
  pair A,B,C;
  C = 3mm*up;
  A = C rotated 120;
  B = C rotated -120;
  picture pic ;
  pic:=nullpicture;
  path p;
  p := A{(C-A) rotated 30} .. C{(C-A) rotated 30};
  addto pic doublepath p withpen currentpen;
  addto pic doublepath p rotated 120 withpen currentpen;
  addto pic doublepath p rotated -120 withpen currentpen;
  for i=-3 upto 3:
    for j=-3 upto 3:
      draw pic shifted( i*(B-A) + j*(C-A) );
    endfor;
  endfor;
  clip currentpicture to (-2cm,-2cm)--(2cm,-2cm)--(2cm,2cm)--(-2cm,2cm)--cycle;
endfig;
```

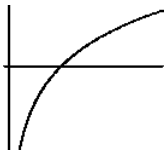


```
beginfig(166)
  ux:=2mm;
  uy:=5mm;
  numeric xmin, xmax, ymin, ymax, M;
  xmin := -6.3; xmax := 12.6;
  ymin := -2;   ymax := 2;
  M := 100;
  draw (ux*xmin,0) -- (ux*xmax,0);
  draw (0,uy*ymin) -- (0,uy*ymax);
  pair a[];
  for i=0 upto M:
    a[i] := (
```

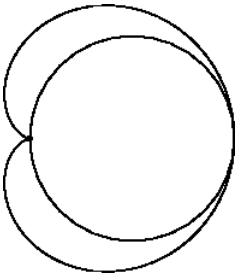
```
      xmin + (i/M)*(xmax-xmin),
      sind(180/3.14*( xmin + (i/M)*(xmax-xmin) ))
    ) xscaled ux yscaled uy;
  endfor;
  draw a[0] for i=1 upto M: --a[i] endfor;
endfig;
```



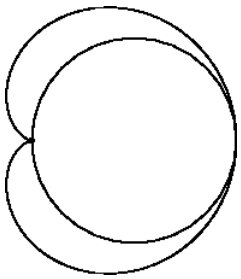
```
beginfig(167)
  ux:=5mm;
  uy:=2mm;
  numeric xmin, xmax, ymin, ymax, M;
  xmin := -2; xmax := 2;
  ymin := -.1;   ymax := 8;
  M := 100;
  draw (ux*xmin,0) -- (ux*xmax,0);
  draw (0,uy*ymin) -- (0,uy*ymax);
  pair a[];
  for i=0 upto M:
    a[i] := (
      xmin + (i/M)*(xmax-xmin),
      mexp(256*( xmin + (i/M)*(xmax-xmin) ))
    ) xscaled ux yscaled uy;
  endfor;
  draw a[0] for i=1 upto M: --a[i] endfor;
endfig;
```



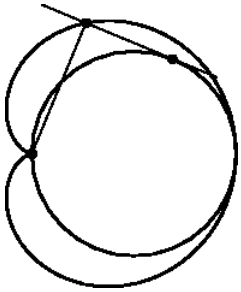
```
beginfig(168)
  ux:=5mm;
  uy:=5mm;
  numeric xmin, xmax, ymin, ymax, M;
  xmin := .2; xmax := 3;
  ymin := -1.6;   ymax := 1.2;
  M := 100;
  draw (ux*-.1,0) -- (ux*xmax,0);
  draw (0,uy*ymin) -- (0,uy*ymax);
  pair a[];
  for i=0 upto M:
    a[i] := (
      xmin + (i/M)*(xmax-xmin),
      (1/256)*mlog(( xmin + (i/M)*(xmax-xmin) ))
    ) xscaled ux yscaled uy;
  endfor;
  draw a[0] for i=1 upto M: --a[i] endfor;
endfig;
```



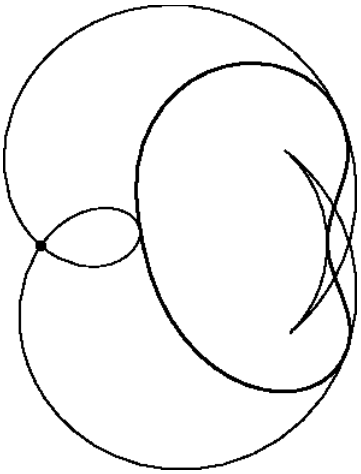
```
beginfig(169)
  path p;
  p  = fullcircle scaled 2cm;
  z0 = (-1cm,0);
  draw p;
  draw z0 withpen pencircle scaled 2pt;

  pair A[];
  for i=0 step length(p)/100 until length(p):
    pair M,N;
    M = point i of p;
    N-M = whatever * direction i of p;
    N-z0 = whatever * direction i of p rotated 90;
    A[i] := N;
  endfor;
  draw for i=0 step length(p)/100 until length(p):
    A[i] ..
  endfor cycle;
endfig;
```



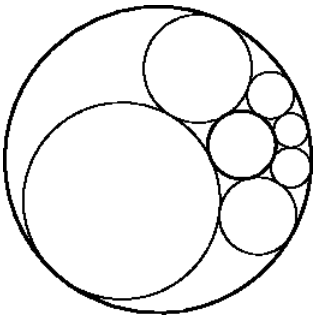
```
beginfig(170)
vardef cardiodide(expr p, O) =
  save i,M,N;
  numeric i;
  for i=0 step length(p)/100 until length(p):
    hide(
      pair M,N;
      M = point i of p;
      N-M = whatever * direction i of p;
      N-O = whatever * direction i of p rotated 90;
    )
    N ..
  endfor cycle
enddef;

  path p;
  p  = fullcircle scaled 2cm;
  z0 = (-1cm,0);
  draw p;
  draw z0 withpen pencircle scaled 2pt;
  draw cardiodide(p,z0);
endfig;
```



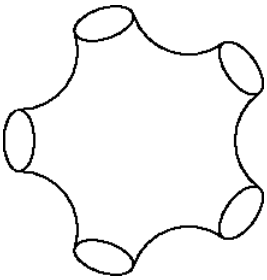
```
beginfig(171)
vardef cardiodide(expr p, O) =
  save i,M,N;
  numeric i;
  for i=0 step length(p)/100 until length(p):
    hide(
      pair M,N;
      M = point i of p;
      N-M = whatever * direction i of p;
      N-O = whatever * direction i of p rotated 90;
    )
    N ..
  endfor cycle
enddef;

  path p;
  p  = fullcircle scaled 2cm;
  z0 = (-1cm,0);
  pickup pencircle scaled 1pt
  draw p;
  draw z0 withpen pencircle scaled 3pt;
  draw cardiodide(p,z0);
  pickup pencircle scaled .4pt
  pair M,N;
  i:=1.5;
  M = point i of p;
  N-M  = whatever * direction i of p;
  N-z0 = whatever * direction i of p rotated 90;
  draw z0--N;
  draw (-1/2)[N,M]--(3/2)[N,M];
  draw N withpen pencircle scaled 3pt;
  draw M withpen pencircle scaled 3pt;
endfig;
```



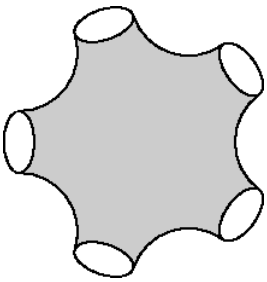
```
beginfig(172)
  path p;
  p := (-1cm,0) .. (1cm,-1cm) .. (8mm,0)
       .. (1cm,1cm) .. (-1cm,1cm) .. cycle;
  z0 = (-2cm,0);
  draw p withpen pencircle scaled 1bp;
  draw z0 withpen pencircle scaled 3pt;
  draw cardiodide(p,z0);
endfig;
```



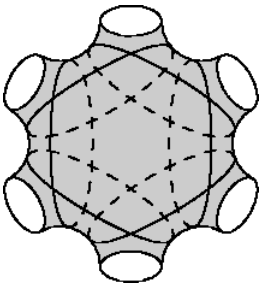
```
beginfig(173)
vardef inversion (expr O,k,M) =
  if pair M:
    (O + k*unitvector(M-O)/abs(M-O))
  elseif path M:
    for i=0 step length(M)/100 until length(M):
      inversion(O,k,point i of M) ..
    endfor
    cycle
  fi
enddef;

  u:=4cm;
  path p[],A,B;

  z0 = (5u,0) rotated 10;

  A = fullcircle scaled 2u;
  B = A scaled 3;
  draw inversion( z0, 2 (u**2), A )
       withpen pencircle scaled 1pt;
  draw inversion( z0, 2 (u**2), B )
       withpen pencircle scaled 1pt;

  p0 = fullcircle scaled 2u shifted (2u,0);
  for i=0 upto 5:
    if i<>0:
      p[i] = p[i-1] rotated (360/6);
    fi;
    draw inversion( z0, 2 (u**2), p[i] );
  endfor;
endfig;
```



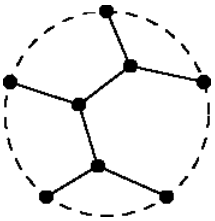
```
beginfig(174)
  def curve(expr p,i,q,j,t) =
    point i of p {direction i of p} ..
    tension t ..
    point j of q {direction j of q}
  enddef;

  vardef sphere_with_holes (expr n) =
    save i;
    c[0] = fullcircle xscaled u yscaled 2u
                      shifted (4u,0) rotated (360/(2n)) ;
    draw c[0];
    for i=1 upto n-1:
      c[i] = c[i-1] rotated (360/n);
      draw c[i];
    endfor;
    l[0] = curve(c[0], 2, c[1], -2, 1);
    draw l[0];
    for i=1 upto n-1:
      l[i] = l[i-1] rotated (360/n);
      draw l[i];
    endfor;
  enddef;

  u:=3mm;
  path c[], l[];
  sphere_with_holes(5);
endfig;
```



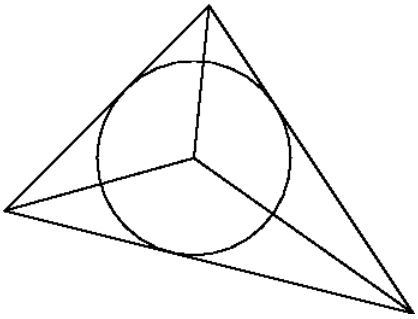
```
beginfig(175)
  def curve(expr p,i,q,j,t) =
    point i of p {direction i of p} ..
    tension t ..
    point j of q {direction j of q}
  enddef;

  vardef sphere_with_holes (expr n) =
    save i;
    c[0] = fullcircle xscaled u yscaled 2u
                      shifted (4u,0) rotated (360/(2n)) ;
    for i=1 upto n-1:
      c[i] = c[i-1] rotated (360/n);
    endfor;
    l[0] = curve(c[0], 2, c[1], -2, 1);
    for i=1 upto n-1:
      l[i] = l[i-1] rotated (360/n);
    endfor;
    fill for i=0 upto n-1:
      ( reverse subpath(2,6) of c[i] ) &
      l[i] &
    endfor
    % To turn it into a cycle (ugly)
    point length(l[n-1]) of l[n-1] -- cycle
    withcolor .8white;
    for i=0 upto n-1:
      draw c[i]; draw l[i];
    endfor;
  enddef;

  u:=3mm;
  path c[], l[];
  sphere_with_holes(5);
endfig;
```



```
beginfig(176)
  path c[], l[];
  sphere_with_holes(6);
  def ellipse(expr a,b,c,d,e) =
    draw curve(a,b,c,d,e) ;
    draw curve(c,d,a,b,e) dashed evenly;
  enddef;

  ellipse (l[0], 2/3, l[2], 1/3, 4);
  ellipse (l[1], 2/3, l[3], 1/3, 4);
  ellipse (l[2], 2/3, l[4], 1/3, 4);
  ellipse (l[3], 2/3, l[5], 1/3, 4);
  ellipse (l[4], 2/3, l[0], 1/3, 4);
  ellipse (l[5], 2/3, l[1], 1/3, 4);
endfig;
```

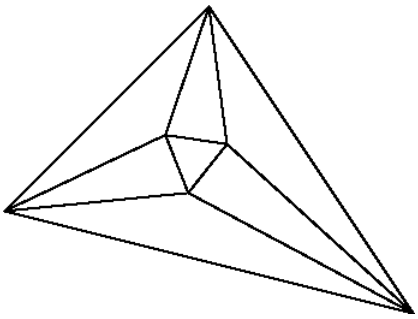


```
beginfig(177)
  vardef bar(expr m,a,b,c) =
    m = 1/3a + 1/3b + 1/3c
  enddef;
  vardef dbar(expr m,a,b,c) =
    draw m--a; draw m--b; draw m--c;
    draw m withpen pencircle scaled 4bp;
  enddef;

  pair P[], A,B,C;
  for i=0 upto 4:
    P[i] = 1cm * up rotated (i*360/5);
    draw P[i] withpen pencircle scaled 4bp;
  endfor;
  bar(A, P[0], P[4], B);
  bar(B, A,    P[1], C);
  bar(C, B,    P[2], P[3]);
  draw fullcircle scaled 2cm dashed evenly;
  dbar(A, P[0], P[4], B);
  dbar(B, A,    P[1], C);
  dbar(C, B,    P[2], P[3]);
endfig;
```

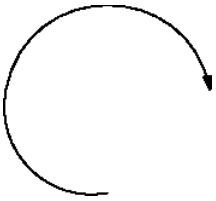


```
beginfig(178)
% M est sur la bissectrice de l'angle A
vardef bissectrice(expr M,A,B,C) =
  (M-A) = whatever * (
    (A-C) rotated 1/2(angle(B-A) - angle(C-A)))
enddef;

% Le cercle inscrit
vardef cercle_inscrit(expr A,B,C) =
  save M, h; pair M, h;
  bissectrice(M,A,B,C);
  bissectrice(M,B,C,A);
  M-h = whatever * (B-C) rotated 90;
  h = whatever[B,C];
  fullcircle scaled 2 length(M-h) shifted M
enddef;
```

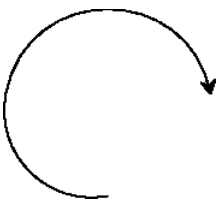
```
  pair A,B,C,M;
  u:=2cm;
  A=(0,0); B=(2u,-.5u); C=(u,u);
  draw A--B--C--cycle;
  bissectrice(M, A,B,C);
  bissectrice(M, B,C,A);
  draw M--A; draw M--B; draw M--C;
  draw cercle_inscrit(A,B,C);
endfig;
```



```
beginfig(179)
vardef premiere_trisectrice (expr M, A,B,C) =
  (M-A) = whatever * ( (A-B) rotated 1/3 (angle(C-A) - angle(B-A)) )
enddef;

vardef deuxieme_trisectrice (expr M, A,B,C) =
  (M-A) = whatever * ( (A-B) rotated 2/3 (angle(C-A) - angle(B-A)) )
enddef;

  pair A,B,C,M[];
  u:=2cm;
  A=(0,0); B=(2u,-.5u); C=(u,u);
  draw A--B--C--cycle;

  premiere_trisectrice(M1,A,B,C);
  deuxieme_trisectrice(M1,B,C,A);
  premiere_trisectrice(M2,B,C,A);
  deuxieme_trisectrice(M2,C,A,B);
  premiere_trisectrice(M3,C,A,B);
  deuxieme_trisectrice(M3,A,B,C);

  draw M1--A; draw M1--B;
  draw M2--B; draw M2--C;
  draw M3--C; draw M3--A;

  draw M1--M2--M3--cycle;
endfig;
```

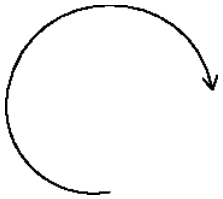


```
beginfig(180)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := point length(p) of p;
    u := unitvector(direction length(p) of p);
    A -- (A - ahlength*u rotated 15) --
    (A - ahlength*u rotated -15) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```

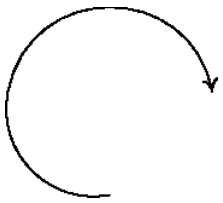


```
beginfig(181)
  save arrowhead;
  vardef arrowhead expr p =
    save A,B,u; pair A,B,u;
    A := point length(p) of p;
```

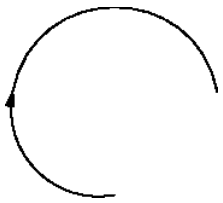
```
    B := p intersectionpoint
         (fullcircle scaled ahlength shifted A);
    u := unitvector(direction length(p) of p);
    A -- (A - ahlength*u rotated 30) -- B --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



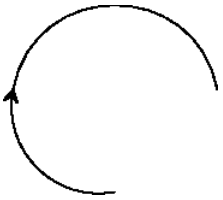
```
beginfig(182)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := point length(p) of p;
    u := unitvector(direction length(p) of p);
    A -- (A - ahlength*u rotated 30) -- A --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



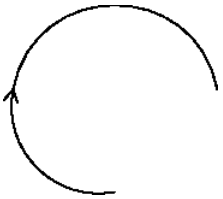
```
beginfig(183)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u,a,b; pair A,u; path a,b;
    A := point length(p) of p;
    u := unitvector(direction length(p) of p);
    a := A{-u} .. (A - ahlength*u rotated 30);
    b := A{-u} .. (A - ahlength*u rotated -30);
    ( a & reverse(a) & b & reverse(b) ) --cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



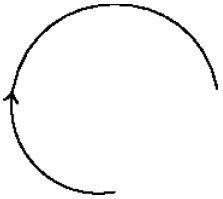
```
beginfig(184)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := point 1/2length(p) of p;
    u := unitvector(direction 1/2length(p) of p);
    A -- (A - ahlength*u rotated 15) --
    (A - ahlength*u rotated -15) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



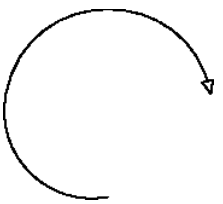
```
beginfig(185)
  save arrowhead;
  vardef arrowhead expr p =
    save A,B,u; pair A,B,u;
    A := point 1/2length(p) of p;
    B := p intersectionpoint
         (fullcircle scaled ahlength shifted A);
    u := unitvector(direction 1/2length(p) of p);
    A -- (A - ahlength*u rotated 30) -- B --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



```
beginfig(186)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u; pair A,u;
    A := point 1/2length(p) of p;
    u := unitvector(direction 1/2length(p) of p);
    A -- (A - ahlength*u rotated 30) -- A --
    (A - ahlength*u rotated -30) -- cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```

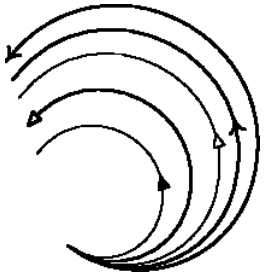


```
beginfig(187)
  save arrowhead;
  vardef arrowhead expr p =
    save A,u,a,b; pair A,u; path a,b;
    A := point 1/2length(p) of p;
    u := unitvector(direction 1/2length(p) of p);
    a := A{-u} .. (A - ahlength*u rotated 30);
    b := A{-u} .. (A - ahlength*u rotated -30);
    ( a & reverse(a) & b & reverse(b) ) --cycle
  enddef;

  u:=1cm;
  drawarrow (0,0) .. (-u,u) .. (u,u);
endfig;
```

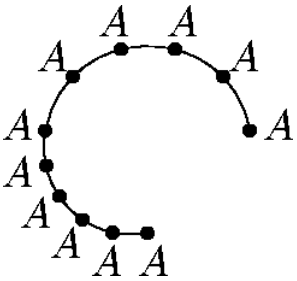


```
beginfig(188)
  def drawwhitearrow expr p = _apth:=p; _finwhitearr enddef;

  def _finwhitearr text t =
    draw _apth t;
    fill arrowhead _apth  t withcolor white;
    draw arrowhead _apth  t
  enddef;
```

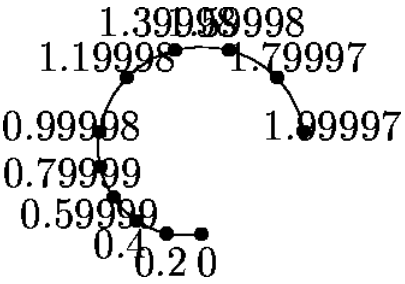
```
  u:=1cm;
  drawwhitearrow (0,0) .. (-u,u) .. (u,u);
endfig;
```



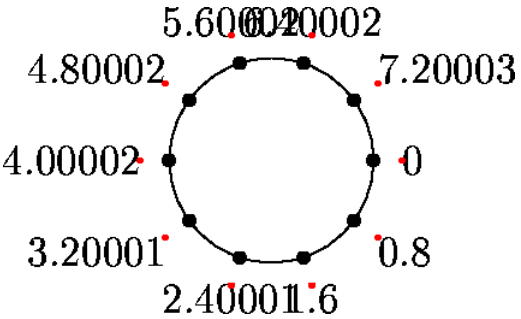
```
beginfig(189)
  def draw_white_arrow expr p = _apth:=p; _fin_white_arr enddef;
  def _fin_white_arr text t =
    draw _apth t;
    fill arrowhead _apth withcolor white;
    draw arrowhead _apth  t
  enddef;

  def draw_middle_arrow expr p = _apth:=p; _fin_middle_arr enddef;
  def _fin_middle_arr text t =
    draw _apth t;
    filldraw arrowhead_middle _apth  t
  enddef;
  vardef arrowhead_middle expr p =
    save A,u; pair A,u;
    A := point (arctime (.5arclength p) of p) of p;
    u := unitvector(direction (arctime (.5arclength p) of p) of p);
    A -- (A - ahlength*u rotated (.5ahangle) ) --
    (A - ahlength*u rotated (-.5ahangle) ) -- cycle
  enddef;

  def draw_middle_white_arrow expr p = _apth:=p; _fin_middle_white_arr enddef;
  let draw_white_middle_arrow = draw_middle_white_arrow;
  def _fin_middle_white_arr text t =
    draw _apth t;
    fill arrowhead_middle _apth  t withcolor white;
    draw arrowhead_middle _apth  t
  enddef;

  def draw_other_arrow expr p = _apth:=p; _fin_other_arr enddef;
  def _fin_other_arr text t =
    draw _apth t;
    draw arrowhead_other _apth  t
  enddef;
  vardef arrowhead_other expr p =
    save A,u,a,b; pair A,u; path a,b;
    A := point (length p) of p;
    u := unitvector(direction (length p) of p);
    a := A{-u} .. (A - ahlength*u rotated 30);
    b := A{-u} .. (A - ahlength*u rotated -30);
    ( a & reverse(a) & b & reverse(b) ) --cycle
  enddef;

  def draw_other_middle_arrow expr p = _apth:=p; _fin_other_middle_arr enddef;
  let draw_middle_other_arrow = draw_other_middle_arrow;
  def _fin_other_middle_arr text t =
    draw _apth t;
    draw arrowhead_other_middle _apth  t
  enddef;
  vardef arrowhead_other_middle expr p =
    save A,u,a,b; pair A,u; path a,b;
    A := point (arctime (.5arclength p) of p) of p;
    u := unitvector(direction (arctime (.5arclength p) of p) of p);
    a := A{-u} .. (A - ahlength*u rotated 30);
    b := A{-u} .. (A - ahlength*u rotated -30);
    ( a & reverse(a) & b & reverse(b) ) --cycle
  enddef;

  path p;
  p := halfcircle scaled 2cm;
  p := (0,0) .. (3cm,1cm) .. (-1cm,3cm);
  p := p scaled .3;
  draw_middle_arrow p;
  draw_white_arrow p scaled 1.3 withpen pencircle scaled 1bp;
  draw_white_middle_arrow p scaled 1.6;
  draw_other_middle_arrow p scaled 1.8 withpen pencircle scaled 1bp;
  draw_other_arrow p scaled 2 withpen pencircle scaled 1bp;
endfig;
```




```
beginfig(190)
  vardef mylabel(expr pic, p, t) =
    save A; pair A;
    A = point t of p +
        8bp * unitvector(direction t of p) rotated 90;
    label(pic, A);
  enddef;
  path p; u:=1cm;
  p = (0,0)..(-u,u)..(u,u);
  draw p;
  for i=0 step .2 until length(p):
    draw point i of p withpen pencircle scaled 4bp;
    mylabel(btex $A$ etex,p,i);
  endfor;
endfig;
```

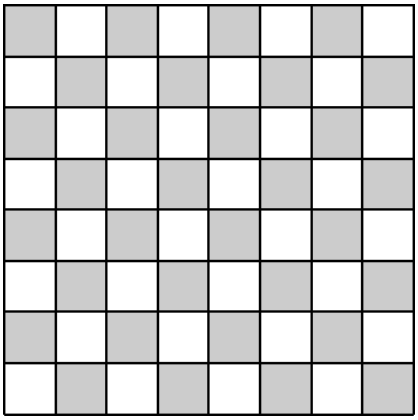



```
beginfig(191)
  vardef mylabel(expr pic, p, t) =
    save A; pair A;
    A = point t of p +
        8bp * unitvector(direction t of p) rotated 90;
    label(pic, A);
  enddef;
  path p; u:=1cm;
  p = (0,0)..(-u,u)..(u,u);
  draw p;
  for i=0 step .2 until length(p):
    draw point i of p withpen pencircle scaled 4bp;
    mylabel(TEX decimal(i),p,i);
  endfor;
endfig;
```

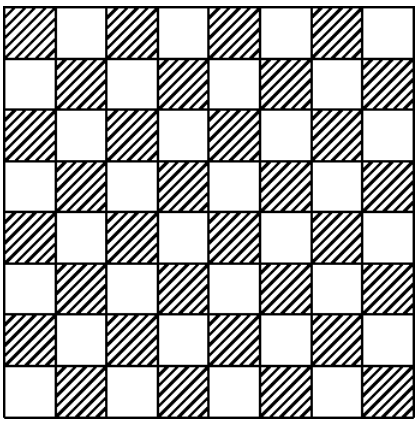



```
beginfig(192)
  vardef mylabel(expr pic, p, t) =
    save A,a; pair A; numeric a;
    a := angle(direction t of p rotated 90);
    show(a);
    a := a + 45/2;
    if a>180: a := a - 360 fi;
    save ll, lr, ul, ur;
    pair ul,ur,ll,lr;
    ll := llcorner pic;
    lr := lrcorner pic;
    ul := ulcorner pic;
    ur := urcorner pic;
    A = point t of p +
        8bp * unitvector(direction t of p) rotated 90;
    label(pic shifted ll, A +
      if     (a >=    0) and (a <=   45): 1/2(ur-ul)
      elseif (a >=   45) and (a <=   90): 1/2(ur-ul) + 1/2(ur-lr)
      elseif (a >=   90) and (a <=  135): 1/2(ur-lr)
      elseif (a >=  135) and (a <=  180): 1/2(ll-lr) + 1/2(ur-lr)
```

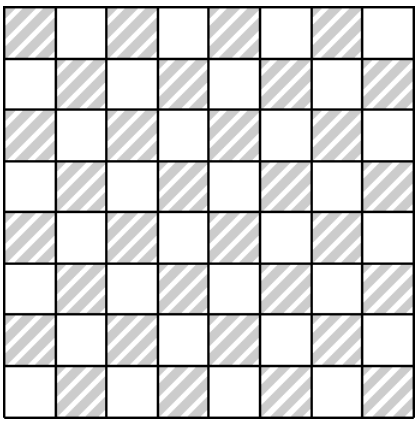
```
      elseif (a >= -180) and (a <= -135): 1/2(ll-lr)
      elseif (a >= -135) and (a <=  -90): 1/2(ll-ul) + 1/2(ll-lr)
      elseif (a >=  -90) and (a <=  -45): 1/2(ll-ul)
      elseif (a >=  -45) and (a <=    0): 1/2(ll-ul) - 1/2(ll-lr)
      else: hide(show "BUG") (0,0)
      fi);
    draw A withpen pencircle scaled 2bp withcolor red;
  enddef;
  path p; u:=1cm;
  p = (0,0)..(-u,u)..(u,u);
  p := reverse fullcircle scaled 2u;
  draw p;
  for i=0 step length(p)/10 until length(p):
    draw point i of p withpen pencircle scaled 4bp;
    mylabel(TEX decimal(i),p,i);
  endfor;
endfig;
```



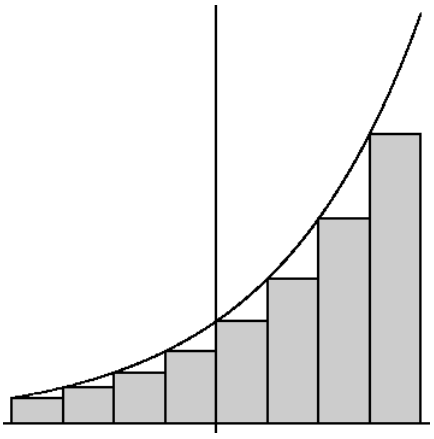
```
beginfig(193)
  n:=8;
  u:=5mm;
  for i=0 upto n-1:
    for j=0 upto n-1:
      if odd(i+j):
        fill (0,0)--(u,0)--(u,u)--(0,u)--cycle
             shifted (i*u,j*u) withcolor .8white;
      fi;
    endfor;
  endfor;
  for i=0 upto n:
    draw (0,i*u)--(n*u,i*u);
    draw (i*u,0)--(i*u,n*u);
  endfor;
endfig;
```



```
beginfig(194)
  n:=8;
  u:=5mm;
  for i=0 upto n-1:
    for j=0 upto n-1:
      if odd(i+j):
        for k=0 step u/5 until u:
          draw ( (k,0)--(u,u-k) )
               shifted (i*u,j*u);
          draw ( (0,k)--(u-k,u) )
               shifted (i*u,j*u);
        endfor;
      fi;
    endfor;
  endfor;
  for i=0 upto n:
    draw (0,i*u)--(n*u,i*u);
    draw (i*u,0)--(i*u,n*u);
  endfor;
endfig;
```

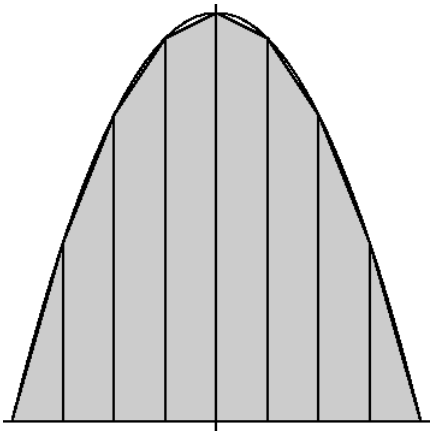


```
beginfig(195)
  n:=8;
  u:=5mm;
  for i=0 upto n-1:
    for j=0 upto n-1:
      if odd(i+j):
        for k=0 step u/5 until 4/5 u:
          if odd(k*5/u):
            fill ( (k,0)--(u,u-k)--(u,u-k-u/5)--
                   (k+u/5,0)--cycle )
                 shifted (i*u,j*u) withcolor .8white;
          else:
            fill ( (0,k)--(u-k,u)--(u-k-u/5,u)--
                   (0,k+u/5)--cycle )
                 shifted (i*u,j*u) withcolor .8white;
          fi;
        endfor;
      fi;
    endfor;
  endfor;
  for i=0 upto n:
    draw (0,i*u)--(n*u,i*u);
    draw (i*u,0)--(i*u,n*u);
  endfor;
endfig;
```

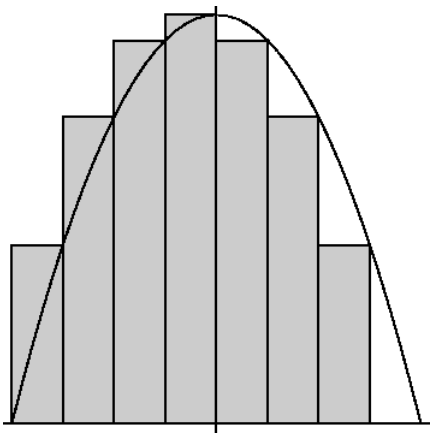


```
beginfig(196)
  vardef trace (suffix f)(expr a,b,inc) =
    save i; numeric i;
    for i=a step inc until b:
      (i*1cm, f(i)*1cm) ..
    endfor (b*1cm, f(b)*1cm)
  enddef;

  vardef axes =
    save p; picture p;
    p:=nullpicture;
    addto p doublepath (-infinity,0)--(infinity,0) withpen currentpen;
    addto p doublepath (0,-infinity)--(0,infinity) withpen currentpen;
    clip p to bbox currentpicture;
    draw p;
  enddef;

  vardef trace_rectangles_left (suffix f)(expr a,b,inc) =
    save i; numeric i;
    for i=a step inc until b-inc:
      path p;
      p = (i,0)--(i+inc,0)--(i+inc,f(i))--(i,f(i))--cycle;
      p := p scaled 1cm;
      fill p withcolor .8*white;
      draw p;
    endfor;
  enddef;
```

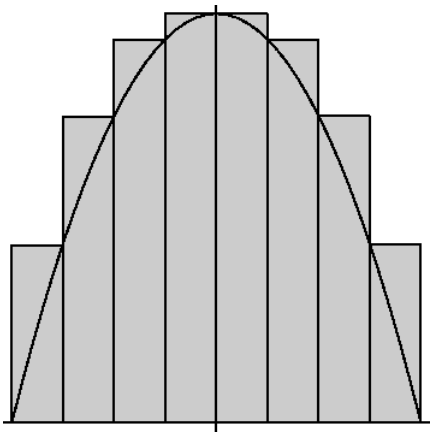
```
  vardef f(expr x) = 2 ** x enddef;

  trace_rectangles_left(f,-2,2,.5);
  draw trace(f, -2,2,.1);
  axes;
endfig;
```



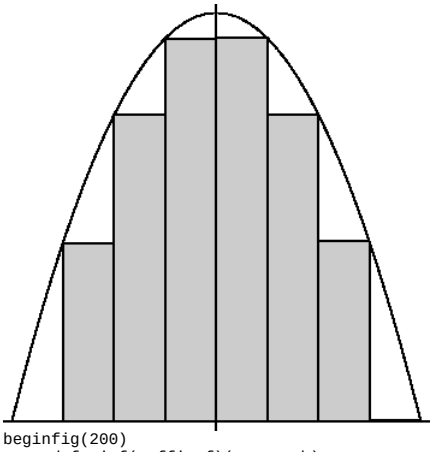
```
beginfig(197)
  vardef trace_trapezes (suffix f)(expr a,b,inc) =
    save i; numeric i;
    for i=a step inc until b-inc:
      path p;
      p = (i,0)--(i+inc,0)--(i+inc,f(i+inc))--(i,f(i))--cycle;
      p := p scaled 1cm;
      fill p withcolor .8*white;
      draw p;
    endfor;
  enddef;

  vardef f(expr x) = 4 - x**2 enddef;

  trace_trapezes(f,-2,2,.5);
  draw trace(f, -2,2,.1);
  axes;
endfig;
```



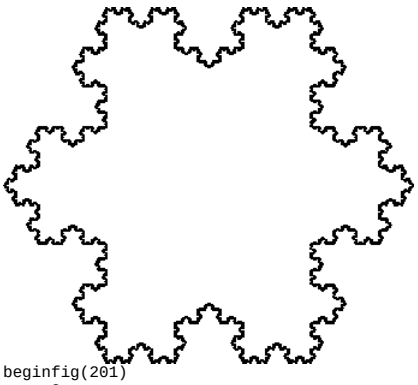
```
beginfig(198)
  vardef trace_rectangles_right (suffix f)(expr a,b,inc) =
    save i; numeric i;
    for i=a step inc until b-inc:
      path p;
      p = (i,0)--(i+inc,0)--(i+inc,f(i+inc))--(i,f(i+inc))--cycle;
      p := p scaled 1cm;
      fill p withcolor .8*white;
      draw p;
    endfor;
  enddef;

  trace_rectangles_right(f,-2,2,.5);
  draw trace(f, -2,2,.1);
  axes;
endfig;
```



```
beginfig(199)
  vardef maxf(suffix f)(expr a,b) =
    save m,i; numeric m,i;
    m:=f(a);
    for i=a step (b-a)/100 until b:
      if m
```

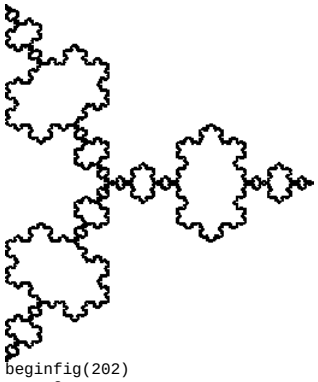


```
beginfig(200)
  vardef minf(suffix f)(expr a,b) =
    save m,i; numeric m,i;
    m:=f(a);
    for i=a step (b-a)/100 until b:
      if m>f(i): m:=f(i); fi;
    endfor;
    m
  enddef;

  vardef trace_rectangles_min (suffix f)(expr a,b,inc) =
    save i; numeric i;
    for i=a step inc until b-inc:
      path p; numeric m;
      m:=minf(f,i,i+inc);
      p = (i,0)--(i+inc,0)--(i+inc,m)--(i,m)--cycle;
      p := p scaled 1cm;
      fill p withcolor .8*white;
      draw p;
    endfor;
  enddef;

  vardef f(expr x) = 4 - x**2 enddef;

  trace_rectangles_min(f,-2,2,.5);
  draw trace(f, -2,2,.1);
  axes;
endfig;
```

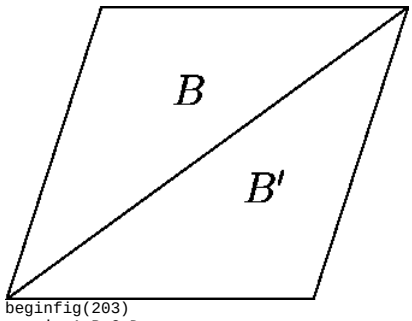


```
beginfig(201)
  u:=2cm;
  vardef koch(expr A,B,n) =
```

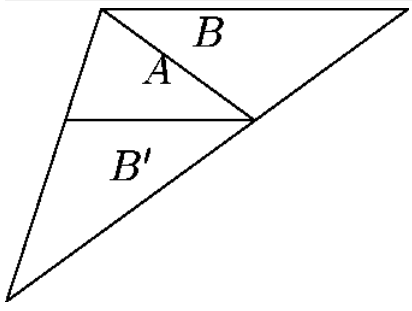
```
    save C; pair C;
    C = A rotatedaround(1/3[A,B], 120);
    if n>0:
      koch( A,        1/3[A,B], n-1);
      koch( 1/3[A,B], C,        n-1);
      koch( C,        2/3[A,B], n-1);
      koch( 2/3[A,B], B,        n-1);
    else:
      draw A--1/3[A,B]--C--2/3[A,B]--B;
    fi;
  enddef;
  z0=(u,0);
  z1=z0 rotated 120;
  z2=z1 rotated 120;
  koch( z0, z1, 4 );
  koch( z1, z2, 4 );
  koch( z2, z0, 4 );
endfig;
```



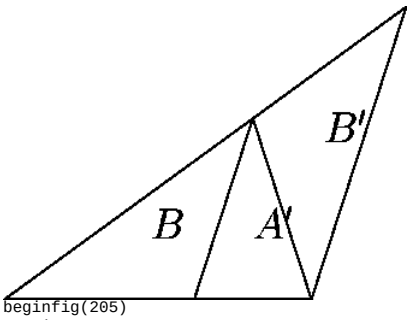
```
beginfig(202)
  u:=2cm;
  vardef koch(expr A,B,n) =
    save C; pair C;
    C = A rotatedaround(1/3[A,B], -120);
    if n>0:
      koch( A,        1/3[A,B], n-1);
      koch( 1/3[A,B], C,        n-1);
      koch( C,        2/3[A,B], n-1);
      koch( 2/3[A,B], B,        n-1);
    else:
      draw A--1/3[A,B]--C--2/3[A,B]--B;
    fi;
  enddef;
  z0=(u,0);
  z1=z0 rotated 120;
  z2=z1 rotated 120;
  koch( z0, z1, 4 );
  koch( z1, z2, 4 );
  koch( z2, z0, 4 );
endfig;
```



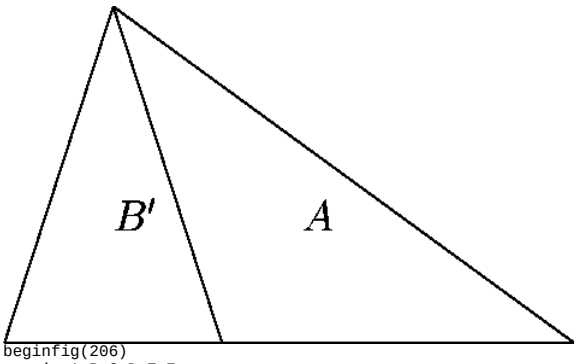

```
beginfig(203)
  pair A,B,C,D;
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  draw A--B--C--D--cycle;
  draw A--C;
  draw btex $B'$ etex shifted 1/3 (A+B+C);
  draw btex $B$ etex shifted 1/3 (A+D+C);

endfig;
```



```
beginfig(204)
  pair A,B,C,D,E,F;
  numeric d[];
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  d[0] := 1;
  d[1] := sqrt( 2*(1+cosd(72)) );
  d[2] := sqrt( 2*(1-cosd(36)) );
  A := A;
  B := C;
  C := D;
  draw A--B--C--cycle;
       E := (d1/(d0+d1)) [A,C];
       F := (d0/(d0+d2)) [A,B];
  draw E--C--F--cycle;
  draw btex $A$ etex shifted 1/3(E+C+F);
  draw B--C--F--cycle;
  draw btex $B$ etex shifted 1/3(B+C+E);
  draw E--F--A--cycle;
  draw btex $B'$ etex shifted 1/3(E+F+A);
endfig;
```

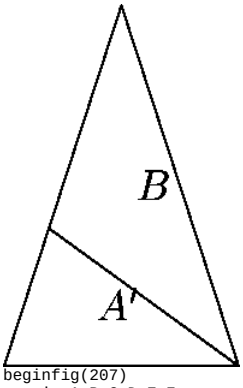



```
beginfig(205)
  pair A,B,C,D,E,F;
  numeric d[];
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  draw A--B--C--cycle;
  d[0] := 1;
  d[1] := sqrt( 2*(1+cosd(72)) );
  d[2] := sqrt( 2*(1-cosd(36)) );
  E := (d1/(d0+d1)) [A,C];
  F := (d0/(d0+d2)) [A,B];
  draw E--F--B--cycle;
  draw btex $A'$ etex shifted 1/3(E+F+B);
  draw E--A--F--cycle;
  draw btex $B$ etex shifted 1/3(E+A+F);
  draw C--E--B--cycle;
  draw btex $B'$ etex shifted 1/3(C+E+B);
endfig;
```

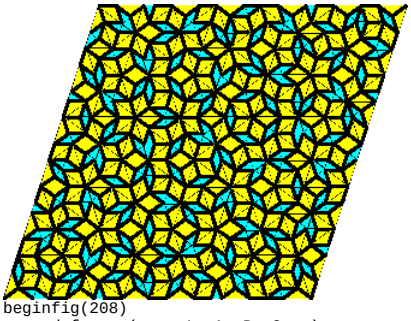



```
beginfig(206)
  pair A,B,C,D,E,F;
  numeric d[];
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  d[0] := 1;
  d[1] := sqrt( 2*(1+cosd(72)) );
  d[2] := sqrt( 2*(1-cosd(36)) );
  A := A;
  B := C;
  C := D;
       E := (d1/(d0+d1)) [A,C];
       F := (d0/(d0+d2)) [A,B];
  B := 3*(C-E);
  C := 3*(F-E);

  draw A--B--C--cycle;

  D := (d0/(d0+d2)) [C,A];
  draw B--C--D--cycle;
  draw btex $A$ etex shifted 1/3(B+C+D);
  draw B--D--A--cycle;
  draw btex $B'$ etex shifted 1/3(B+D+A);
```

```
endfig;
```

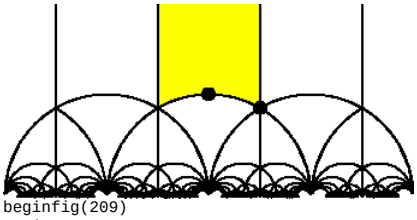



```
beginfig(207)
  pair A,B,C,D,E,F;
  numeric d[];
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  d[0] := 1;
  d[1] := sqrt( 2*(1+cosd(72)) );
  d[2] := sqrt( 2*(1-cosd(36)) );

  % B'
  E := (d1/(d0+d1)) [A,C];
  F := (d0/(d0+d2)) [A,B];

  % A'
  A := 2*(E-E);
  C := 2*(B-E);
  B := 2*(F-E);

  draw A--B--C--cycle;
  D := (d0/(d0+d2)) [A,B];
  draw C--D--B--cycle;
  draw btex $A'$ etex shifted 1/3(C+D+B);
  draw C--A--D--cycle;
  draw btex $B$ etex shifted 1/3(C+A+D);
endfig;
```



```
beginfig(208)
  vardef pave(expr t, A, B, C, n) =
    save D, E, d;
    pair D,E;
    numeric d[];
    d[0] := 1;
    d[1] := sqrt( 2*(1+cosd(72)) );
    d[2] := sqrt( 2*(1-cosd(36)) );
    if n>0:
      if t=1:
        D := (d0/(d0+d2)) [A,C];
        pave(1,B,C,D,n-1);
        pave(4,B,D,A,n-1);
      elseif t=2:
        D := (d0/(d0+d2)) [A,B];
        pave(2,C,D,B,n-1);
        pave(3,C,A,D,n-1);
      elseif t=3:
        D := (d1/(d0+d1)) [A,B];
        E := (d0/(d0+d2)) [A,C];
        pave(1,D,C,E,n-1);
        pave(3,B,C,D,n-1);
        pave(4,D,E,A,n-1);
      elseif t=4:
        D := (d1/(d0+d1)) [A,C];
        E := (d0/(d0+d2)) [A,B];
        pave(2,D,E,B,n-1);
        pave(3,D,A,E,n-1);
        pave(4,C,D,B,n-1);
      fi;
    else:
      draw A--B--C--cycle;
      if t=1:
        fill A--B--C--cycle withcolor green + blue;
        draw A--C withpen pencircle scaled 1bp;
        draw A--B withpen pencircle scaled 1bp;
      elseif t=2:
        fill A--B--C--cycle withcolor green + blue;
        draw A--B withpen pencircle scaled 1bp;
        draw A--B withpen pencircle scaled 1bp;
```

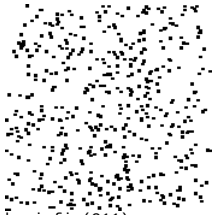
```
      elseif t=3:
        fill A--B--C--cycle withcolor green + red;
        draw A--C withpen pencircle scaled 1bp;
        draw C--B withpen pencircle scaled 1bp;
      elseif t=4:
        fill A--B--C--cycle withcolor green + red;
        draw B--C withpen pencircle scaled 1bp;
        draw A--B withpen pencircle scaled 1bp;
      fi;
    fi;
  enddef;
  numeric M;
  M := 6;
  pair A,B,C,D;
  u := 3cm;
  A := (0,0);
  B := (u,0);
  D := B rotated 72;
  C := (u,0) + D;
  pave(3,A,C,D,M);
  pave(4,A,B,C,M);
endfig;
```



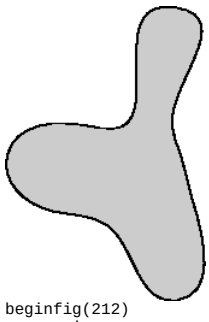
```
beginfig(209)
u:=1cm;

def milieu (expr s, ss, n) =
  if n<>0:
    for i="A","BA","BBA":
      milieu(s&i, ss, n-1);
    endfor;
  else:
    dessine(s&ss);
  fi;
enddef;

def doit(expr N) =
  for n=1 upto N:
    for i="A","BA","BBA":
      for j="","B","BB":
        milieu(i,j,n-1);
      endfor;
    endfor;
  endfor;
  milieu("","",0);
enddef;

vardef A(expr a) =
  save x,y,n;
  numeric x,y,n;
  x := xpart a;
  y := ypart a;
  n:=x*x+y*y;
  (-x/n, y/n)
enddef;

vardef B(expr a) =
  save x,y,n;
  numeric x,y,n;
  x := 1 + xpart a;
  y := ypart a;
  n := x*x+y*y;
  (-x/n, y/n)
enddef;

def dessine(expr s) =
  _dessine(s, dir(60),  up,       dir(120));
  _dessine(s, dir(60),  dir(31),  dir(2));
  _dessine(s, dir(120), dir(149), dir(178));
enddef;

def _dessine(expr s, a,b,c) =
  pair p[];
  p[0] := a;
  p[1] := b;
  p[2] := c;

  for i=0 upto length(s)-1:
    if (substring(i,i+1) of s) = "A":
        for j=0 upto 2: p[j] := A( p[j] ); endfor;
    elseif (substring(i,i+1) of s) = "B":
        for j=0 upto 2: p[j] := B( p[j] ); endfor;
    else: show("Ceci n'est pas un g^^e9n^^e9rateur : "&substring(i,i+1) of s)
    fi;
  endfor;
  if (xpart p[0] < 2) and (xpart p[0] > -2) and
     (xpart p[2] > -2) and (xpart p[2] < 2) and
     (ypart p[0] < 2) and (ypart p[2] < 2):
    draw ( p[0] .. p[1] .. p[2] ) scaled u;
  fi;

enddef;

  fill (dir(60) .. up .. dir(120) --
        (dir120+up) -- (dir60+up) -- cycle)  scaled u
       withcolor red+green;
  doit(7);
```

```
  draw (u*dir(60)) withpen pencircle scaled 4bp;
  draw (0,u) withpen pencircle scaled 4bp;
  draw (u*dir(60) -- u*dir(60)+u*up) shifted (0u,0);
  draw (u*dir(60) -- u*dir(60)+u*up) shifted (1u,0);
  draw (u*dir(60) -- u*dir(60)+u*up) shifted (-u,0);
  draw (u*dir(60) -- u*dir(60)+u*up) shifted (-2u,0);
endfig;
```

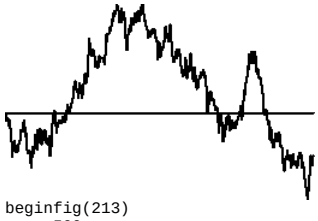
```
beginfig(210)
  u:=5mm;
  pickup pencircle scaled 1pt;
  for i=0 upto 500:
    draw (u*normaldeviate, u*normaldeviate);
  endfor;
endfig;
```



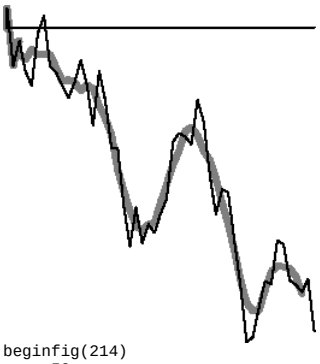
```
beginfig(211)
  u:=2cm;
  pickup pencircle scaled 1pt;
  for i=0 upto 500:
    draw (u*uniformdeviate(1), u*uniformdeviate(1));
  endfor;
endfig;
```



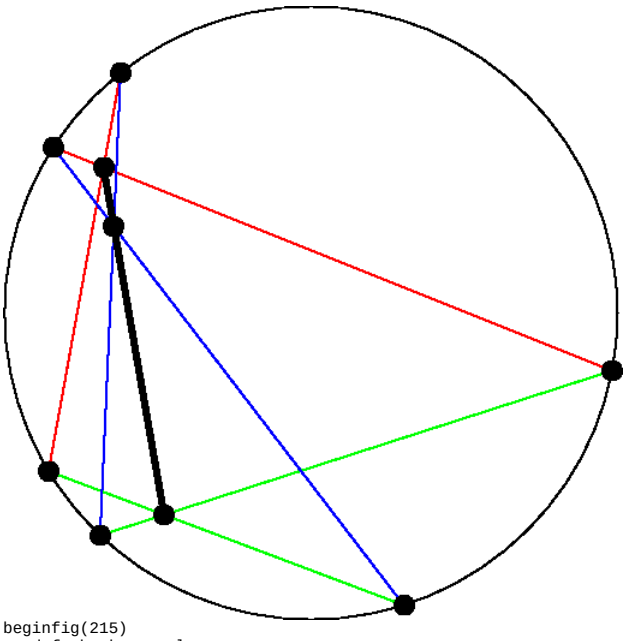
```
beginfig(212)
  numeric n;
  n:=10;
  path p;
  p := for i=0 upto n-1:
    ((1cm + 3mm*normaldeviate,0) rotated (i*360/n)) ..
  endfor cycle;
  fill p withcolor .8white;
  draw p;
endfig;
```



```
beginfig(213)
  m:=500;
  u:=1cm;
  pair A,B;
  A:=(0,0);
  for i=0 upto m:
    B:=(i/m*3u, (ypart A)+normaldeviate/m*30u);
    draw A--B;
    A:=B;
  endfor;
  draw (0,0)--(3u,0);
endfig;
```

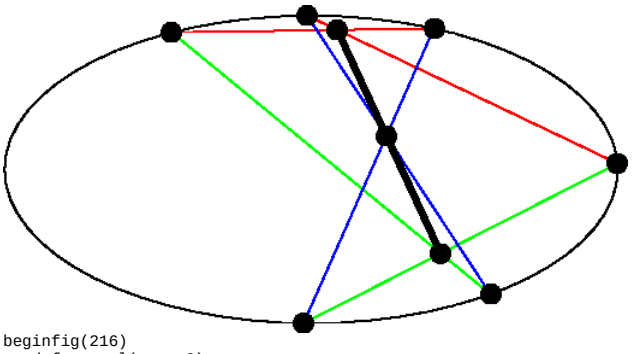


```
beginfig(214)
  m:=50;
  pair A[], B[];
  A[0]=(0,0);
  for i=0 upto m:
    A[i+1]=(i/m*3u, (ypart A[i])+(uniformdeviate(2)-1)/m*30u);
  endfor;
  B[0]=A[0];
  B[1]=A[1];
  B[2]=A[2];
  B[3]=A[3];
  for i=4 upto m+1:
    B[i] = ( A[i-4] + A[i-3] + A[i-2] + A[i-1] + A[i])/5;
  endfor;
  for i=0 upto m:
    draw B[i]--B[i+1] withpen pencircle scaled 2pt
         withcolor .5*white;
  endfor;
  for i=0 upto m:
    draw A[i]--A[i+1];
  endfor;
  draw (0,0)--(3u,0);
endfig;
```



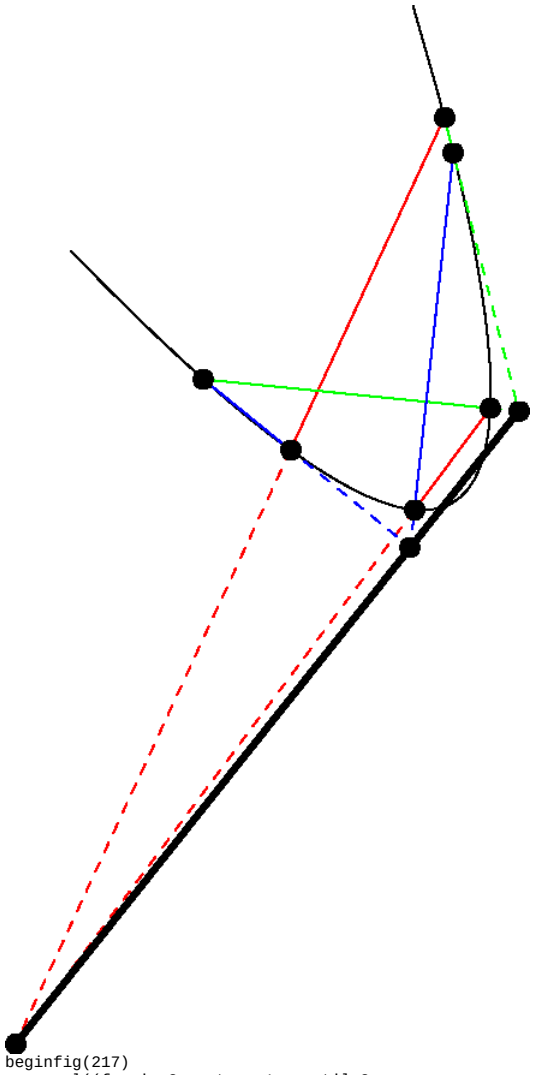
```
beginfig(215)
vardef check_pascal =
  save again;
  boolean again;
  again := false;
  % Distance entre deux points sur le cercle : au moins 1mm
  for i=0 upto 5:
    for j=i+1 upto 5:
      show( decimal(i) &" "& decimal(j) &" "& decimal(abs(A[i]-A[j])/2mm) );
      if abs(A[i]-A[j]) < 2mm:
        again := true;
      fi;
    endfor;
  endfor;
  % Distance entre deux des M[i] : au moins 2mm, au plus 10cm
  for i=0 upto 2:
    for j=i+1 upto 2:
      if (abs(M[i]-M[j]) > 10cm) or (abs(M[i]-M[j]) < 2mm):
        again := true;
      fi;
    endfor;
  endfor;
  % Distance entre un A[i] et un M[i] : au moins 2mm
  for i=0 upto 5:
    for j=0 upto 2:
      if abs(A[i]-M[j]) < 2mm:
        again := true;
      fi;
    endfor;
  endfor;
  % Distance entre le cercle et l'un des M[i] : au plus 10cm
```

```
  if abs(M[0])>10cm:
    again:=true;
  fi;
  show again;
  not again
enddef;

  forever:
    path C;
    C := fullcircle scaled 6cm;
    pair A[], M[];
    for i=0 upto 5:
      A[i] := point uniformdeviate(length(C)) of C;
    endfor;
    M[0] = whatever[ A[0], A[1] ];
    M[0] = whatever[ A[3], A[4] ];
    M[1] = whatever[ A[1], A[2] ];
    M[1] = whatever[ A[4], A[5] ];
    M[2] = whatever[ A[2], A[3] ];
    M[2] = whatever[ A[5], A[0] ];
    exitif check_pascal;
  endfor;

  draw C;
  draw A[0]--A[1] withcolor red;
  draw A[3]--A[4] withcolor red;
  draw A[0]--M[0] withcolor red dashed evenly;
  draw A[3]--M[0] withcolor red dashed evenly;

  draw A[1]--A[2] withcolor green;
  draw A[4]--A[5] withcolor green;
  draw A[1]--M[1] withcolor green dashed evenly;
  draw A[4]--M[1] withcolor green dashed evenly;

  draw A[2]--A[3] withcolor blue;
  draw A[5]--A[0] withcolor blue;
  draw A[2]--M[2] withcolor blue dashed evenly;
  draw A[5]--M[2] withcolor blue dashed evenly;

  draw M[0]--M[1]--M[2]--cycle withpen pencircle scaled 2bp;
  for i=0 upto 2:
    draw M[i] withpen pencircle scaled 6bp;
  endfor;
  for i=0 upto 5:
    draw A[i] withpen pencircle scaled 6bp;
  endfor;

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```

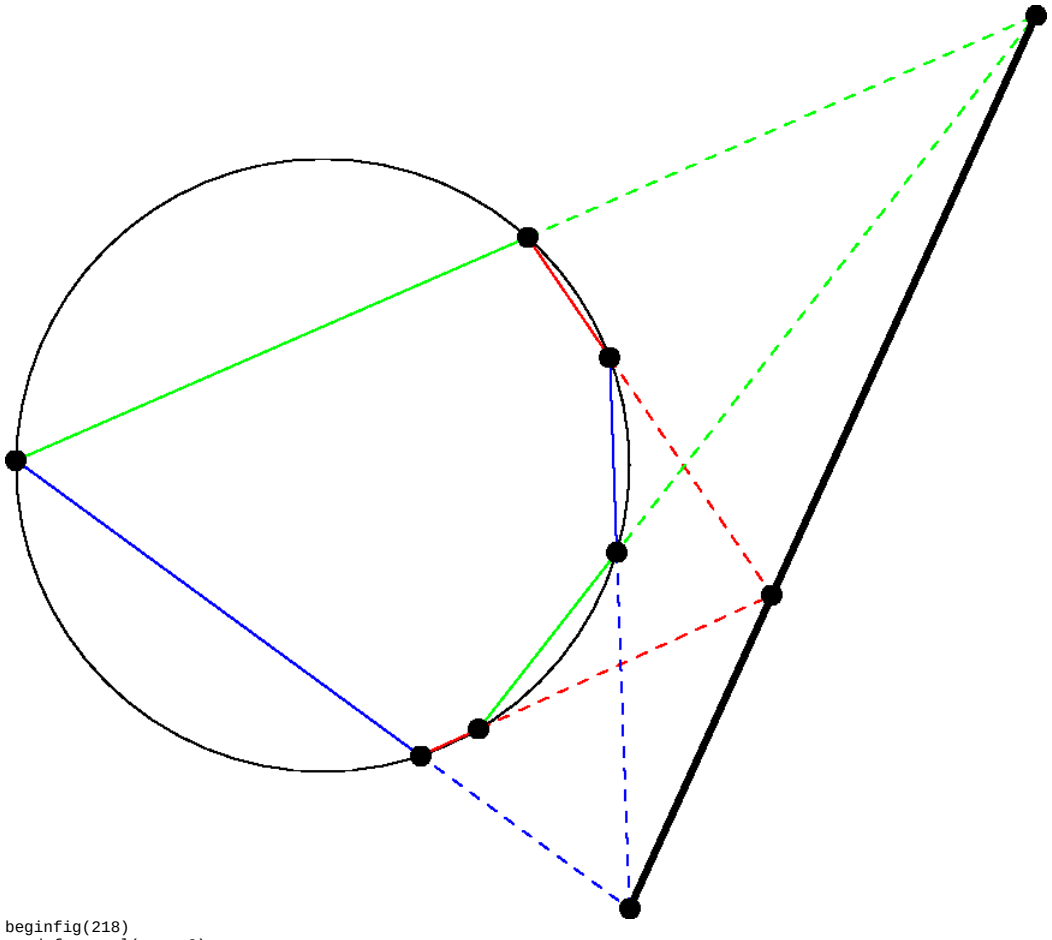


```
beginfig(216)
vardef pascal(expr C) =
  forever:
    pair A[], M[];
    for i=0 upto 5:
      A[i] := point uniformdeviate(length(C)) of C;
    endfor;
    M[0] = whatever[ A[0], A[1] ];
    M[0] = whatever[ A[3], A[4] ];
    M[1] = whatever[ A[1], A[2] ];
    M[1] = whatever[ A[4], A[5] ];
    M[2] = whatever[ A[2], A[3] ];
    M[2] = whatever[ A[5], A[0] ];
    exitif check_pascal;
  endfor;

  draw C;
  draw A[0]--A[1] withcolor red;
  draw A[3]--A[4] withcolor red;
  draw A[0]--M[0] withcolor red dashed evenly;
  draw A[3]--M[0] withcolor red dashed evenly;

  draw A[1]--A[2] withcolor green;
  draw A[4]--A[5] withcolor green;
  draw A[1]--M[1] withcolor green dashed evenly;
  draw A[4]--M[1] withcolor green dashed evenly;

  draw A[2]--A[3] withcolor blue;
  draw A[5]--A[0] withcolor blue;
  draw A[2]--M[2] withcolor blue dashed evenly;
  draw A[5]--M[2] withcolor blue dashed evenly;

  draw M[0]--M[1]--M[2]--cycle withpen pencircle scaled 2bp;
  for i=0 upto 2:
    draw M[i] withpen pencircle scaled 6bp;
  endfor;
```

```
  for i=0 upto 5:
    draw A[i] withpen pencircle scaled 6bp;
  endfor;
enddef;

  pascal(fullcircle xscaled 6cm yscaled 3cm);

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```

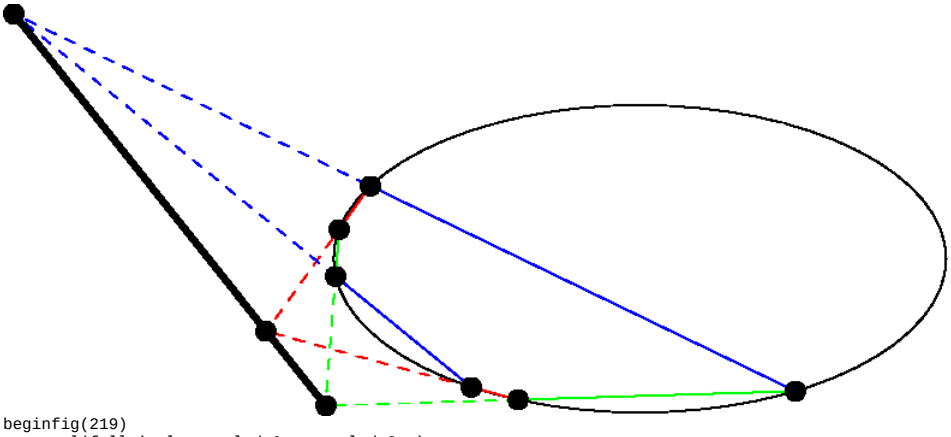


```
beginfig(217)
  pascal((for i=-2cm step .1cm until 2cm:
        (i, (i/1cm)**2 *1cm) ..
      endfor (2.1cm,4.41cm)) rotated 30);

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```

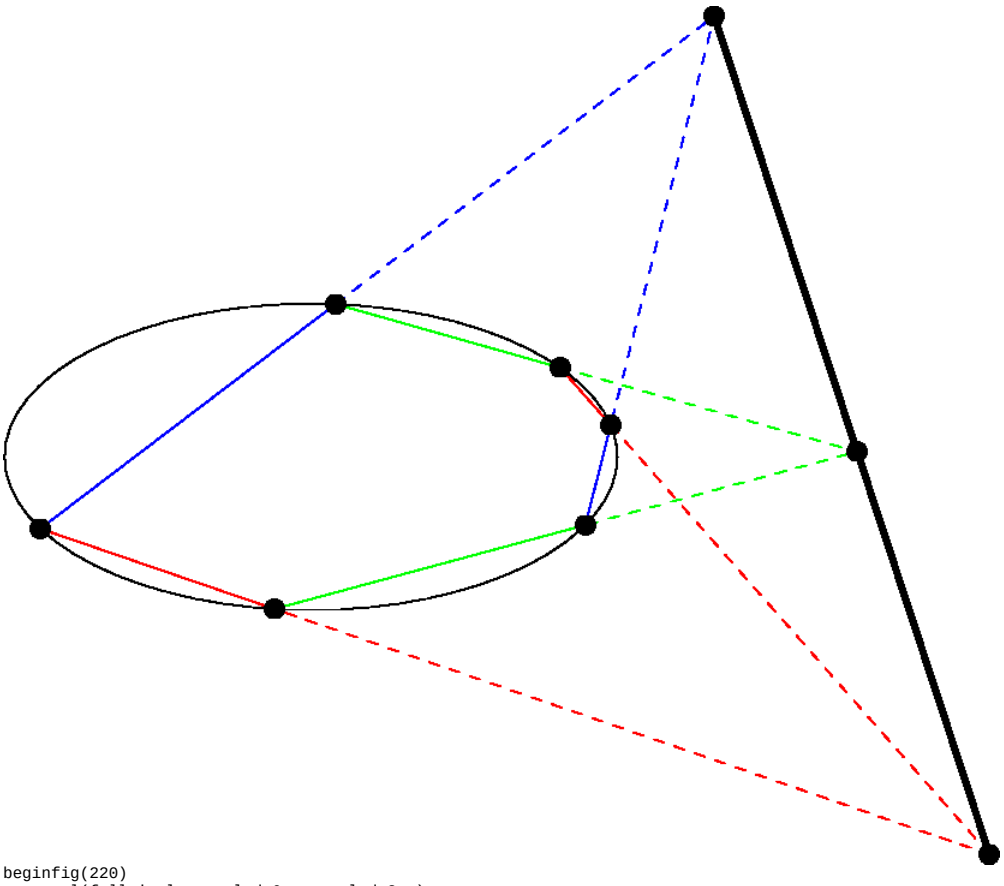


```
beginfig(218)
vardef pascal(expr C) =
  forever:
    pair A[], M[];

    numeric t[];
    for i=0 upto 5:
      t[i] = uniformdeviate(length(C));
    endfor;
    for i=0 upto 5:
      for j=0 upto 4:
        if t[j]>t[j+1]:
          tt:=t[j]; t[j]:=t[j+1]; t[j+1]:=tt;
        fi;
      endfor;
    endfor;
    for i=0 upto 5:
      A[i] = point t[i] of C;
    endfor;

    M[0] = whatever[ A[0], A[1] ];
    M[0] = whatever[ A[3], A[4] ];
    M[1] = whatever[ A[1], A[2] ];
    M[1] = whatever[ A[4], A[5] ];
    M[2] = whatever[ A[2], A[3] ];
    M[2] = whatever[ A[5], A[0] ];
    exitif check_pascal;
  endfor;

  draw C;
  draw A[0]--A[1] withcolor red;
  draw A[3]--A[4] withcolor red;
  draw A[0]--M[0] withcolor red dashed evenly;
  draw A[3]--M[0] withcolor red dashed evenly;

  draw A[1]--A[2] withcolor green;
  draw A[4]--A[5] withcolor green;
  draw A[1]--M[1] withcolor green dashed evenly;
  draw A[4]--M[1] withcolor green dashed evenly;

  draw A[2]--A[3] withcolor blue;
  draw A[5]--A[0] withcolor blue;
  draw A[2]--M[2] withcolor blue dashed evenly;
  draw A[5]--M[2] withcolor blue dashed evenly;

  draw M[0]--M[1]--M[2]--cycle withpen pencircle scaled 2bp;
  for i=0 upto 2:
    draw M[i] withpen pencircle scaled 6bp;
  endfor;
  for i=0 upto 5:
    draw A[i] withpen pencircle scaled 6bp;
  endfor;
enddef;

  pascal(fullcircle scaled 6cm);

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
```

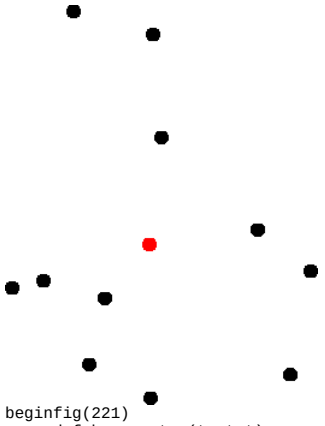
```
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```



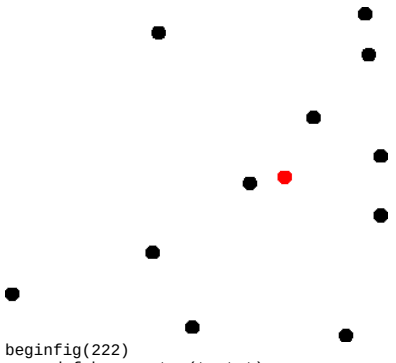
```
beginfig(219)
  pascal(fullcircle xscaled 6cm yscaled 3cm);

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```



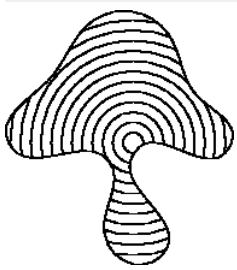
```
beginfig(220)
  pascal(fullcircle xscaled 6cm yscaled 3cm);

  if xpart (lrcorner currentpicture - llcorner currentpicture) > 4cm:
    currentpicture := currentpicture scaled 4cm /
      xpart (lrcorner currentpicture - llcorner currentpicture);
  fi;
endfig;
```



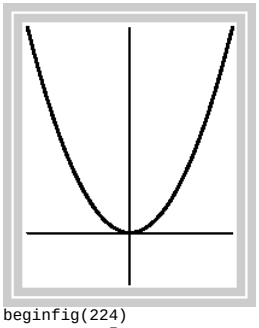
```
beginfig(221)
  vardef barycentre(text t) =
    save n, G;
    pair G; numeric n;
    G := origin; n:=0;
    for a=t:
      G:=G+a;
      n:=n+1;
    endfor;
    G/n
  enddef;

  pair A[];
  n:=10;
  for i=0 upto n:
    A[i] = 1cm*(normaldeviate, normaldeviate);
    draw A[i] withpen pencircle scaled 4bp;
  endfor;
  draw barycentre(A[0] for i=1 upto 10: ,A[i] endfor)
    withpen pencircle scaled 4bp withcolor red;
endfig;
```



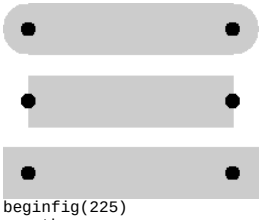
```
beginfig(222)
  vardef barycentre(text t) =
    save a, i, n, G, X;
    pair G,X; numeric n,i;
    G := origin; n:=0; i:=0;
    for a=t:
      show("i = "& decimal(i));
      show a;
      if odd(i):
        show("odd");
        n:=n+a;
        G:= G + a*X;
      else:
        show("even");
        X:=a;
      fi;
      i:=i+1;
    endfor;
    G/n
  enddef;

  pair A[];
  n:=10;
  for i=0 upto n:
    A[i] = 1cm*(normaldeviate, normaldeviate);
    draw A[i] withpen pencircle scaled 4bp;
  endfor;
  draw barycentre(A[0],0 for i=1 upto 10: ,A[i],i endfor)
    withpen pencircle scaled 4bp withcolor red;
endfig;
```



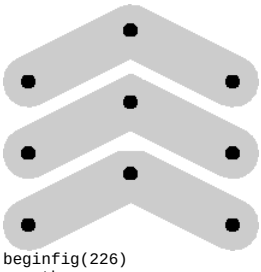
```
beginfig(223)
    numeric n;
    n:=10;
    path p;
    p := for i=0 upto n-1:
        ((1cm + 3mm*normaldeviate,0) rotated (i*360/n)) ..
      endfor cycle;
    for i=0 step 2mm until 3cm:
      draw fullcircle scaled i;
    endfor;
    clip currentpicture to p;
    draw p;

endfig;
```



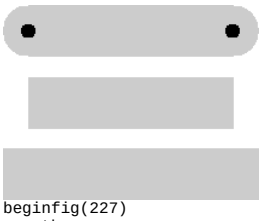
```
beginfig(224)
      u := 5mm;
  draw (-2u,0) -- (2u,0);
  draw (0,-u) -- (0,4u);
  draw (-2u,4u) for i=-1.9 step .1 until 2.01: .. (i*u, i*i*u) endfor
    withpen pencircle scaled 1bp;
  draw bbox currentpicture withpen pensquare scaled 2bp withcolor .8white;
  draw bbox currentpicture withpen pensquare scaled 2bp withcolor .8white;
endfig;
```



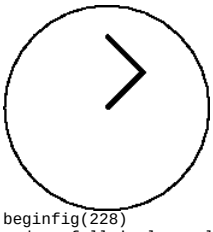
```
beginfig(225)
  path p;
  p := (0,0) -- (2cm,0);
  def doit (suffix p)(expr t) =
    begingroup
      interim linecap := t;
      draw p withpen pencircle scaled 5mm withcolor .8white;
    endgroup;
    draw point 0 of p withpen pencircle scaled 4bp;
    draw point 1 of p withpen pencircle scaled 4bp;
    p := p shifted (0,-7mm)
  enddef;
  doit(p,rounded);
  doit(p,butt);
  doit(p,squared);

endfig;
```



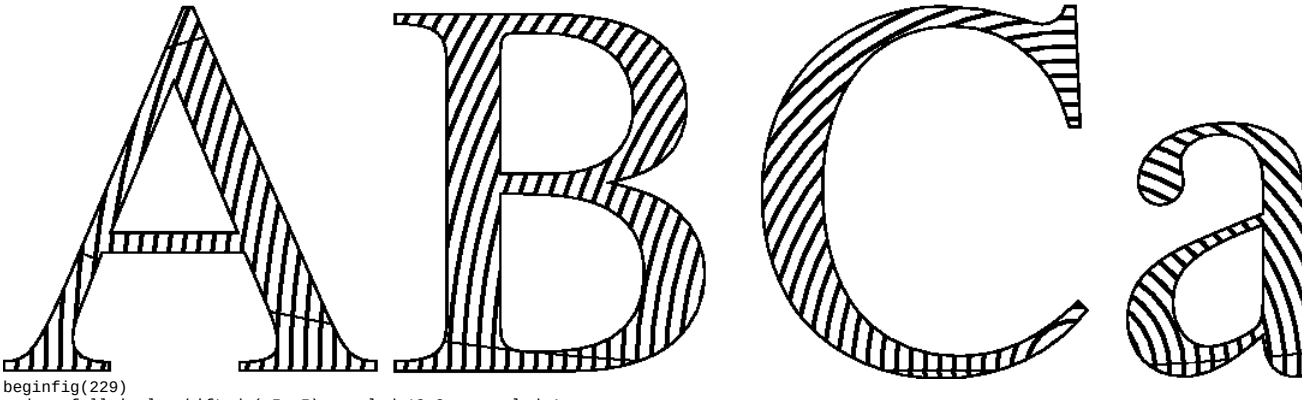
```
beginfig(226)
  path p;
  p := (0,0) -- (1cm,5mm) -- (2cm,0);
  def doit (suffix p)(expr t) =
    begingroup
      interim linejoin := t;
      draw p withpen pencircle scaled 5mm withcolor .8white;
    endgroup;
    draw point 0 of p withpen pencircle scaled 4bp;
    draw point 1 of p withpen pencircle scaled 4bp;
    draw point 2 of p withpen pencircle scaled 4bp;
    p := p shifted (0,-7mm)
  enddef;
  doit(p,rounded);
  doit(p,mitered);
  doit(p,beveled);

endfig;
```



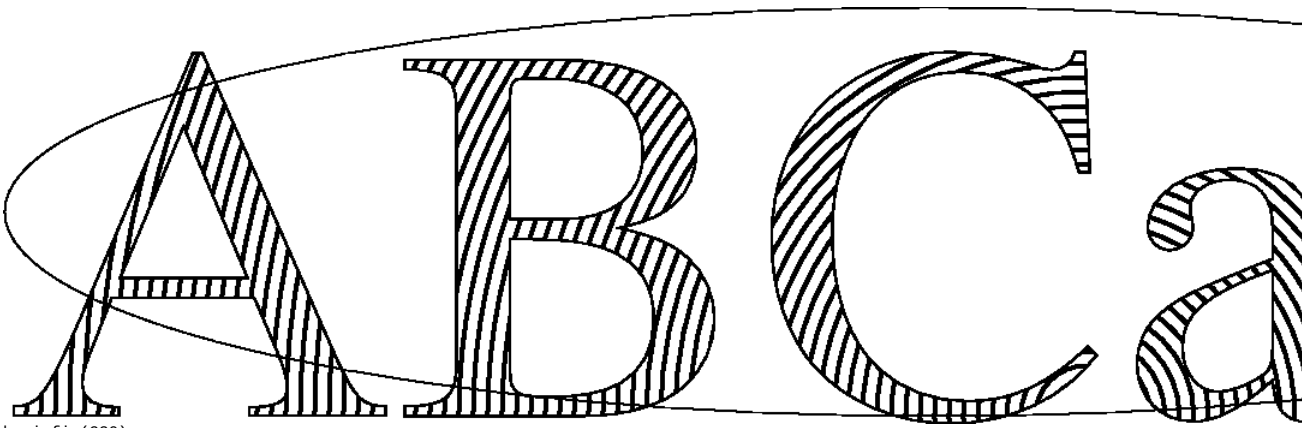
```
beginfig(227)
  path p;
  p := (0,0) -- (2cm,0);
  vardef doit (suffix p)(expr t) =
    interim linecap := t;
    draw p withpen pencircle scaled 5mm withcolor .8white;
    draw point 0 of p withpen pencircle scaled 4bp;
    draw point 1 of p withpen pencircle scaled 4bp;
    draw point 2 of p withpen pencircle scaled 4bp;
    p := p shifted (0,-7mm)
  enddef;
  doit(p,rounded);
  doit(p,butt);
  doit(p,squared);

endfig;
```



```
beginfig(228)
  draw fullcircle scaled 2cm;
  special("0 0 moveto 10 10 rlineto -10 10 rlineto stroke");

endfig;
```



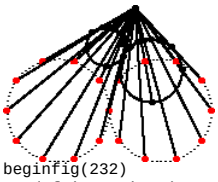
```
beginfig(229)
  draw fullcircle shifted (.5,.5) xscaled 18.2cm yscaled 4cm;
  special(
    "/Times-Roman findfont 150 scalefont setfont " &
    "0 0 moveto (ABCabc) false charpath clip stroke " &
    "gsave 300 0 translate " &
    "  2 4 600 {dup 0 moveto 0 exch 0 exch 0 360 arc stroke} for " &
    "grestore "
    );

endfig;
```



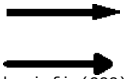
```
beginfig(230)
  draw fullcircle shifted (.5,.5) xscaled 18.2cm yscaled 4cm;
  special(
    "gsave " &
    "/Times-Roman findfont 150 scalefont setfont " &
    "0 0 moveto (ABCabc) false charpath clip stroke " &
```

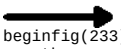
```
    "gsave 300 0 translate " &
    "  2 4 600 {dup 0 moveto 0 exch 0 exch 0 360 arc stroke} for " &
    "grestore " &
    "grestore "
    );

endfig;
```

```
beginfig(231)
  path p;
  p =
  (0,u)
  for i=.1 step .1 until 10:
    hide( pair A; A = (i*u, (sind (i*180/3.14))/i *u);
          draw A withpen pencircle scaled 2pt )
    .. A
  endfor;
  draw p;
endfig;
```

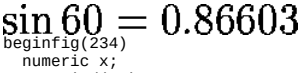


```
beginfig(232)
vardef inversion (expr O,k,M) =
  if pair M:
    (O + k*unitvector(M-O)/abs(M-O))
  elseif path M:
    for i=0 step length(M)/10 until length(M):
      hide(
        draw O--inversion(O,k,point i of M)--(point i of M);
        draw inversion(O,k,point i of M) withpen pencircle scaled 2pt;
        draw point i of M withpen pencircle scaled 2pt withcolor red;
        draw O withpen pencircle scaled 2pt;
        )
      inversion(O,k,point i of M) ..
    endfor
    cycle
  fi
enddef;

  path p[];
  p1=fullcircle scaled 2u shifted (u,0);
  p2=fullcircle scaled 2u shifted (-u,0);
  draw p1 dashed withdots scaled .25;
  draw p2 dashed withdots scaled .25;
  z0 = (.5u,2u);
  draw inversion( z0, 2 (u**2), p1 );
  draw inversion( z0, 2 (u**2), p2 );
endfig;
```

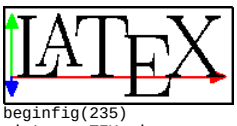




```
beginfig(233)
  path p;
  p := (0,0) -- (1cm,0);
  show ahangle;
  begingroup
    interim ahangle := 30;
    interim linejoin := mitered;
    interim linecap := butt;
    drawarrow p withpen pencircle scaled 2bp;
  endgroup;
  begingroup
    interim ahangle := 60;
    drawarrow p shifted (0,-5mm) withpen pencircle scaled 2bp;
  endgroup;

endfig;
```

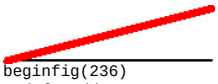


```
beginfig(234)
  numeric x;
  x = sind(60);
  draw TEX("$\sin 60 = " & decimal(x) & "$");

endfig;
```



```
beginfig(235)
picture _TEX_pic;

def largeur(expr p) =
  if picture p:
    xpart( lrcorner(p) - llcorner(p) )
  elseif string p:
    hide(_TEX_pic := TEX(p);)
    xpart( lrcorner(_TEX_pic) - llcorner(_TEX_pic) )
  else:
    hide( errmessage("largeur: wrong type"); 0)
  fi
```

```
enddef;

def hauteur(expr p) =
  if picture p:
    ypart(ulcorner(p))
  elseif string p:
    hide(_TEX_pic := TEX(p);)
    ypart(ulcorner(_TEX_pic))
  else:
    hide( errmessage("hauteur: wrong type"); 0)
  fi
enddef;

def profondeur (expr p) =
  if picture p:
    -ypart(llcorner(p))
  elseif string p:
    hide(_TEX_pic := TEX(p);)
    -ypart(llcorner(_TEX_pic))
  else:
    hide( errmessage("profondeur: wrong type"); 0)
  fi
enddef;

  picture p;
  string s;
  s := "\huge\LaTeX";
  p := TEX(s);
  drawarrow (0,0) -- (largeur(s),0) withcolor red;
  drawarrow (0,0) -- (0,hauteur(s)) withcolor green;
  drawarrow (0,0) -- (0,-profondeur(s)) withcolor blue;
  draw p;
  draw bbox p;
endfig;
```



```
beginfig(236)
  def reddraw text t = draw t withcolor red enddef;
  draw (0,0) -- (2cm,0);
  reddraw (0,0) -- (2cm,5mm) withpen pencircle scaled 2bp;

endfig;
```

```
beginfig(237)
path _myfill_p;

def myfill expr p =
  _myfill_p := p;
  _myfill
enddef;

def _myfill text t =
  fill _myfill_p t;
  draw _myfill_p t withcolor red;
enddef;

  myfill fullcircle scaled 1cm withcolor .8white withpen pencircle scaled 2bp;
endfig;
```


```
beginfig(238)
    boxit.a(btex Essai etex);
    a.c = (0,0);
    drawboxed(a);

endfig;
```


```
beginfig(239)
    circleit.a(btex Essai etex);
    a.c = (0,0);
    drawboxed(a);

endfig;
```

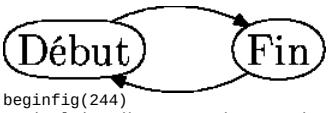

```
beginfig(240)
  rboxit.a(btex Essai etex);
  a.c = (0,0);
  drawboxed(a);

endfig;
```


```
beginfig(241)
  boxit.a(btex Essai etex);
  a.c = (0,0);
  drawunboxed(a);
```

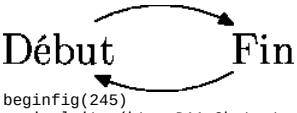
```
endfig;
```


```
beginfig(242)
  circleit.a(btex Essai etex);
  a.c = (0,0);
  a.dx = a.dy;
  drawboxed(a);
endfig;
```

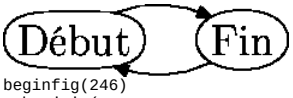

```
beginfig(243)
  circleit.a(btex Essai etex);
  a.c = (0,0);
  a.dx = a.dy;
  drawunboxed(a);
  for i=0 step 10 until 360:
    draw (0,0) -- 1cm*right rotated i cutbefore bpath.a;
  endfor;
endfig;
```



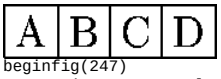

```
beginfig(244)
  circleit.a(btex D^^e9but etex);
  a.c = (0,0);
  a.dx = a.dy;
  circleit.b(btex Fin etex);
  b.c = (2cm,0);
  b.dx = b.dy;
  drawboxed(a,b);
  drawarrow  a.c {dir 45} .. b.c {dir -45}
  cutbefore bpath.a cutafter bpath.b;
  drawarrow  b.c {dir -135} .. a.c {dir 135}
  cutbefore bpath.b cutafter bpath.a;
endfig;
```



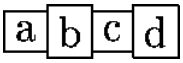

```
beginfig(245)
  circleit.a(btex D^^e9but etex);
  a.c = (0,0);
  a.dx = a.dy;
  circleit.b(btex Fin etex);
  b.c = (2cm,0);
  b.dx = b.dy;
  drawunboxed(a,b);
  drawarrow  a.c {dir 45} .. b.c {dir -45}
  cutbefore bpath.a cutafter bpath.b;
  drawarrow  b.c {dir -135} .. a.c {dir 135}
  cutbefore bpath.b cutafter bpath.a;
endfig;
```



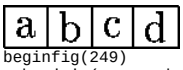

```
beginfig(246)
  boxjoin(
    a.dx = a.dy;
    b.dx = b.dy;
    a.e + (5mm,0) = b.w;
    );
  circleit.a(btex D^^e9but etex);
  a.c = (0,0);
  circleit.b(btex Fin etex);
  drawboxed(a,b);
  drawarrow  a.c {dir 45} .. b.c {dir -45}
  cutbefore bpath.a cutafter bpath.b;
  drawarrow  b.c {dir -135} .. a.c {dir 135}
  cutbefore bpath.b cutafter bpath.a;
endfig;
```



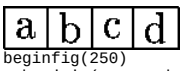

```
beginfig(247)
  % Toujours mettre la commande boxjoin au d^^e9but.
  boxjoin(a.e = b.w);
  boxit.a(btex A etex);
  boxit.b(btex B etex);
  boxit.c(btex C etex);
  boxit.d(btex D etex);
  drawboxed(a,b,c,d);
endfig;
```



```
beginfig(248)
  boxjoin(a.e = b.w);
  boxit.a(btex a etex);
  boxit.b(btex b etex);
  boxit.c(btex c etex);
  boxit.d(btex d etex);
  drawboxed(a,b,c,d);
endfig;
```

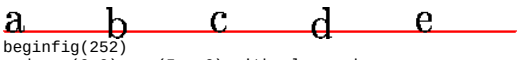


```
beginfig(249)
  boxjoin(a.se = b.sw; a.ne = b.nw);
  boxit.a(btex a etex);
  boxit.b(btex b etex);
  boxit.c(btex c etex);
  boxit.d(btex d etex);
  drawboxed(a,b,c,d);
endfig;
```



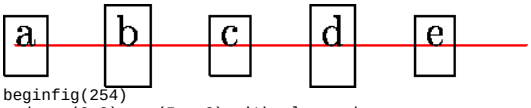
```
beginfig(250)
  boxjoin(a.se = b.sw; a.ne = b.nw);
  boxit.a1(btex a etex);
  boxit.a2(btex b etex);
  boxit.a3(btex c etex);
  boxit.a4(btex d etex);
  drawboxed(a1,a2,a3,a4);
endfig;
```

```
beginfig(251)
  draw (0,0) -- (5cm,0) withcolor red;
  draw btex a etex ;
  draw btex b etex shifted (1cm,0);
  draw btex c etex shifted (2cm,0);
  draw btex d etex shifted (3cm,0);
  draw btex e etex shifted (4cm,0);
endfig;
```



```
beginfig(252)
  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawunboxed(a,b,c,d,e);
endfig;
```

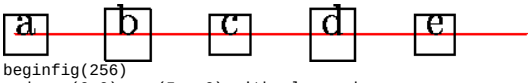
```
beginfig(253)
  vardef boxTEX expr s =
    save p,h,l;
    picture p;
    numeric h,l;
    p := TEX(s);
    h := max(ypart ulcorner(p), abs(ypart llcorner(p)));
    l := xpart lrcorner(p);
    setbounds p to (0,-h)--(l,-h)--(l,h)--(0,h)--cycle;
    p
  enddef;

  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (boxTEX "a");
  boxit b (boxTEX "b");
  boxit c (boxTEX "c");
  boxit d (boxTEX "d");
  boxit e (boxTEX "e");
  drawunboxed(a,b,c,d,e);
endfig;
```



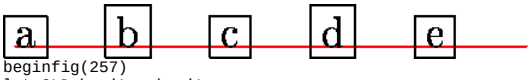
```
beginfig(254)
  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (boxTEX "a");
  boxit b (boxTEX "b");
  boxit c (boxTEX "c");
  boxit d (boxTEX "d");
  boxit e (boxTEX "e");
  drawboxed(a,b,c,d,e);
endfig;
```

```
beginfig(255)
  let OLD_beginbox_ = beginbox_;
  def beginbox_(expr pp,sp)(suffix $)(text t) =
    _n_ := str $;
    generic_declare(pair) _n.off, _n.c;
    generic_declare(string) pproc_._n, sproc_._n;
    generic_declare(picture) pic_._n;
    pproc_$:=pp; sproc_$:=sp;
    pic_$ = nullpicture;
    for _p_=t:
      pic_$:=
```

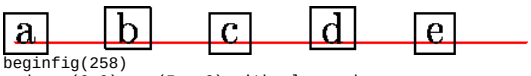
```
      if picture _p_: _p_
      else: _p_ infont defaultfont scaled defaultscale
      fi;
    endfor
    $c = $off + (.5[xpart llcorner pic_$, xpart urcorner pic_$], 0)
  enddef;

  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawunboxed(a,b,c,d,e);
endfig;
```



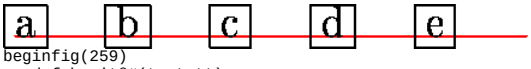
```
beginfig(256)
  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```



```
beginfig(257)
let OLD_boxit = boxit;
let OLD_sizebox_ = sizebox_;
let OLD_clearb_ = clearb_;

vardef boxit@#(text tt) =
  beginbox_("boxpath_","sizebox_",@#,tt);
  generic_declare(pair) _n.sw, _n.s, _n.se, _n.e, _n.ne, _n.n, _n.nw, _n.w;
  0 = xpart (@#nw-@#sw) = ypart(@#se-@#sw);
  0 = xpart(@#ne-@#se) = ypart(@#ne-@#nw);
  @#s = .5[@#sw,@#se];
  @#n = .5[@#ne,@#nw];
  xpart @#w = xpart @#nw;
  xpart @#e = xpart @#ne;
  ypart @#w = ypart @#c = ypart @#e;
  @#ne-@#c = (@#dx,@#dy) +
             (xpart(.5*(urcorner pic_@# - llcorner pic_@#)), ypart urcorner pic_@#);
  @#c-@#sw = (@#dx,@#dy) +
             (xpart(.5*(urcorner pic_@# - llcorner pic_@#)), -ypart lrcorner pic_@#);
  endbox_(clearb_,@#);
enddef;

  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```

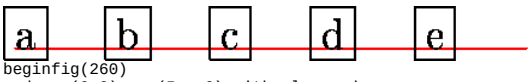


```
beginfig(258)
  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0); b.ne - b.se = a.nw - a.sw);
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```

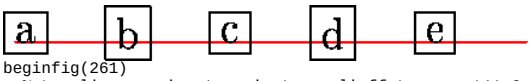


```
beginfig(259)
vardef boxit@#(text tt) =
  beginbox_("boxpath_","sizebox_",@#,tt);
  generic_declare(pair) _n.sw, _n.s, _n.se, _n.e, _n.ne, _n.n, _n.nw, _n.w;
  0 = xpart (@#nw-@#sw) = ypart(@#se-@#sw);
  0 = xpart(@#ne-@#se) = ypart(@#ne-@#nw);
  @#s = .5[@#sw,@#se];
  @#n = .5[@#ne,@#nw];
  xpart @#w = xpart @#nw;
  xpart @#e = xpart @#ne;
  ypart @#w = ypart @#c = ypart @#e;
  @#ne-@#c = (@#dx,@#dyup) +
             (xpart(.5*(urcorner pic_@# - llcorner pic_@#)), ypart urcorner pic_@#);
  @#c-@#sw = (@#dx,@#dydown) +
             (xpart(.5*(urcorner pic_@# - llcorner pic_@#)), -ypart lrcorner pic_@#);
  endbox_(clearb_,@#);
enddef;

def sizebox_(suffix $) =
  if unknown $.dx: $.dx=defaultdx; fi
  if unknown $.dyup:
    if unknown $.dy:
      $.dyup=defaultdy
    else:
      $.dyup=$.dy
```

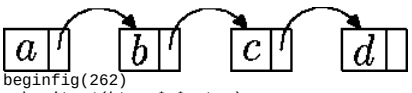
```
    fi;
  fi;
  if unknown $.dydown:
    if unknown $.dy:
      $.dydown=defaultdy
    else:
      $.dydown=$.dy
    fi;
  fi;
enddef;

vardef clearb_(suffix $) =
  _n_ := str $;
  generic_redeclare(numeric) _n.sw, _n.s, _n.se, _n.e, _n.ne, _n.n, _n.nw, _n.w,
    _n.c, _n.off, _n.dx, _n.dy, _n.dyup, _n.dy_down;
enddef;

  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0); ypart( b.ne - a.ne ) = 0; ypart( b.se - a.se ) = 0;);
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```



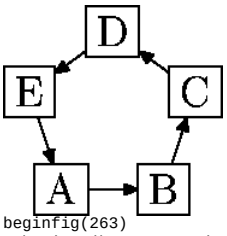

```
beginfig(260)
  draw (0,0) -- (5cm,0) withcolor red;
  picture p;
  numeric h;
  p := btex b etex;
  h = ypart ( ulcorner p - llcorner p );
  boxjoin(
    b.c - a.c = (1cm,0);
    ypart( b.ne - a.ne ) = 0;
    ypart( a.ne - a.se ) = h+2defaultdy;
  );
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```



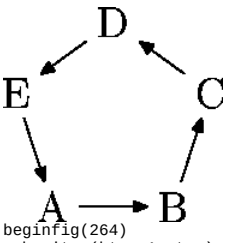

```
beginfig(261)
  % Les lignes suivantes n'ont pas l'effet escompt^^e9...
  let beginbox_ = OLD_beginbox_;
  let boxit     = OLD_boxit;
  let sizebox_  = OLD_sizebox_;
  let clearb_   = OLD_clearb_;

  % On recopie donc les d^^e9finitions initiales des macros que l'on a modifi^^e9es...
  extra_beginfig := "";
  extra_endfig   := "";
  input boxes;

  % Et on regarde sur un exemple si ^^e7a marche...
  draw (0,0) -- (5cm,0) withcolor red;
  boxjoin(b.c - a.c = (1cm,0));
  boxit a (btex a etex);
  boxit b (btex b etex);
  boxit c (btex c etex);
  boxit d (btex d etex);
  boxit e (btex e etex);
  drawboxed(a,b,c,d,e);
endfig;
```



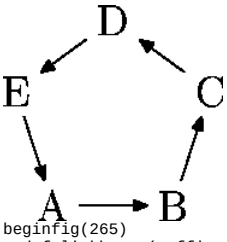

```
beginfig(262)
  boxit.a1(btex $a$ etex);
  boxit.b1(btex etex);
  boxit.a2(btex $b$ etex);
  boxit.b2(btex etex);
  boxit.a3(btex $c$ etex);
  boxit.b3(btex etex);
  boxit.a4(btex $d$ etex);
  boxit.b4(btex etex);
  for i=1 upto 4:
    % Les boites a[i] et b[i] sont coll^^e9es
    a[i].e = b[i].w;
    % Il y a un peu d'espace entre b[i] et a[i+1]
    b[i].e + (5mm,0) = a[i+1].w;
    % La hauteur des boites est la m^^eame
    a[i].n - a[i].s = a[i+1].n - a[i+1].s = b[i].n - b[i].s ;
    % Les b[i] ne sont pas tr^^e8s larges
    b[i].e - b[i].w = (2mm,0);
  endfor;
  drawboxed(a1,b1,a2,b2,a3,b3,a4,b4);
  for i=1 upto 3:
    drawarrow b[i].c {up} .. a[i+1].c {down}
    cutafter bpath.a[i+1];
  endfor;
endfig;
```



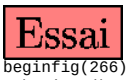

```
beginfig(263)
  boxit.a(btex A etex);
  boxit.b(btex B etex);
  boxit.c(btex C etex);
  boxit.d(btex D etex);
  boxit.e(btex E etex);

  b.c-a.c = (1cm,0);
  c.c-b.c = (b.c-a.c) rotated 72;
  d.c-c.c = (c.c-b.c) rotated 72;
  e.c-d.c = (d.c-c.c) rotated 72;
  a.c-e.c = (e.c-d.c) rotated 72;
  b.c-a.c = (a.c-e.c) rotated 72;

  drawboxed(a,b,c,d,e);

  drawarrow a.c -- b.c cutbefore bpath.a cutafter bpath.b;
  drawarrow b.c -- c.c cutbefore bpath.b cutafter bpath.c;
  drawarrow c.c -- d.c cutbefore bpath.c cutafter bpath.d;
  drawarrow d.c -- e.c cutbefore bpath.d cutafter bpath.e;
  drawarrow e.c -- a.c cutbefore bpath.e cutafter bpath.a;
endfig;
```



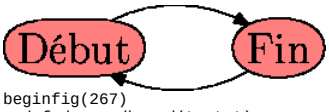

```
beginfig(264)
  boxit.a(btex A etex);
  boxit.b(btex B etex);
  boxit.c(btex C etex);
  boxit.d(btex D etex);
  boxit.e(btex E etex);

  d.c = 1cm*up;
  e.c = 1cm*up rotated (1*72);
  a.c = 1cm*up rotated (2*72);
  b.c = 1cm*up rotated (3*72);
  c.c = 1cm*up rotated (4*72);

  drawunboxed(a,b,c,d,e);

  drawarrow a.c -- b.c cutbefore bpath.a cutafter bpath.b;
  drawarrow b.c -- c.c cutbefore bpath.b cutafter bpath.c;
  drawarrow c.c -- d.c cutbefore bpath.c cutafter bpath.d;
  drawarrow d.c -- e.c cutbefore bpath.d cutafter bpath.e;
  drawarrow e.c -- a.c cutbefore bpath.e cutafter bpath.a;
endfig;
```



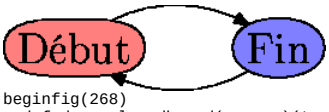

```
beginfig(265)
  def linkboxes(suffix a,b) =
    drawarrow a.c -- b.c cutbefore bpath.a cutafter bpath.b;
  enddef;

  boxit.a(btex A etex);
  boxit.b(btex B etex);
  boxit.c(btex C etex);
  boxit.d(btex D etex);
  boxit.e(btex E etex);

  d.c = 1cm*up;
  e.c = 1cm*up rotated (1*72);
  a.c = 1cm*up rotated (2*72);
  b.c = 1cm*up rotated (3*72);
  c.c = 1cm*up rotated (4*72);

  drawunboxed(a,b,c,d,e);

  linkboxes(a,b);
  linkboxes(b,c);
  linkboxes(c,d);
  linkboxes(d,e);
  linkboxes(e,a);
endfig;
```



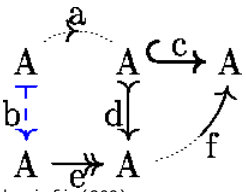

```
beginfig(266)
  boxit a(btex Essai etex);
  fill bpath a withcolor .5[red,white];
  drawboxed(a);

endfig;
```




```
beginfig(267)
  def drawredboxed(text t) =
    forsuffixes a=t:
      fill bpath a withcolor .5[red,white];
    endfor;
    drawboxed(t);
  enddef;

  circleit.a(btex D^^e9but etex);
  a.c = (0,0);
  a.dx = a.dy;
  circleit.b(btex Fin etex);
  b.c = (2cm,0);
  b.dx = b.dy;
  drawredboxed(a,b);
  drawarrow  a.c {dir 45} .. b.c {dir -45}
  cutbefore bpath.a cutafter bpath.b;
  drawarrow  b.c {dir -135} .. a.c {dir 135}
  cutbefore bpath.b cutafter bpath.a;

endfig;
```




```
beginfig(268)
  def drawcoloredboxed(expr c)(text t) =
    forsuffixes a=t:
      fill bpath a withcolor c;
    endfor;
    drawboxed(t);
  enddef;

  circleit.a(btex D^^e9but etex);
  a.c = (0,0);
  a.dx = a.dy;
  circleit.b(btex Fin etex);
  b.c = (2cm,0);
  b.dx = b.dy;
  drawcoloredboxed(.5[red,white],  a);
  drawcoloredboxed(.5[blue,white], b);
  drawarrow  a.c {dir 45} .. b.c {dir -45}
  cutbefore bpath.a cutafter bpath.b;
  drawarrow  b.c {dir -135} .. a.c {dir 135}
  cutbefore bpath.b cutafter bpath.a;

endfig;
```

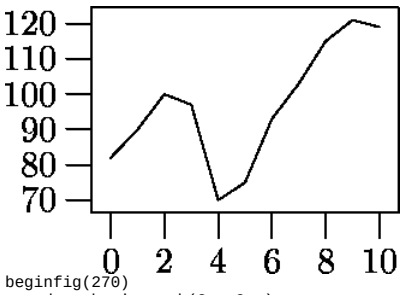



```
beginfig(269)
def begindiag =
  begingroup;
  save _diag_x, _diag_x_max, _diag_y, _diag_y_max, _diag;
  numeric _diag_x, _diag_x_max, _diag_y, _diag_y_max;
  string _diag[][];
  % Num^^e9ro de ligne et de colonne courrants
  _diag_x = -1; _diag_y = 0;
  % Num^^e9ro de ligne et de colonne maximaux
  _diag_x_max = _diag_y_max = 0;
  save _diag_ar_n, _diag_ar_source, _diag_ar_but, _diag_ar_up, _diag_ar_down;
  % Nombre de fl^^e8ches
  numeric _diag_ar_n; _diag_ar_n=-1;
  % Source et but de la fl^^e8che
  pair _diag_ar_source[], _diag_ar_but[];
  % Ce qu'il faut ^^e9crire au dessus ou au dessous
  string _diag_ar_up[], _diag_ar_down[];
  save _diag_ar_curved, _diag_ar_shape, _diag_ar_color, _diag_ar_width;
  % ^^ab courbure ^^bb (c'est une distance)
  numeric _diag_ar_curved[];
  % Forme de la fl^^e8che
  string _diag_ar_shape[];
  % Couleur, ^^e9paisseur, pointill^^e9s
  color _diag_ar_color[];
  numeric _diag_ar_width[];
  picture _diag_ar_dashed[];
enddef;

def node expr A =
  _diag_x := _diag_x + 1;
  _diag_x_max := max(_diag_x,_diag_x_max);
  _diag[_diag_x][_diag_y] := A;
enddef;
```

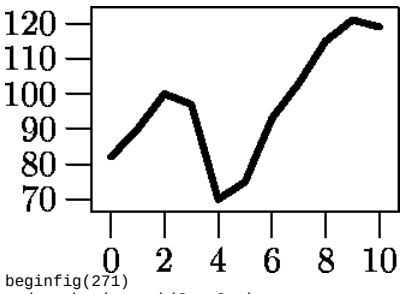
```
def nextline =
  _diag_x := -1;
  _diag_y := _diag_y + 1;
  _diag_y_max := max(_diag_y, _diag_y_max);
enddef;

tertiarydef a => b = a, b enddef;
def even (expr a) = not odd(a) enddef;

vardef rarrowto(expr a,b)(text t) =
  save i,p;
  _diag_ar_n := _diag_ar_n + 1;
  _diag_ar_source[_diag_ar_n] = (_diag_x, _diag_y);
  _diag_ar_but[_diag_ar_n] = (_diag_x + a, _diag_y + b);

  numeric i; i:=0;
  string current;
  for p=t:
    if even(i):
      current := p;
    else:
      if current = "above":
        _diag_ar_up[_diag_ar_n] = p;
      elseif current = "below":
        _diag_ar_down[_diag_ar_n] = p;
      elseif current = "shape":
        _diag_ar_shape[_diag_ar_n] = p;
      elseif current = "curved":
        _diag_ar_curved[_diag_ar_n] = p;
      elseif current = "color":
        _diag_ar_color[_diag_ar_n] = p;
      elseif current = "width":
        _diag_ar_width[_diag_ar_n] = p;
      elseif current = "dashed":
        _diag_ar_dashed[_diag_ar_n] = p;
      else:
        errmessage("rarrowto: Wrong argument "&ditto¤t&ditto);
      fi;
    fi;
    i := i + 1;
  endfor;
  if odd i:
    errmessage("rarrowto: Odd number of arguments "&decimal(i));
  fi;
enddef;

%% Les t^^eates de fl^^e8ches

picture withsmalldots, notdashed;
withsmalldots := withdots scaled .3;
notdashed := dashpattern(on 50cm);

vardef diag_arrow_head (expr p, t) =
  save A,B,C,u; pair A,B,C,u;
  B := point t of p;
  u := -unitvector(direction t of p);
  A := B + ahlength*u rotated(-ahangle);
  C := B + ahlength*u rotated(+ahangle);
  A .. {-u} B {u} .. C
enddef;

vardef diag_arrow_bar (expr p, t) =
  save A,B,C,u; pair A,B,C,u;
  B := point t of p;
  u := unitvector(direction t of p);
  A := B + ahlength*sind(ahangle)*u rotated(90);
  C := B + ahlength*sind(ahangle)*u rotated(-90);
  A .. B .. C
enddef;

%% Les fl^^e8ches

def diag_draw_arrow_default(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  draw pp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
enddef;

def diag_draw_arrow_middle(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  draw pp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head(p,1)
    withcolor col withpen pencircle scaled w;
enddef;

def diag_draw_arrow_epi(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  draw pp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
  path ppp;
  ppp := pp cutafter (fullcircle scaled 1mm shifted point length(pp) of pp);
  draw diag_arrow_head(ppp, length(ppp))
```

```
    withcolor col withpen pencircle scaled w;
enddef;

def diag_draw_arrow_mono(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  path ppp;
  ppp := pp cutbefore (fullcircle scaled 1mm shifted point 0 of pp);
  draw ppp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
  draw diag_arrow_head(ppp, 0)
    withcolor col withpen pencircle scaled w;
enddef;

vardef diag_draw_arrow_inj(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  path ppp;
  ppp := pp cutbefore (fullcircle scaled 1mm shifted point 0 of pp);
  draw ppp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
  save u,A,B,C;
  pair u,A,B,C;
  A := point 0 of ppp;
  u := unitvector(direction 0 of ppp);
  B := A + ahlength*(-u) rotated (-ahangle);
  C := A + 2 ahlength*sind(ahangle)*u rotated 90;
  draw C {-u} .. B .. A {u}
  withcolor col withpen pencircle scaled w dashed dash;
enddef;

def diag_draw_arrow_mapsto(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  draw pp
    withcolor col withpen pencircle scaled w dashed dash;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
  draw diag_arrow_bar (pp, 0)
    withcolor col withpen pencircle scaled w;
enddef;

def diag_draw_arrow_half_dotted(suffix a,b)(expr curved, w, col, dash) =
  p = a.c ..
  (1/2 [a.c,b.c] + curved*unitvector(b.c-a.c) rotated 90)
  .. b.c;
  pp := p cutbefore bpath.a cutafter bpath.b;
  draw subpath(0,1) of pp
    withcolor col withpen pencircle scaled w dashed withsmalldots;
  draw subpath(1,2) of pp
    withcolor col withpen pencircle scaled w;
  draw diag_arrow_head (pp, length(pp))
    withcolor col withpen pencircle scaled w;
enddef;

%% Fin des fl^^e8ches

def color_to_string (expr a) =
  "("&
    decimal(redpart a)
    &","&
    decimal(greenpart a)
    &","&
    decimal(bluepart a)
    &")"
enddef;

def enddiag =
  save i,j,k,l,mm,a,A,p,b;
  for i=0 upto _diag_x_max:
    for j=0 upto _diag_y_max:
      if known _diag[i][j]:
        circleit.a[i][j]( _diag[i][j] );
        a[i][j].dx = a[i][j].dy;
        a[i][j].c = 1cm * (i,-j);
        drawunboxed( a[i][j] );
      fi;
    endfor;
  endfor;
  for m=0 upto _diag_ar_n:
    % V^^e9rifier que le but existe
    i := xpart _diag_ar_source[m];
    j := ypart _diag_ar_source[m];
    k := xpart _diag_ar_but[m];
    l := ypart _diag_ar_but[m];

    % On trace la fl^^e8che. Le chemin est mis dans la variable p.
    path p,pp;
    if unknown _diag_ar_shape[m]:  _diag_ar_shape[m]  := "default" fi;
    if unknown _diag_ar_color[m]:  _diag_ar_color[m]  := black     fi;
    if unknown _diag_ar_width[m]:  _diag_ar_width[m]  := .5bp      fi;
    if unknown _diag_ar_curved[m]: _diag_ar_curved[m] := 0         fi;
    if unknown _diag_ar_dashed[m]: _diag_ar_dashed[m] := notdashed fi;
    % On ne peut PAS utiliser m dans une cha^^eene ce caract^^e8res que l'on donne
    % ^^e0 scantokens, car m est une variable de boucle. C'est vraiment sp^^e9cial,
    % une variable de boucle.
    mm := m;
```

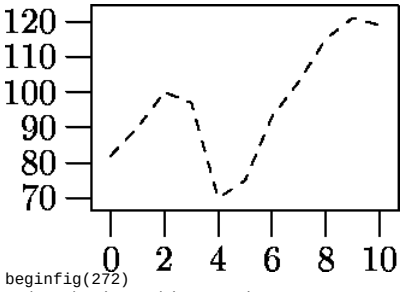
```
    scantokens(
      "diag_draw_arrow_"& _diag_ar_shape[m]
      &"("
        &"a[i][j], a[k][l],_diag_ar_curved[mm],_diag_ar_width[mm],"
        &"_diag_ar_color[mm],_diag_ar_dashed[mm]"
        &");"
      );

    % On ^^e9crit des choses au dessus ou au dessous des fl^^e8ches
    pair A;
    A = point 1/2 length(p) of p;
    if known _diag_ar_up[m]:
      boxit.b[m](_diag_ar_up[m]);
      b[m].c = A + 4bp*unitvector(direction 1/2 length(p) of p rotated 90);
      drawunboxed(b[m]);
    fi;
    if known _diag_ar_down[m]:
      boxit.c[m](_diag_ar_down[m]);
      c[m].c = A + 4bp*unitvector(direction 1/2 length(p) of p rotated -90);
      drawunboxed(c[m]);
    fi;
  endfor;
  endgroup;
enddef;

  begindiag;
    node "A";
      rarrowto(1,0, "above" => "a",
               "shape" => "middle",
               "curved" => 3mm,
               "dashed" => withsmalldots);
      rarrowto(0,1, "below" => "b",
               "color" => blue,
               "shape" => "mapsto",
               "dashed" => evenly);
    node "A";
      rarrowto(1,0, "above" => "c", "width" => 1bp, "shape" => "inj");
      rarrowto(0,1, "below" => "d", "shape" => "mono");
    node "A";
    nextline;
    node "A";
      rarrowto(1,0, "below" => "e", "shape" => "epi");
    node "A";
      rarrowto(1,-1, "below" => "f", "curved" => -3mm, "shape" => "half_dotted");
  enddiag;
endfig;
```




```
beginfig(270)
    draw begingraph(3cm,2cm)
      gdraw "data1";
    endgraph;

endfig;
```

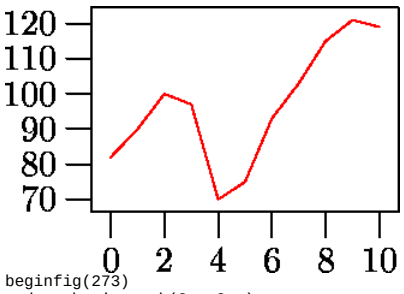



```
beginfig(271)
  draw begingraph(3cm,2cm)
    gdraw "data1" withpen pencircle scaled 2bp;
  endgraph;
endfig;
```

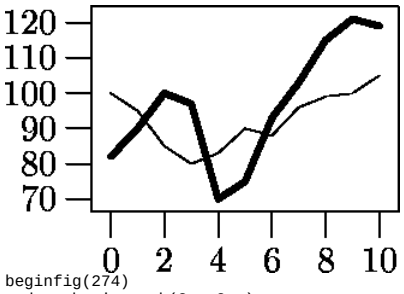



```
beginfig(272)
  draw begingraph(3cm,2cm)
    gdraw "data1" dashed evenly;
```

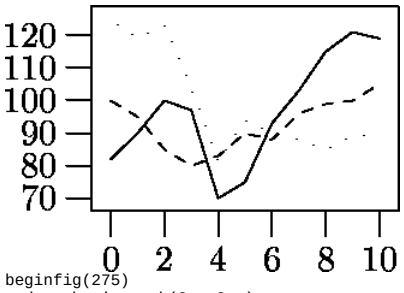
```
  endgraph;
endfig;
```



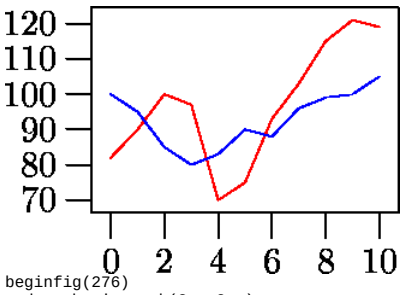

```
beginfig(273)
  draw begingraph(3cm,2cm)
    gdraw "data1" withcolor red;
  endgraph;
endfig;
```



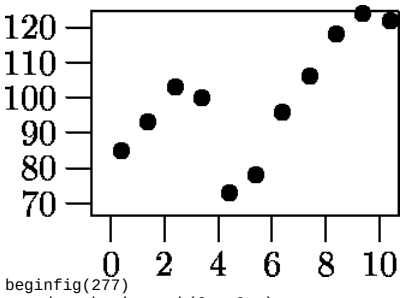

```
beginfig(274)
  draw begingraph(3cm,2cm)
    gdraw "data1" withpen pencircle scaled 2bp;
    gdraw "data2" ;
  endgraph;
endfig;
```




```
beginfig(275)
  draw begingraph(3cm,2cm)
    gdraw "data1" ;
    gdraw "data2" dashed evenly;
    gdraw "data3" dashed withdots;
  endgraph;
endfig;
```

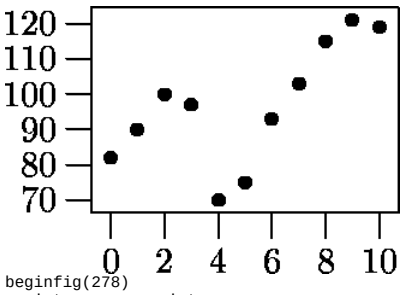



```
beginfig(276)
  draw begingraph(3cm,2cm)
    gdraw "data1" withcolor red;
    gdraw "data2" withcolor blue;
  endgraph;
endfig;
```

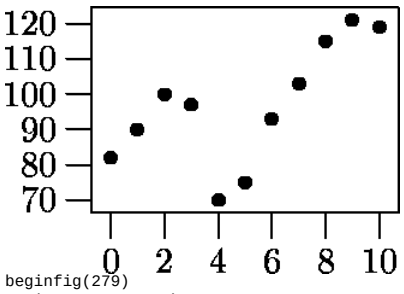



```
beginfig(277)
    draw begingraph(3cm,2cm)
```

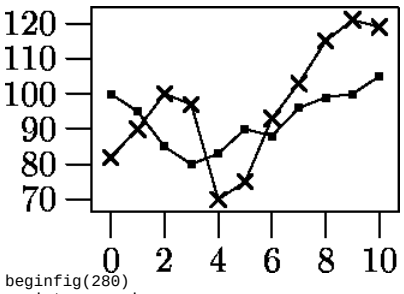
```
      gdraw "data1" plot btex $\bullet$ etex;
    endgraph;

endfig;
```



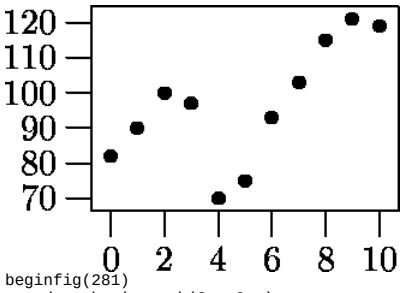

```
beginfig(278)
  picture gros_point;
  draw (0,0) withpen pencircle scaled 4bp;
  gros_point := currentpicture;
  currentpicture := nullpicture;
  draw begingraph(3cm,2cm)
    gdraw "data1" plot gros_point;
  endgraph;
endfig;
```




```
beginfig(279)
  picture gros_point;
  gros_point := nullpicture;
  addto gros_point doublepath (0,0)
        withpen pencircle scaled 4bp;
  draw begingraph(3cm,2cm)
    gdraw "data1" plot gros_point;
  endgraph;
endfig;
```

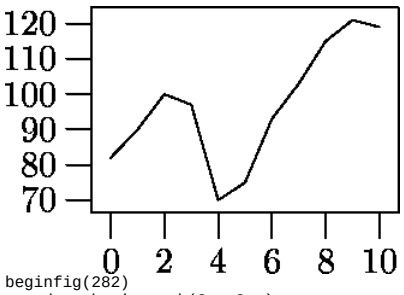



```
beginfig(280)
  picture croix;
  croix := nullpicture;
  addto croix doublepath (-2bp,2bp)--(2bp,-2bp)
              withpen pencircle scaled 1bp;
  addto croix doublepath (-2bp,-2bp)--(2bp,2bp)
              withpen pencircle scaled 1bp;
  picture gros_carre;
  gros_carre := nullpicture;
  addto gros_carre contour unitsquare shifted (-.5,-.5) scaled 2bp;

  draw begingraph(3cm,2cm)
    gdraw "data1";
    gdraw "data1" plot croix;
    gdraw "data2";
    gdraw "data2" plot gros_carre;
  endgraph;
endfig;
```

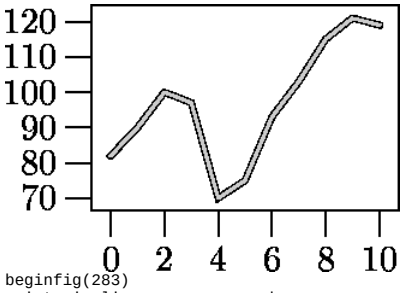



```
beginfig(281)
    draw begingraph(3cm,2cm)
      gdata("data1",
            v,
```

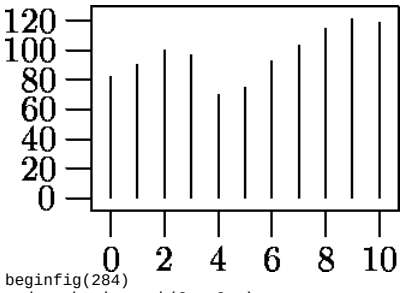
```
          glabel(gros_point, v1, v2);
         );
    endgraph;

endfig;
```



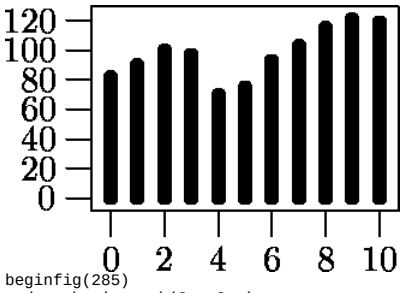

```
beginfig(282)
    draw begingraph(3cm,2cm)
      path p;
      gdata("data1",
            v,
            augment.p(v1,v2);
           );
      gdraw p;
    endgraph;

endfig;
```




```
beginfig(283)
  interim linecap := squared;
  interim linejoin := mitered;
  draw begingraph(3cm,2cm)
    path p;
    gdata("data1",
      v,
      augment.p(v1,v2);
      );
    gdraw p withpen pencircle scaled 2bp;
    gdraw p withpen pencircle scaled 1bp withcolor .8white;
  endgraph;
endfig;
```

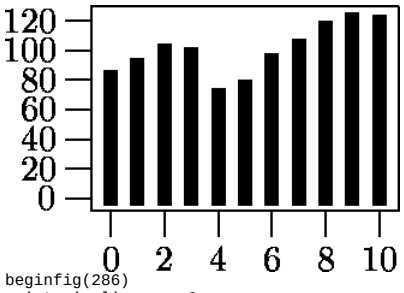



```
beginfig(284)
  draw begingraph(3cm,2cm)
    gdata("data1",
          v,
          path p;
          augment p (v1,0);
          augment p (v1,v2);
          gdraw p;
         );
  endgraph;
endfig;
```

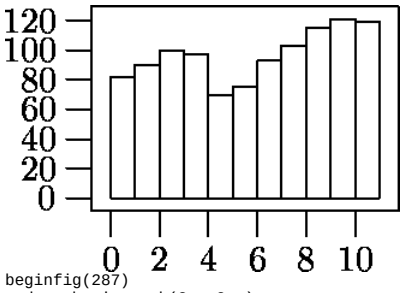



```
beginfig(285)
  draw begingraph(3cm,2cm)
    gdata("data1",
          v,
```

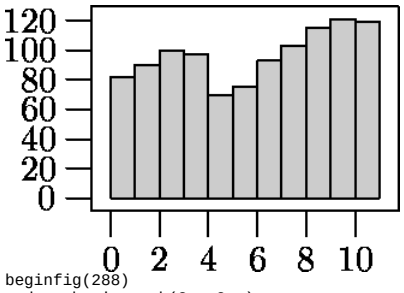
```
          path p;
          augment p (v1,0);
          augment p (v1,v2);
          gdraw p withpen pencircle scaled 4bp;
         );
  endgraph;
endfig;
```



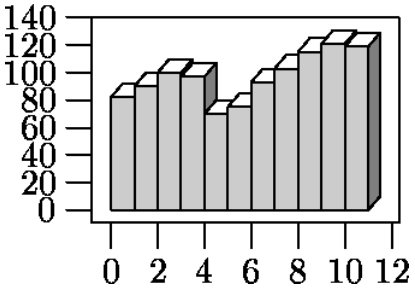

```
beginfig(286)
  interim linecap:=2;
  draw begingraph(3cm,2cm)
    gdata("data1",
          v,
          path p;
          augment p (v1,0);
          augment p (v1,v2);
          gdraw p withpen pencircle scaled 4bp;
         );
  endgraph;
endfig;
```



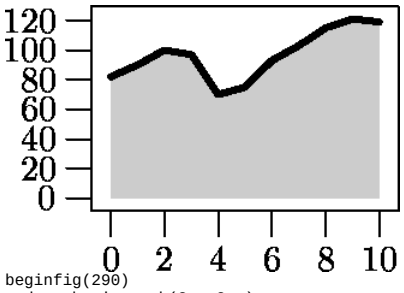

```
beginfig(287)
  draw begingraph(3cm,2cm)
    gdata("data1", v,
          path p;
          augment p (v1,0);
          augment p (v1,v2);
          augment p (v1 Sadd "1",v2);
          augment p (v1 Sadd "1",0);
          gdraw p--cycle;
         );
  endgraph;
endfig;
```



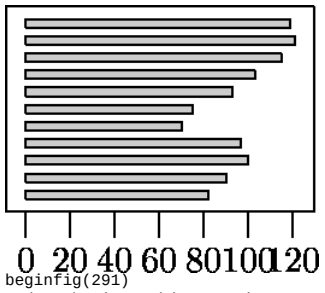

```
beginfig(288)
  draw begingraph(3cm,2cm)
    gdata("data1", v,
          path p;
          augment p (v1,0);
          augment p (v1,v2);
          augment p (v1 Sadd "1",v2);
          augment p (v1 Sadd "1",0);
          gfill p--cycle withcolor .8white;
          gdraw p--cycle;
          );
  endgraph;
endfig;
```



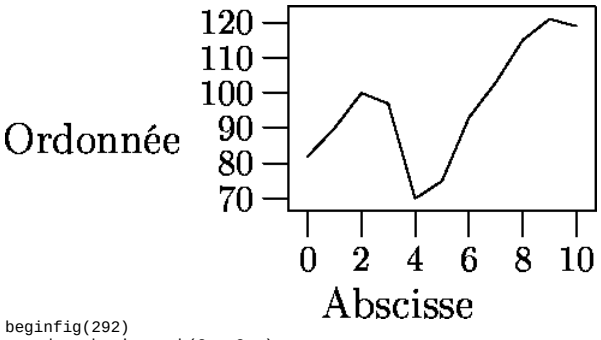
```
beginfig(289)
  draw begingraph(3cm,2cm)
    gdata("data1", v,

            % Le dessus
            path p;
            augment p (v1,v2);
            augment p (v1 Sadd "1",v2);
            augment p (v1 Sadd "1.5", v2 Sadd "10");
            augment p (v1 Sadd ".5",  v2 Sadd "10");
            gfill p--cycle withcolor white;
            gdraw p--cycle;

            % Le c^^f4t^^e9
            path p;
            augment p (v1 Sadd "1",0);
            augment p (v1 Sadd "1",v2);
            augment p (v1 Sadd "1.5", v2 Sadd "10");
            augment p (v1 Sadd "1.5", "10");
            gfill p--cycle withcolor .5white;
            gdraw p--cycle;

            % Le devant
            path p;
            augment p (v1,0);
            augment p (v1,v2);
            augment p (v1 Sadd "1",v2);
            augment p (v1 Sadd "1",0);
            gfill p--cycle withcolor .8white;
            gdraw p--cycle;

           );
 endgraph;
endfig;
```



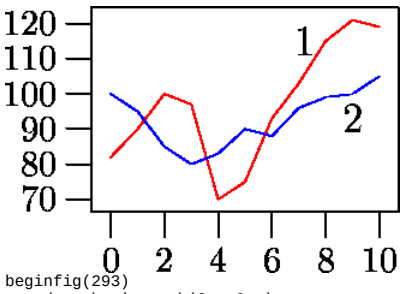

```
beginfig(290)
  draw begingraph(3cm,2cm)
    path p,q;
    gdata("data1", v,
          augment.p(v1,v2);
         );
    q:= (xpart point 0 of p, 0) -- p --
      (xpart point length(p) of p, 0) -- cycle;
    gfill q withcolor .8white;
    gdraw p withpen pencircle scaled 2bp;
  endgraph;
endfig;
```



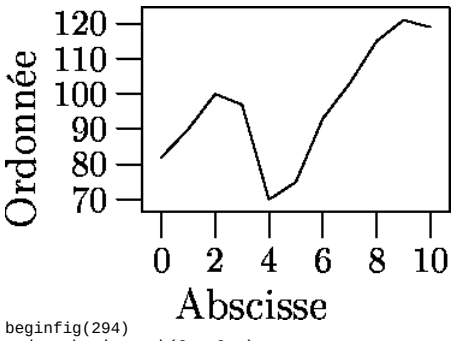

```
beginfig(291)
  draw begingraph(3cm,2cm)
    gdata("data1",
      v,
      path p;
      augment p ("0", i);
      augment p (v2, i);
      augment p (v2, i Sadd ".5");
      augment p (0, i Sadd ".5");
      gfill p--cycle withcolor .8white;
      gdraw p--cycle;
      );
    autogrid(otick.bot,);
  endgraph;
endfig;
```



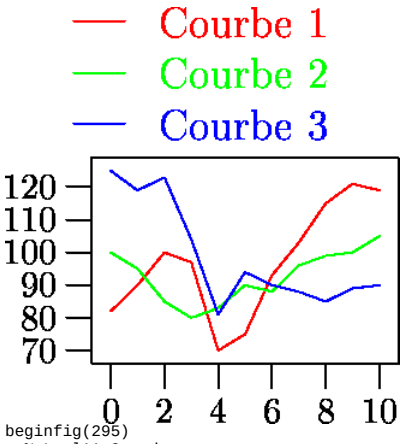

```
beginfig(292)
    draw begingraph(3cm,2cm)
      gdraw "data1";
      glabel.bot(btex Abscisse etex, OUT);
      glabel.lft(btex Ordonn^^e9e etex, OUT);
    endgraph;

endfig;
```




```
beginfig(293)
    draw begingraph(3cm,2cm)
      gdraw "data1" withcolor red;
      glabel.lft(btex 1 etex, 8);
      gdraw "data2" withcolor blue;
      glabel.bot(btex 2 etex, 9);
    endgraph;

endfig;
```

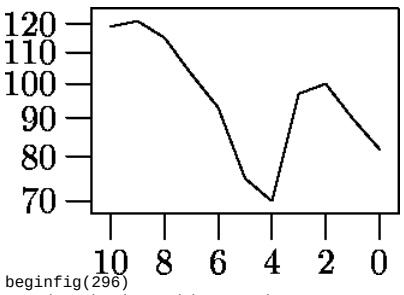



```
beginfig(294)
  draw begingraph(3cm,2cm)
    gdraw "data1";
    glabel.bot(btex Abscisse etex, OUT);
    glabel.lft(btex Ordonn^^e9e etex rotated 90, OUT);
  endgraph;
endfig;
```

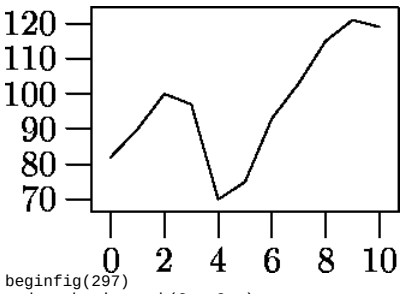



```
beginfig(295)
  % La l^^e9gende
  color couleur[];
  couleur[0] := red;
  couleur[1] := green;
  couleur[2] := blue;
  picture p[], legende;
  legende := nullpicture;
```

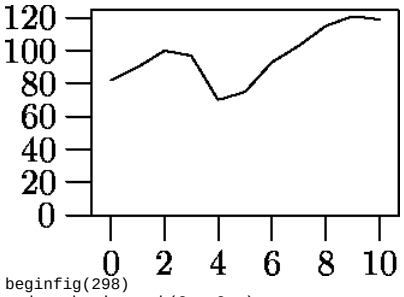
```
  for i=0 upto 2:
    p[i] := nullpicture;
    addto p[i] also TEX("Courbe " & decimal(i+1))
                withcolor couleur[i];
    addto legende doublepath ( (0,0)--(5mm,0) )
                  shifted (0, -5mm*i)
                  withpen currentpen withcolor couleur[i];
    addto legende also p[i]
                  shifted 1/2(lrcorner p[i] - ulcorner p[i])
                  shifted (0, -5mm*i);
  endfor;
  % Les courbes
  draw begingraph(3cm,2cm)
    for i=1 upto 3:
      gdraw "data" & decimal(i) withcolor couleur[i-1];
    endfor;
    glabel.top(legende,OUT);
  endgraph;
endfig;
```



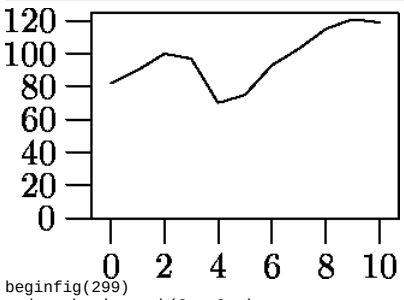

```
beginfig(296)
    draw begingraph(3cm,2cm)
      setcoords(-linear,log);
      gdraw "data1";
    endgraph;

endfig;
```




```
beginfig(297)
  draw begingraph(3cm,2cm)
    gdraw "data1";
  endgraph;

endfig;
```

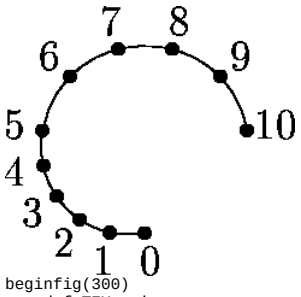



```
beginfig(298)
  draw begingraph(3cm,2cm)
    setrange( (whatever,0), (whatever,whatever) );
    gdraw "data1";
  endgraph;

endfig;
```

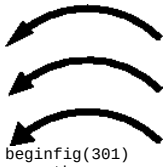



```
beginfig(299)
  draw begingraph(3cm,2cm)
    setrange(whatever, 0, whatever, whatever);
    gdraw "data1";
  endgraph;
```

```
endfig;
```

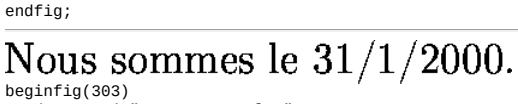



```
beginfig(300)
 vardef TEX primary s =
   write "verbatimtex"                    to "mptextmp.mp";
   write "\documentclass[12pt]{article}"  to "mptextmp.mp";
   write "\usepackage[T1]{fontenc}"       to "mptextmp.mp";
   write "\usepackage{amsmath,amssymb}"   to "mptextmp.mp";
   write "\begin{document}"               to "mptextmp.mp";
   write "etex"                           to "mptextmp.mp";
   write "btex "&s&" etex"                to "mptextmp.mp";
   write EOF                              to "mptextmp.mp";
   scantokens "input mptextmp"
 enddef;
   vardef mylabel(expr pic, p, t) =
     save A; pair A;
     A = point t of p +
         8bp * unitvector(direction t of p) rotated 90;
     label(pic, A);
   enddef;
   path p; u:=1cm;
   p = (0,0)..(-u,u)..(u,u);
   draw p;
   for i=0 step .2 until length(p):
     draw point i of p withpen pencircle scaled 4bp;
     mylabel(TEX("$"&decimal(round(5*i))&"$"),p,i);
   endfor;

endfig;
```

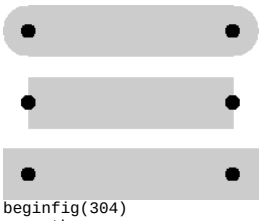


```
beginfig(301)
   path p;
   p := subpath(1,3) of fullcircle scaled 2cm;
   interim linejoin := mitered;
   interim linecap := butt;

   interim ahangle := 30;
   drawarrow p withpen pencircle scaled 2bp;

   interim ahangle := 45;
   drawarrow p shifted (0,-5mm) withpen pencircle scaled 2bp;

   interim ahangle := 60;
   drawarrow p shifted (0,-10mm) withpen pencircle scaled 2bp;

endfig;
```

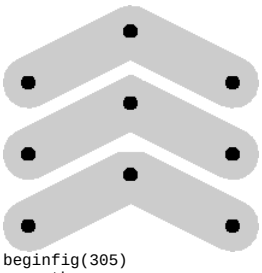
```
beginfig(302)
   path p;
   p := subpath(1,3) of fullcircle scaled 2cm;
   interim linejoin := mitered;
   interim linecap := butt;
   interim ahangle := 30;

   interim ahlength := 2bp;
   drawarrow p withpen pencircle scaled 2bp;

   interim ahlength := 4bp;
   drawarrow p shifted (0,-5mm) withpen pencircle scaled 2bp;

   interim ahlength := 6bp;
   drawarrow p shifted (0,-10mm) withpen pencircle scaled 2bp;

endfig;
```




```
beginfig(303)
   draw TEX( "Nous sommes le "&
             decimal(day)   &"/"&
             decimal(month) &"/"&
             decimal(year) &"."    );

endfig;
```



```
beginfig(304)
   path p;
   p := (0,0) -- (2cm,0);
   def doit (suffix p)(expr t) =
     begingroup
       interim linecap := t;
       draw p withpen pencircle scaled 5mm withcolor .8white;
     endgroup;
     draw point 0 of p withpen pencircle scaled 4bp;
     draw point 1 of p withpen pencircle scaled 4bp;
     p := p shifted (0,-7mm)
   enddef;
   doit(p,rounded);
   doit(p,butt);
   doit(p,squared);

endfig;
```



```
beginfig(305)
   path p;
   p := (0,0) -- (1cm,5mm) -- (2cm,0);
   def doit (suffix p)(expr t) =
     begingroup
       interim linejoin := t;
       draw p withpen pencircle scaled 5mm withcolor .8white;
     endgroup;
     draw point 0 of p withpen pencircle scaled 4bp;
     draw point 1 of p withpen pencircle scaled 4bp;
     draw point 2 of p withpen pencircle scaled 4bp;
     p := p shifted (0,-7mm)
   enddef;
   doit(p,rounded);
   doit(p,mitered);
   doit(p,beveled);

endfig;
```

```
bye;
```