`vec2d` is a package to draw two dimensional mathematical figures. Needed `fp.tex` (fixed point math package) and `pstricks.tex`.

A picture consists of:

- a list of definitions and calculations of points and lines

- the start of drawing: `\VDDdcanvas` or `\VDDdmincanvas`

- drawing and labeling operations and possibly additional definitions and calculations

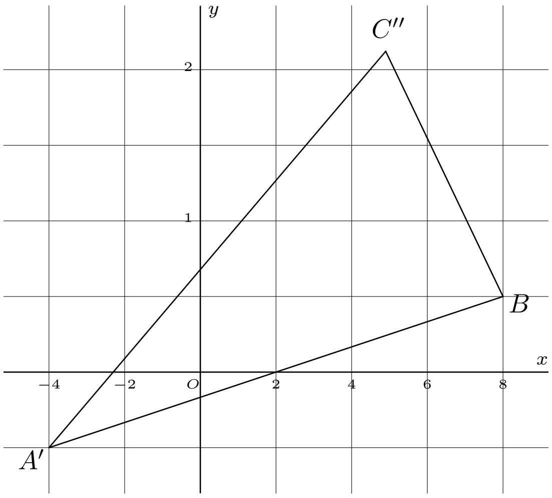
- and end marker `\VDDdcanvasend` or `\VDDdmincanvasend`

Points can be named `A-Z` with any number of accents (`'`). `\VDDdef` is used to define points in cartesian and polar coordinates. A comma signifies cartesian and a semicolon polar (in degrees).

Example: `\VDDdef O(0,0)A(3,1)B'(4;45)` defines the points $O(0,0)$, $A(3,1)$ and $B'(r = 4, \phi = 45°)$.

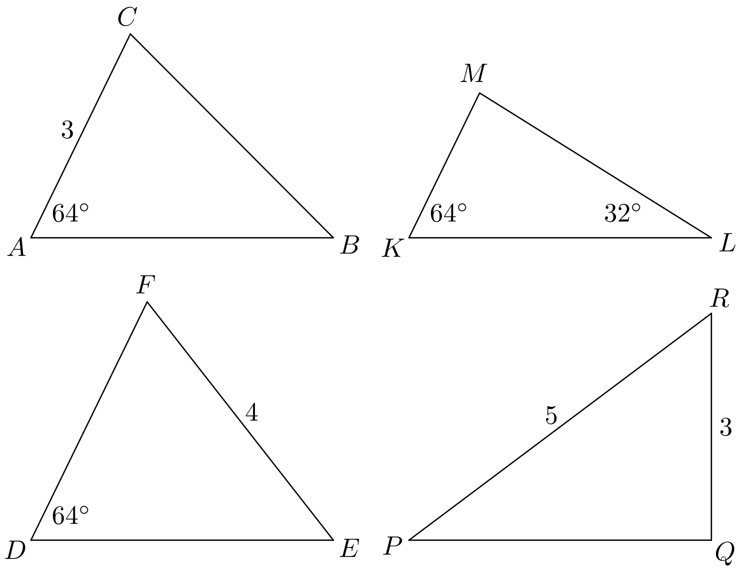
A complete example

```
\VDDdef A'(-2,-1)B(4,1)C''(4.9;60)
\VDDdcanvas1:1 % the points defined up to now determine the viewport
  \VDDdruitjes
  \VDDdoorsprong
  \VDDdhoraxis{1}{2}{x}% the second argument is increase per label
  \VDDdveraxis{2}{1}{y}% the first argument is space between labels
  \VDDdlpolygon A'BC''
\VDDdcanvasend
```



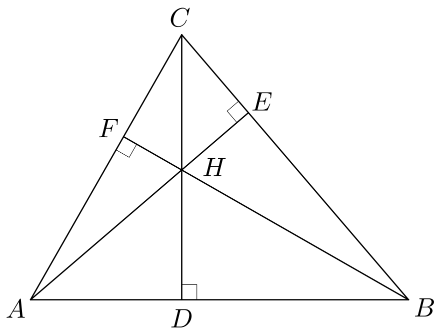
Several commands are available to construct triangles

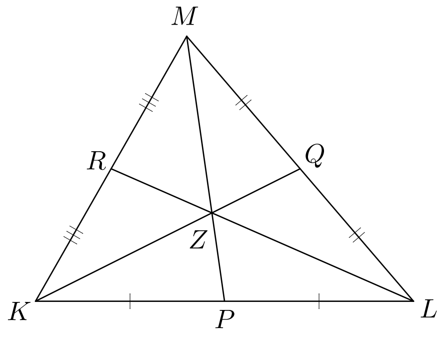
```
\VDDtriangleonesidesameangle C:A(0,0)B(4,0):3,64    % points can be defined
\VDDtriangleonesideotherangle F:D(0,-4)E(4,-4):4,64 % at first usage
\VDDtriangletwoangles M:K(5,0)L(9,0):64,32
\VDDtriangletwolengths R:P(5,-4)Q(9,-4):5,3
\VDDdcanvas1:1
  \VDDdlpolygon ABC
  \VDDllength CA
  \VDDlpointautoangle BAC
  \VDDdlpolygon DEF
  \VDDllength EF
  \VDDlpointautoangle EDF
  \VDDdlpolygon KLM
  \VDDlpointautoangle MLK
  \VDDlpointautoangle LKM
  \VDDdlpolygon PQR
  \VDDllength QR
  \VDDllength RP
\VDDdcanvasend
```



Special lines in triangles. If you want to know the intersection of two lines, you need to calculate it.

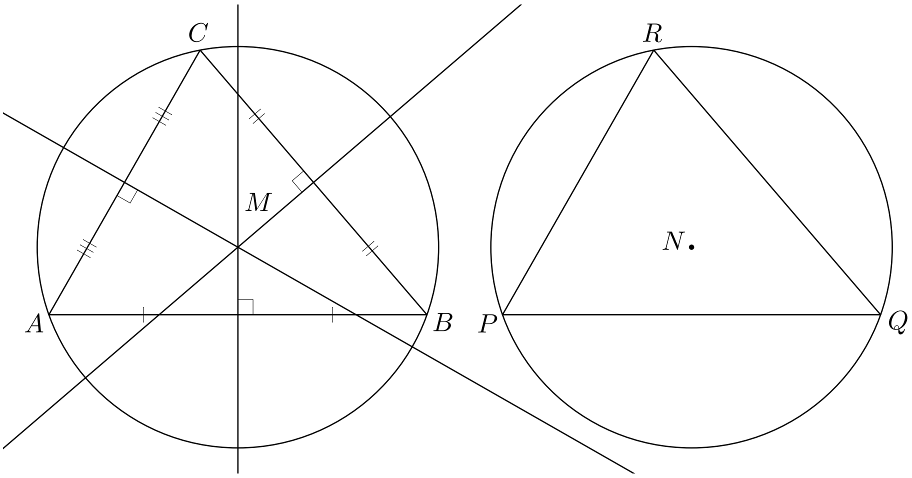
```
\VDDcpleadlinepart F:A(0,0)C(2,3.5):B(5,0) % calculate basepoint of perpendicular
\VDDdef K(0,-5)M(2,-1.5)L(5,-5)
\VDDcavg P:KL                % calculate midpoint (average) of KL
\VDDcavg Q:LM
\VDDcavg R:MK
\VDDdcanvas1:1
  \VDDdlpolygon ABC
  \VDDdline BF                % draw BF
  \VDDlpointletter CFA        % label the point
  \VDDdperp CFB               % display symbol at F
  \VDDcdlpleadlinepart D:AB:C % do it all at once
  \VDDcdlpleadlinepart E:BC:A
  \VDDcppintersect H:AE:CD    % calculate intersection of AE en CD
  \VDDlpointletter BHE        % label H
  \VDDdlpolygon KLM
  \VDDdline MP
  \VDDlpointletter KPL
  \VDDdlinepartequal KPLP     % KP = LP
  \VDDdline KQ
  \VDDlpointletter LQM
  \VDDdlinepartequal LQMQ
  \VDDdline LR
  \VDDlpointletter MRK
  \VDDdlinepartequal MRKR
  \VDDcppintersect Z:MP:LR
  \VDDlpointletter KZP
\VDDdcanvasend
```





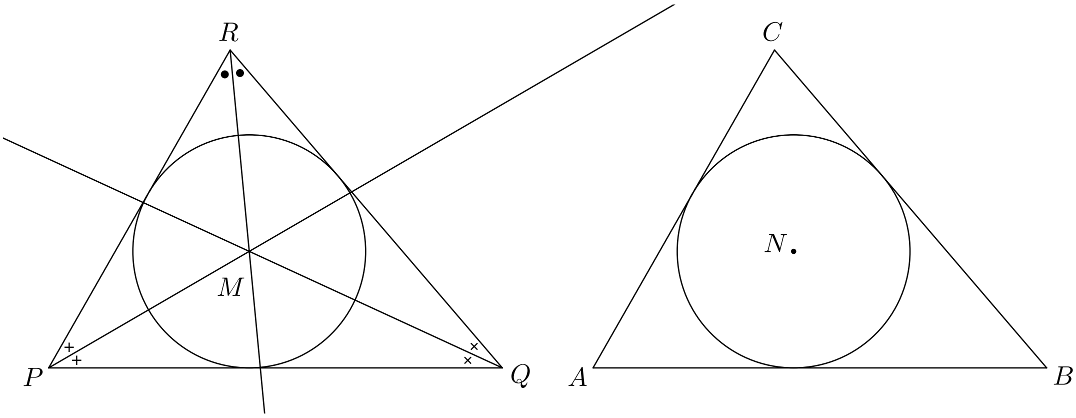
More special lines in triangles.

```
\VDDdef A(0,0)B(5,0)C(2,3.5)
\VDDdef P(6,0)Q(11,0)R(8,3.5)
\VDDdef S(0,-1.5) % stretch viewport to see complete circles
\VDDdcanvas1:1
  \VDDdlpolygon ABC
  \VDDdlpolygon PQR
  \VDDccircumscribedcircle\length:N:PQR
  \VDDdcircle N:\length{}
  \VDDdpoint N
  \VDDlcirclepointletter N:Q:1
  \VDDcavg D:AB
  \VDDcavg E:BC
  \VDDcavg F:CA
  \VDDdeflinepoint n:AB
  \VDDcleadline k:n:D
  \VDDdeflinepoint n:BC
  \VDDcleadline l:n:E
  \VDDdeflinepoint n:CA
  \VDDcleadline m:n:F
  \VDDdwholeline k
  \VDDdwholeline l
  \VDDdwholeline m
  \VDDcintersect M:kl
  \VDDdperp BDM
  \VDDdperp CEM
  \VDDdperp AFM
  \VDDdlinepartequal ADBD
  \VDDdlinepartequal BECE
  \VDDdlinepartequal CFAF
  \VDDllinelinepointletter kl:MA:1:-1
  \VDDcplength\length:AM
  \VDDdcircle M:\length{}
\VDDdcanvasend
```



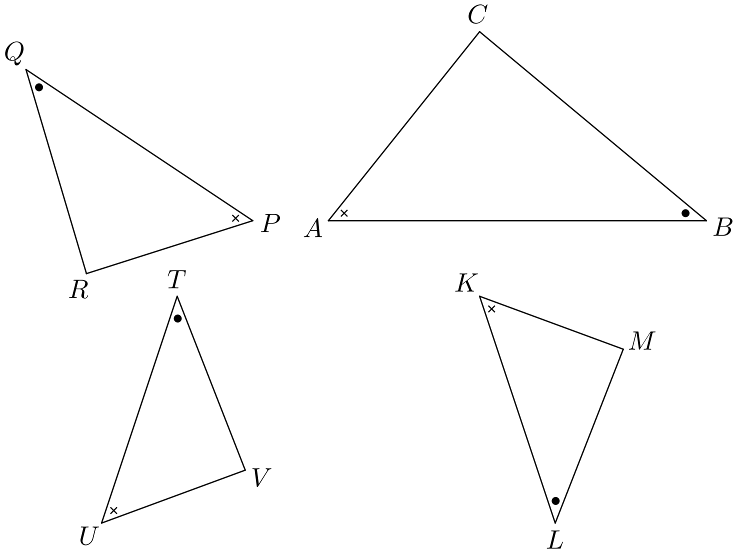
More special lines in triangles.

```
\VDDdef P(0,0)Q(5,0)R(2,3.5)
\VDDdef A(6,0)B(11,0)C(8,3.5)
\VDDdcanvas1.2:1
  \VDDdlpolygon ABC
  \VDDcinscribedcircle\length:N:ABC
  \VDDlcirclepointletter N:B:1
  \VDDdpoint N
  \VDDdcircle N:\length{}
  \VDDdlpolygon PQR
  \VDDcdeellijn l:PQR
  \VDDdhalfline l
  \VDDcdeellijn l:RPQ
  \VDDdhalfline l
  \VDDcdeellijn m:QRP
  \VDDdhalfline m
  \VDDcintersect M:lm
  \VDDllinelinepointletter ml:MP:-1:1
  \VDDdeflinepoint l:PQ
  \VDDcleadline m:l:M
  \VDDcintersect S:lm
  \VDDcplength\length:MS
  \VDDdcircle M:\length{}
  \VDDlangleequal PQMMQR
  \VDDlangleequal QRMMRP
  \VDDlangleequal RPMMPQ
\VDDdcanvasend
```



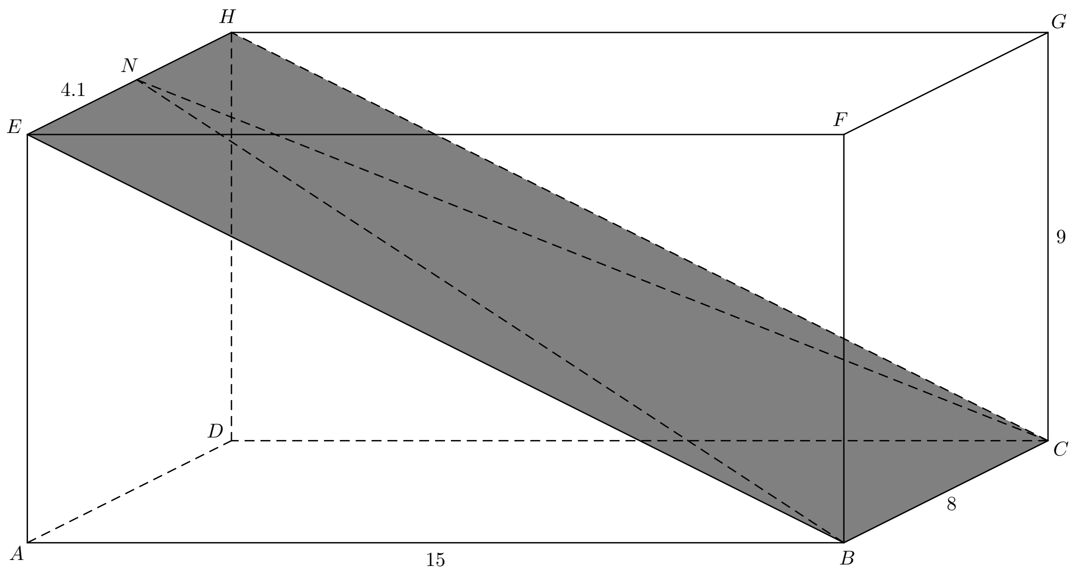
Similar triangles

```
\VDDdef A(0,0)B(5,0)C(2,2.5)
\VDDtrianglegelijkvormig R:P(-1,0)Q(-4,2):ABC
\VDDtrianglegelijkvormig M:K(2,-1)L(3,-4):ABC
\VDDtrianglegelijkvormig V:T(-2,-1)U(-3,-4):BAC
\VDDcanvas1:1
  \VDDdlpolygon ABC
  \VDDdlpolygon PQR
  \VDDdlpolygon KLM
  \VDDdlpolygon TUV
  \VDDlangleequal CABRPQMKLTUV
  \VDDlangleequal ABCPQRKLMVTU
\VDDcanvasend
```



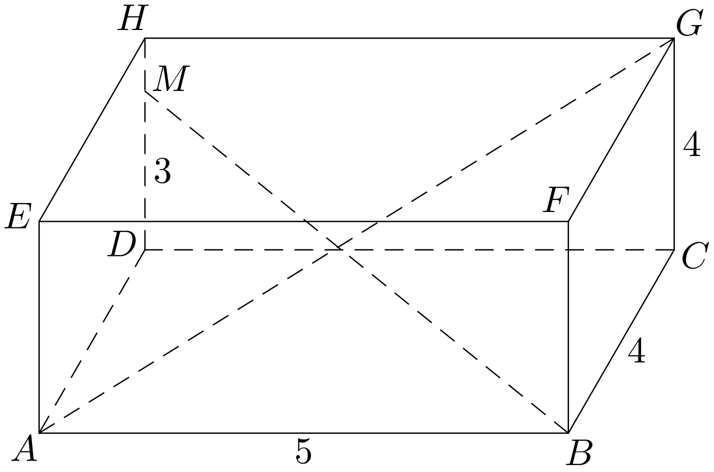
Fake 3d drawing with translate.

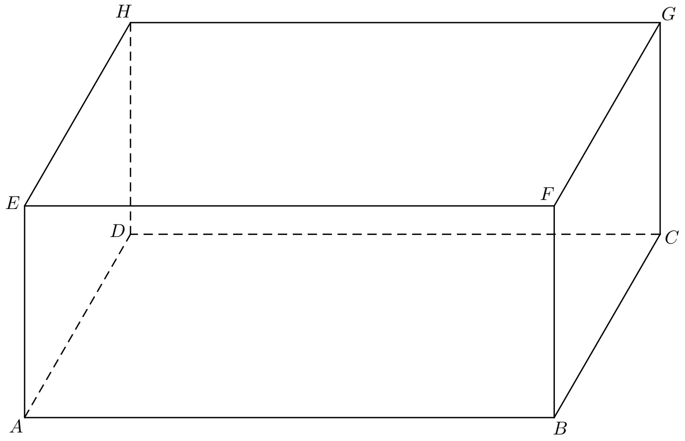
```
\VDDctranslate C:D(2,1):A(0,0)B(8,0) % create C by translating D over A->B
\VDDctranslate FGH:BCD:AE(0,4)      % create FGH by translating BCD over A->E
\VDDcabstransp N:E:EH:1.2
\VDDdcanvas 1.8:0.75                 % shrink everything, enlarge coordinates
  \VDDdfillpolygon{fillcolor=gray,fillstyle=solid}BCHE % this comes first
  \VDDdlbalk ABCDEFGH                % draw '3d' object
  \VDDdhiddenline BNC                % a hidden line is not drawn solid
  \VDDlpointletter HNE
  \VDDdline BE
  \VDDdhiddenline CH
  \VDDllinepart AB:$15$              % label linepart with explicit values
  \VDDllinepart BC:$8$
  \VDDllinepart CG:$9$
  \VDDllinepart NE:$4.1$
\VDDdcanvasend
```



Fake 3d drawing with translate (polar coordinates with custom origin)

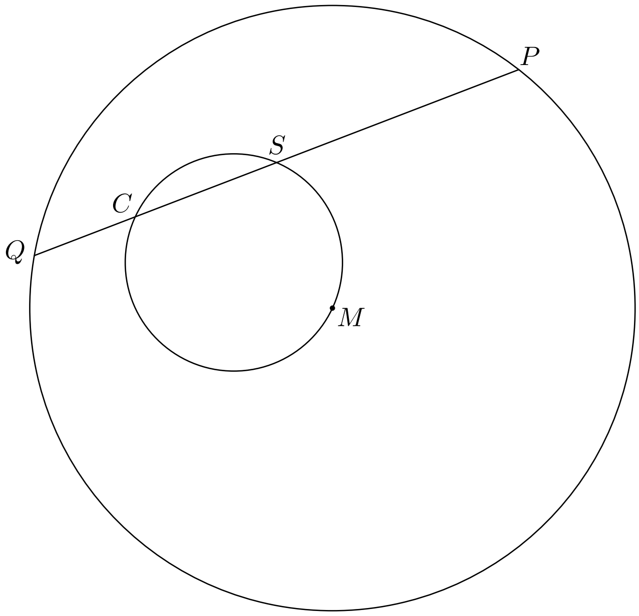
```
\VDDcabstransangle DC:A(0,0)B(10;0):60:4 % draw C(4;60) D(4;60) with A B as origin
\VDDcabstransangle EFGH:ABCD:90:4 % EFGH with ABCD as origin 4 up
\VDDcabstransp M:D:DH:3    % translate M starting from D 3 units in direction of DH
\VDDdcanvas.5:1.4  % notice the size of the labels
  \VDDdlbalk ABCDEFGH
  \VDDlpointletter BMH
  \VDDllinepart CG:$4$ \VDDllinepart BC:$4$ \VDDllinepart AB:$5$
  \VDDllinepart DM:$3$
  \VDDdhiddenline AG
  \VDDdhiddenline BM
\VDDdcanvasend
% now with different scaling
\VDDcabstransangle DC:A(0,0)B(10;0):60:4 % draw C(4;60) D(4;60) with A B as origin
\VDDcabstransangle EFGH:ABCD:90:4 % EFGH wit ABCD as origin 4 up
\VDDdcanvas1:.7 % notice the size of the labels
  \VDDdlbalk ABCDEFGH
\VDDdcanvasend
```





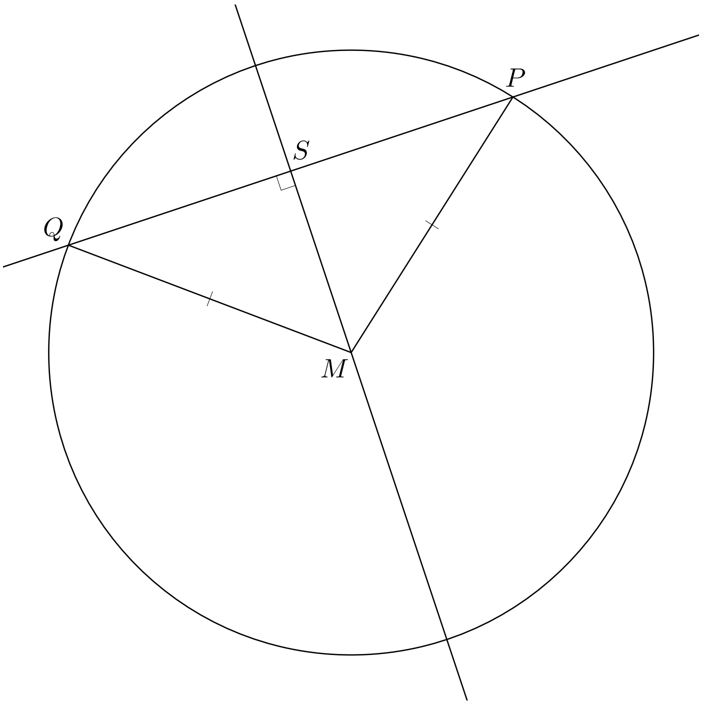
Example of 'real world' construction

```
\VDDdef M(0,0)L(-4,-4)R(4,4) % L and R stretch the viewport to keep the circle visible
\VDDdcanvas 1:1
  \VDDdcircle M:4
  \VDDdpoint M % draw single point
  \VDDdline P(4;52)Q(4;170) % draw line with endpoints on circle
  \VDDcavg S:PQ
  \VDDcabstransp C:S:SQ:2   % go 2 spaces from S in direction of SQ, call result C
  \VDDcavg N:CM
  \VDDcplength\length:NC    % calculate radius of small circle
  \VDDdcircle N:\length{}   % draw circle
  \VDDlcirclepointletter PQ:M:1
  \VDDlcirclepointletter M:N:1
  \VDDdeflinepoint l:PQ
  \VDDlcirclelinepointletter l:NS:1:1
  \VDDlcirclelinepointletter l:NC:1:1
\VDDdcanvasend
```



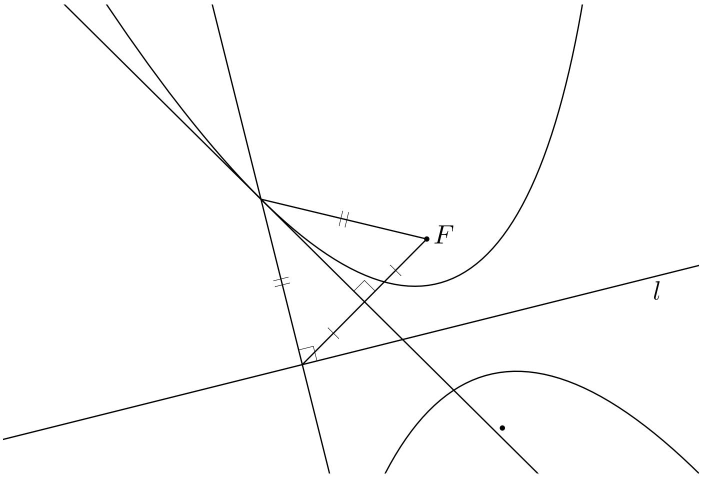
Find intersection of line and circle

```
\VDDdef M(0,0)L(-4,-4)R(4,4)
\VDDdcanvas1:1
   \VDDdcircle M:4               % some circle
   \VDDdeflinepoint l:RA(-5,1)
   \VDDdwholeline l              % some line
   \VDDcleadline m:l:M
   \VDDdwholeline m
   \VDDcintersect S:lm
   \VDDtriangleonesideotherangle P:SM:4,90 % find P
   \VDDctranslate Q:S:PS         % the other point is related
   \VDDcavg M':MQ
   \VDDllinelinepointletter lm:SM':1:1
   \VDDdeflinepoint n:QM
   \VDDllinelinepointletter mn:MP:1:1
   \VDDdperp QSM
   \VDDdline PMQ
   \VDDdlinepartequal QMPM
   \VDDlcirclelinepointletter l:MP:1:1
   \VDDlcirclelinepointletter l:MQ:1:1
\VDDdcanvasend
```



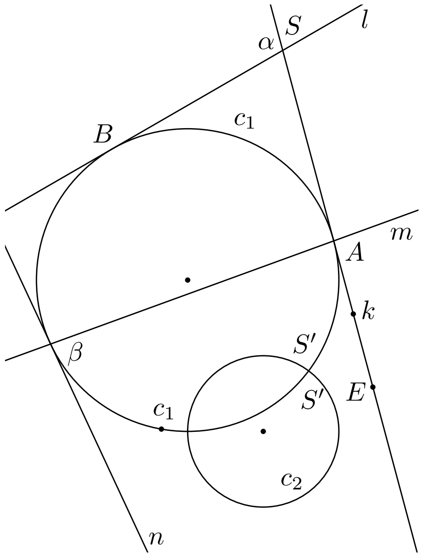
Parabola with focus and directrix.

```
\VDDdef L(-4,-1)R(4,4)F(1,1.5)F'(2,-1)
\VDDdeflinepoint l:O(0,0)O'(4,1) % l = directrix
\VDDcleadline p:l:F
\VDDcintersect V:pl
\VDDcabstransp Q:V:O'O:2         % Q = point on directrix
\VDDdeflinepoint m:QF            % m = line through Q and F
\VDDcavg M:QF
\VDDcleadline n:m:M              % n perpendicular bisector of QF
\VDDcleadline k:l:Q              % k perpendicular to directrix in Q
\VDDcintersect P:kn              % P point on parabola
\VDDdcanvas1:1
  \VDDdef O''(8,2)
  \VDDdpoint FF'
  \VDDdperp OQP
  \VDDdperp FMP
  \VDDdwholeline lkn
  \VDDdlinepartequal FMMQ
  \VDDdlinepartequal FPPQ
  \VDDdline PFQ
  \VDDlpointany OO'O'':$l$
  \VDDlpointletter QFP
  \VDDdparabola F:l              % draw parabola determined by F and l
  \VDDdparabola F':l              % draw parabola determined by F' and l
\VDDdcanvasend
```



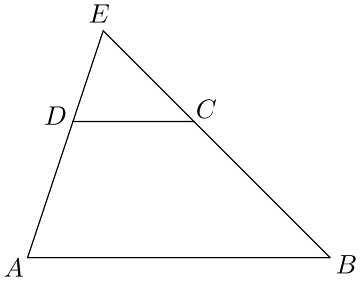
Convenience functions to find tangents and place labels at lines, circles and intersections of those.

```
\VDDdef M(0,0)A(2;15)B(2;120)C(2;205)F(2;260)N(1,-2)N'(0,-3)
\VDDccircletangent kln:ABC:M     % find tangents k,l,n of circle in A,B,C
\VDDcintersect S:kl
\VDDdeflinepoint m:AC
\VDDclinepar D:A:k:M:-1
\VDDclinepar E:A:k:M:-2
\VDDtriangletwolengths S':MN:2,1 % find intersection of the two circles
\VDDdcanvas1:1
  \VDDdwholeline lmkn \VDDdcircle M:2 \VDDdcircle N:1 \VDDdpoint DEMFN
  \VDDllinepointany k:D:M:1:$k$    % label D wrt line k opposite to M with $k$
  \VDDllinepointletter k:E:M:-1    % label E wrt line k at M's side
  \VDDlcirclepointletter B:M:1     % label B with center M on outside
  \VDDlcirclecirclepointany S':MN:-1:-1:$S'$ % label intersection of two circles
                                             % use the first center (M) as reference
  \VDDlcirclecirclepointletter S':MN:1:1 % label intersection of two circles
                                             % use the first center (M) as reference
  \VDDlcirclepointany F:M:-1:$c_1$ % label F with center M on inside with $c_1$
  \VDDllinelinepointletter kl:SM:1:1        % label S wrt k on opposite side of M
                                            % and wrt l on opposite side of M
  \VDDllinelinepointany kl:SM:-1:1:$\alpha$ % label S wrt k on side of M and wrt l
                                            % on opposite side of M with $\alpha$
  \VDDlcirclelinepointletter m:MA:1:1       % label A wrt m on opposite side of M
                                            % and outside of center M
  \VDDlcirclelinepointany m:MC:-1:1:$\beta$ % label C wrt m on the opposite side of
                                            % M and inside of center M with $\beta$
  \VDDllineedgeletter m:M:1:1      % label line m at positive outside wrt M
  \VDDllineedgeletter l:M:-1:-1    % label line l at negative inside wrt M
  \VDDllineedgeletter k:M:-1:1     % label line k at negative outside wrt M
  \VDDllineedgeletter n:M:1:-1     % label line k at positive inside wrt M
  \VDDlcircleangleany M:2:70:1:$c_1$ % label circle M:2 at 70 (wrt xhat) outside
  \VDDlcircleangleany N:1:-60:-1:$c_2$ % label circle N:1 at -60 (wrt xhat) inside
\VDDdcanvasend
```



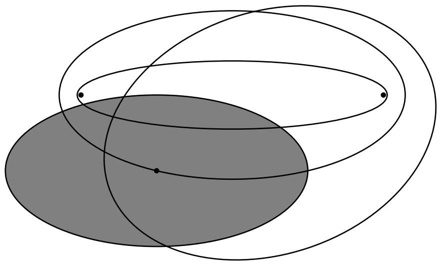
Multiplication wrt point.

```
\catcode`@=11%
\VDDdef A(0,0)B(4,0)E(1,3)
\VDDcpointmul DC:AB:E:0.4 % D <- A, C <- B multiplication wrt E factor 0.4
\VDDdcanvas1:1
  \VDDdline CD
  \VDDdlpolygon ABCED
\VDDdcanvasend
```



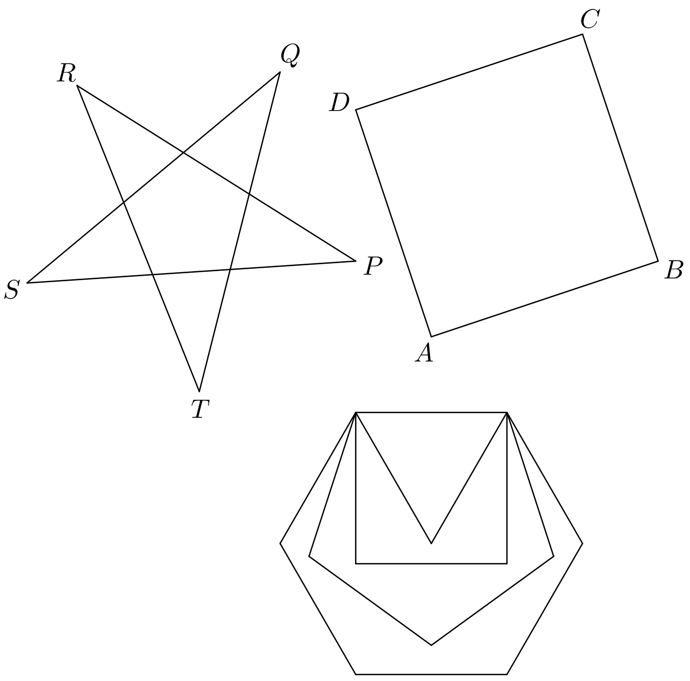
Ellipses

```
\catcode`@=11%
\VDDdef A(0,0)B(3,1)X(-2,-2)Y(6,2)C(-1,1)
\VDDdcanvas1:1
  \VDDdfillellipseradii{fillcolor=gray,fillstyle=solid}A:2,1
  \VDDdellipseradii A:2,1
  \VDDdellipspoint BC:A
  \VDDdellips AB:4.5
  \VDDdellips BC:4.1
  \VDDdpoint ABC
\VDDdcanvasend
```



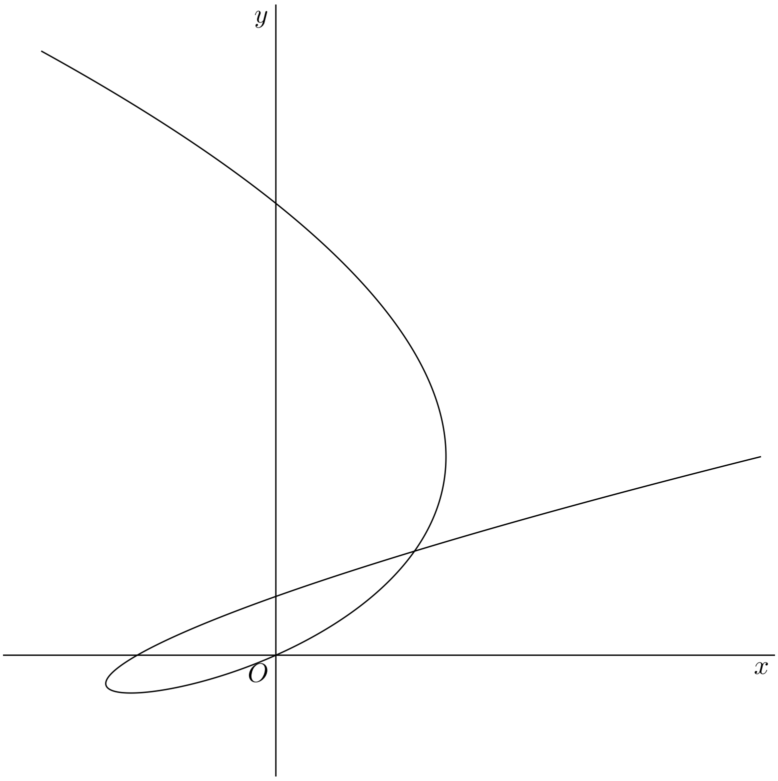
Regular polygons

```
\catcode`@=11%
\VDDdef A(0,0)B(3,1)
\VDDdef P(-1,1)Q(-2,3.5)
\VDDcsquare CD:AB
\VDDcpentagon RST:PQ
\VDDdef K(1,-1)L(-1,-1)
\VDDcregulartriangle M:KL
\VDDcsquare M'N:KL
\VDDcpentagon M''N'O:KL
\VDDchexagon M'''N''O'U:KL
\VDDdcanvas1:1
  \VDDdlpolygon ABCD
  \VDDlpolygon PQRST
  \VDDdpolygon PRTQS
  \VDDdpolygon KLM
  \VDDdline LM'NK
  \VDDdline LM''N'OK
  \VDDdline LM'''N''O'UK
\VDDdcanvasend
```



Making parametric plots etc. Functions are written in postscript notation (the operator comes after the operands). So "x x *" means "x * x" and "2 5 x * +" means "5 * x + 2".

```
\catcode`@=11%
\VDDdef O(0,0)X(1,0)Y(0,1) % OX and OY are unit vectors
\VDDdef W(-3,-1)W'(6,8) % W and W' determine the window
\VDDdeflinepoint x:OX % x-axis
\VDDdeflinepoint y:OY % y-axis
\VDDdcanvas1:1
  \VDDdwholeline xy
  \VDDllinelinepointletter xy:OW':1:1 % label O wrt W'
  \VDDllineedgeletter x:W':1:1 % label x axis wrt W'
  \VDDllineedgeletter y:W':-1:1 % label y axis wrt W'
  \parametricplot[plotpoints=400]{-3.12}{3.5}{%
    % formula for x = 1/3 t^3 - 2.25 t
    t t t 1 3 div mul mul mul -2.25 t mul add
    % formula for y = (t^2 - 2t) / 2
    t t mul -2 t mul add 2 div
  }%
\VDDdcanvasend
```



Plotting functions and areas.

```
\catcode`@=11%
\VDDdef O(0,0)X(1,0)Y(0,1) % OX and OY are unit vectors
\VDDdef W(-3,-1)W'(6,8) % W and W' determine the window
\VDDdeflinepoint x:OX % x-axis
\VDDdeflinepoint y:OY % y-axis
\VDDdcanvas1:1
  \VDDdwholeline xy
  \VDDllinelinepointletter xy:OW':1:1 % label O wrt W'
  \VDDllineedgeletter x:W':1:1 % label x axis wrt W'
  \VDDllineedgeletter y:W':-1:1 % label y axis wrt W'
  \parametricplot[plotpoints=400]{-3.12}{3.5}{%
    % formula for x = 1/3 t^3 - 2.25 t
    t t t 1 3 div mul mul mul -2.25 t mul add
    % formula for y = (t^2 - 2t) / 2
    t t mul -2 t mul add 2 div
  }%
\VDDdcanvasend
```