

```
> restart; Digits:=30; with(geometry); _EnvHorizontalName := x;
  _EnvVerticalName := y;
```

*Digits := 30*

*[Apollonius, AreCollinear, AreConcurrent, AreConcyclic, AreConjugate, AreHarmonic, AreOrthogonal, AreParallel, ArePerpendicular, AreSimilar, AreTangent, CircleOfSimilitude, CrossProduct, CrossRatio, DefinedAs, Equation, EulerCircle, EulerLine, ExteriorAngle, ExternalBisector, FindAngle, GergonnePoint, GlideReflection, HorizontalCoord, HorizontalName, InteriorAngle, IsEquilateral, IsOnCircle, IsOnLine, IsRightTriangle, MajorAxis, MakeSquare, MinorAxis, NagelPoint, OnSegment, ParallelLine, PedalTriangle, PerpenBisector, PerpendicularLine, Polar, Pole, RadicalAxis, RadicalCenter, RegularPolygon, RegularStarPolygon, SensedMagnitude, SimsonLine, SpiralRotation, StretchReflection, StretchRotation, TangentLine, VerticalCoord, VerticalName, altitude, apothem, area, asymptotes, bisector, center, centroid, circle, circumcircle, conic, convexhull, coordinates, detail, diagonal, diameter, dilatation, directrix, distance, draw, dsegment, ellipse, excircle, expansion, foci, focus, form, homology, homothety, hyperbola, incircle, inradius, intersection, inversion, line, medial, median, method, midpoint, orthocenter, parabola, perimeter, point, powerpc, projection, radius, randpoint, reciprocation, reflection, rotation, segment, sides, similitude, slope, square, stretch, tangentpc, translation, triangle, vertex, vertices]*

*\_EnvHorizontalName := x*

*\_EnvVerticalName := y*

```
>
> tanto_alg3:=proc(p1,p2,p3,d)
>   local k, sq, a, aa, b, c, f, g, p, s, i, c_a, c_b, c_z, c_q, t,
  para, para2, para3, solx, solsq, cero, EQ, p4, y1, y2, solYY,
  solVP, solY, res;
>   k := p1 * p2;
>   sq := evalf(sqrt(k));
>   p := (k/p3)+p3;
>   p4:= evalf(k/p3);
>
>   aa := evalf( k / ( k - sq ) );
>   c_a:= aa * sq;
>
>   print( p1,p2,k,sq,trunc(sq*2));
>
>   a := sqrt(k) / (p*sqrt(k)-(2*k));
>   b := 1-(p*sqrt(k)) / (p*sqrt(k)-(2*k));
>   c := (k*sqrt(k)) / (p*sqrt(k)-(2*k));
>   s := (sq-p3) / (sq-((sq*sqrt((-2*k*k)+p*k*sq))/k)-p3);
>
>   f := a*x^2+b*x+c=sqrt(k);
>   g := a*x^2+b*x+c=x;
>   para := a*x^2+b*x+c=y;
```

```

> para2:= -a*x^2-b*x+c=y;
> para3:= a*x^2+b*x-p3*c=y;
>
> parabola(P,para,[x,y]);
> parabola(PP,para2,[x,y]);
> parabola(PPP,para3,[x,y]);
>
> print( evalf(coordinates(focus(P))) );
>
> point(V1,sq,k);
> point(V2,sq,evalf(k-coordinates(focus(P))[2]));
> parabola(VP,['focus'=V2,'vertex'=V1]);
>
> solx := solve(g,x);
> solsq := solve(f,x);
> cero := fsolve({a*x^2+b*x+c=y,x=0});
>
> point( P1, [solx[2],solx[2]] );
> point( P2, [solsq[2],sq] );
>
> if( rhs(cero[1]) > 0) then
>   line(Top,y=rhs(cero[1])*2,[x,y]);
> else
>   line(Top,y=rhs(cero[2])*2,[x,y]);
> end if;
>
> line(Y,y=x,[x,y]);
> line(l0,y=0,[x,y]);
> line(l1,x=0,[x,y]);
> line(sqv,x=sq,[x,y]);
> line(sqh,y=sq,[x,y]);
>
> point(P3,sq*p3,sq);
> point(P4,0,(k/(sq-p3)));
> point(K,k,0);
> point(OK,0,(k/(sqrt(k)-p3))*p3);
>
> # Begin from tanto_slope_6
>
> point(_K,-k,0);
> point(KK,k,k);
> point(_KK,-k,k);
> point(K_K,k,-k);
> point(_K_K,-k,-k);
> point(SQ,sq,sq);
> point(SQ0,sq,0);
>
> line(A,[K,SQ]);
> line(B,[KK,SQ0]);

```

```

>
> intersection(A3,A,B);
>
> c_b := frac(k/aa) + d;
>
> i := evalf(k/c_b);
> line(C,[point(PC,0,i),A3]);
>
> # End from tanto_slope_6
> # Begin from algo_test_8
>
> c_z := (c_a*k)/(k-(p3-1)*c_a);
> c_q := -(c_z - sq);
> t := (((c_q*k)-(c_a*k))/(c_a*c_q));
> c_q := (c_a*k)/(k-t*c_a);
>
> point(PP1,[t,c_z]):
> point(PP2,[ (p3-1), c_q]):
> line(lin1,[PP1,PP2]);
>
> point(PP1,[ (p3-1), c_z]):
> point(PP2,[t,c_q]):
> line(lin2,[PP1,PP2]);
>
> # End from algo_test_8
>
> # Building y^-1
> y1 := fsolve({Equation(PP),x=p3},{x,y},{x=-k..k});
> y2 := fsolve({Equation(PP),x=p4},{x,y},{x=-k..k});
>
> print( y1 );
> print( y2 );
>
> if( rhs(y1[1]) > p3 ) then
>   point(PY1, rhs(y1[2]), rhs(y1[1]));
> else
>   point(PY1, rhs(y1[1]), rhs(y1[2]));
> end if;
> if( rhs(y2[1]) > p4 ) then
>   point(PY2, rhs(y2[2]), rhs(y2[1]));
> else
>   point(PY2, rhs(y2[1]), rhs(y2[2]));
> end if;
>
> line(YY,[PY1,PY2]);
>
> print( "print block for YY");
> print( fsolve({Equation(YY),y=0}) );
> # print( fsolve({Equation(YY),Equation(Red)}) );

```

```

> # print( fsolve({Equation(YY),y=k}) );
> # print( solve({Equation(P),Equation(YY)})) );
>
> # From tanto_slope_6
> print( "from tanto_slope_6" );
> print( "i", i );
>
> print("From algo_test_8");
> print("slope A ", slope(lin1));
> print("interseccion R-X ",
coordinates(intersection(Q4,lin2,l0)));
> print("interseccion R-Y ",
coordinates(intersection(Q5,lin2,l1)));
> print("interseccion A-X ",
coordinates(intersection(Q6,lin1,l0)));
> print("interseccion A-Y ",
coordinates(intersection(Q7,lin1,l1)));
> print("interseccion A-R ",
coordinates(intersection(Q8,lin1,lin2)));
>
> # line(NN,x*VerticalCoord(Q7)+VerticalCoord(Q5)=y,[x,y]);
>
> res:=sq/((frac(HorizontalCoord(Q4))+d)*2);
> print( "res", res );
> print(solve({Equation(C),y=0}));
> print(solve({Equation(C),x=0}));
> print(solve({Equation(C),Equation(YY)}));
> # print("NN-X", solve({Equation(NN),y=0}) );
>
>
draw([Y,sqv,sqh,YY(colour=green),C(colour=green),PP(colour=green),
P(colour=green),lin1(colour=blue),lin2(colour=red)],axes=normal,co
lour=black,view=[0..k/2,0..k/2]);
>
> end proc:
[
>
> tanto_alg3(3,11,3,1);
3, 11, 33, 5.74456264653802865985061146822, 11
[5.74456264653802865985061146823, 0.627718676730985670074694265200]
{x=3.00000000000000000000000000000000,y=23.2856604584846841109418861532}
{x=11.00000000000000000000000000000000,y=15.2856604584846841109418861531}
"print block for YY"
{x=26.2856604584846841109418861528, y=-0.}
"from tanto_slope_6"
"i", 26.2856604584846841109418861528
"From algo_test_8"

```

"slope A ", 2.28787307893019376934232125868

"interseccion R-X ", [7.25543735346197134014938853176, 0.]

"interseccion R-Y ", [0., 16.5995197968501769459792332861]

"interseccion A-X ", [4.74456264653802865985061146821, 0.]

"interseccion A-Y ", [0., -10.8549571503121482861286218179]

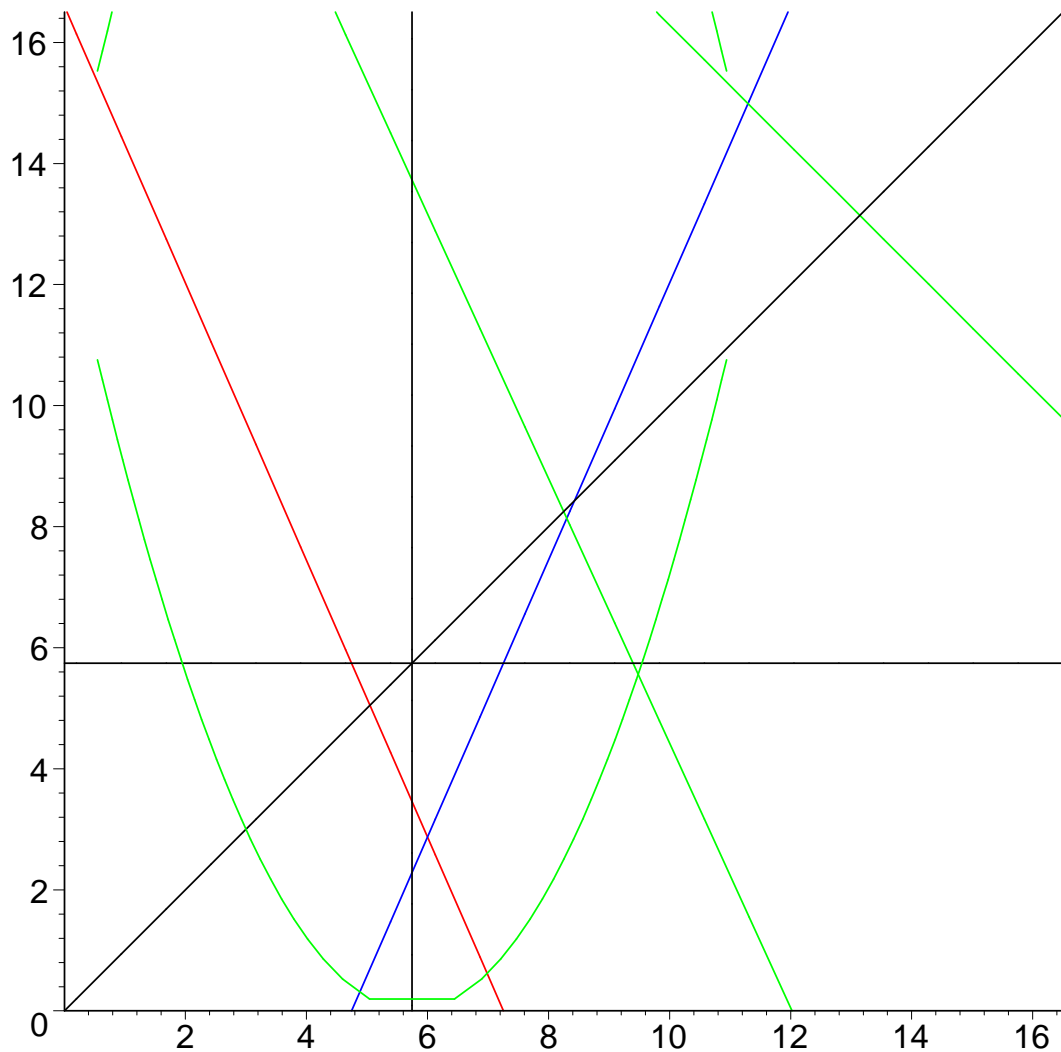
"interseccion A-R ", [6., 2.87228132326901432992530573410]

"res", 2.28787307893019376934232125871

{y = 0., x = 12.0237736389897894072945907688 }

{y = 26.2856604584846841109418861528, x = 0. }

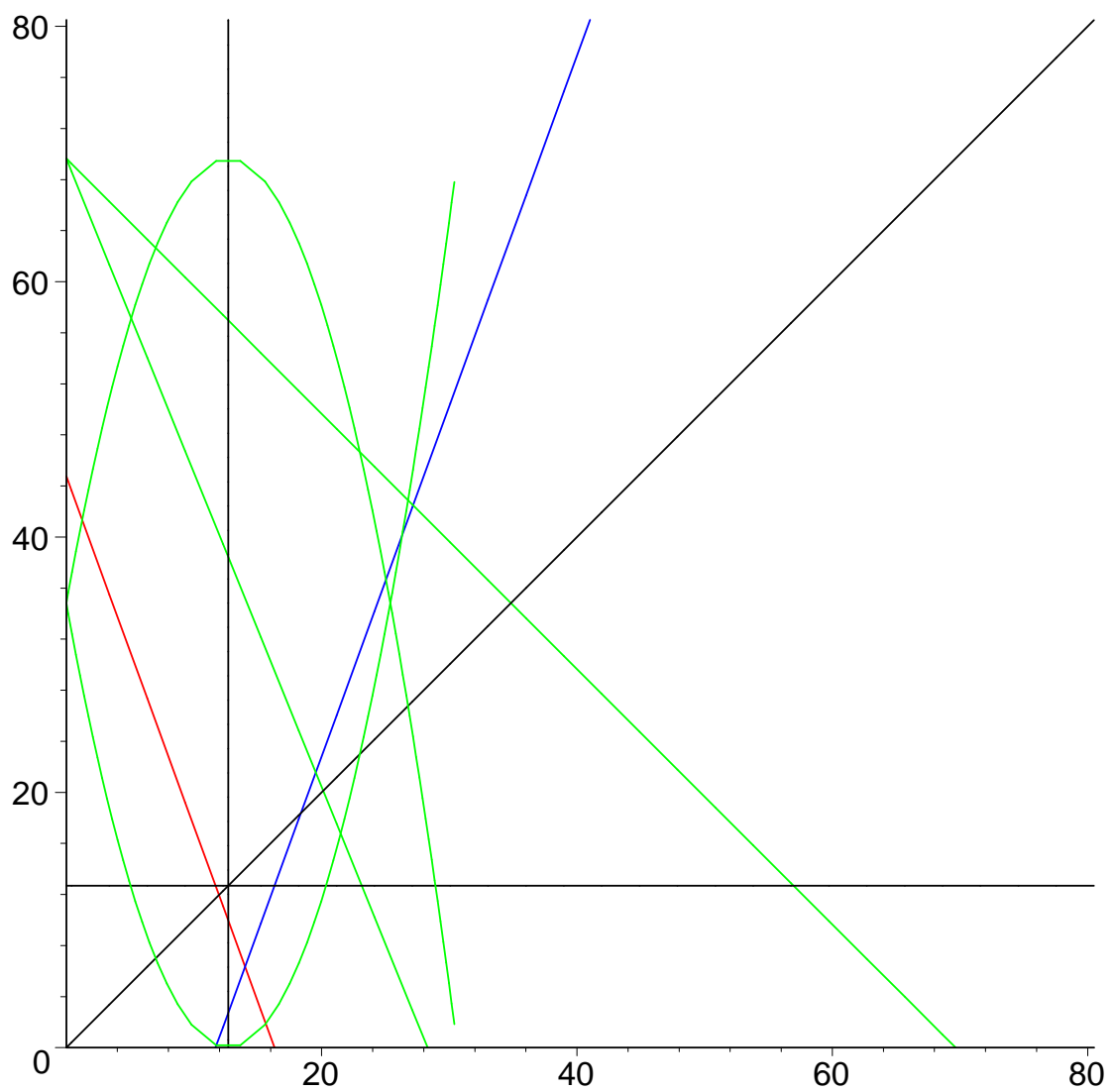
{y = 26.2856604584846841109418861537, x = -0.415989708371089915570877713753  $10^{-27}$ }



```

> tanto_alg3(7,23,7,2);
      7, 23, 161, 12.6885775404495203801937727461, 25
      [12.6885775404495203801937727457, 1.15571122977523980990311363702]
      {x=7.000000000000000000000000000000, y=62.6540778751933247064249595642}
      {x=23.000000000000000000000000000000, y=46.6540778751933247064249595642}
      "print block for YY"
      {x=69.6540778751933247064249595644, y=-0.}
      "from tanto_slope_6"
      "i", 69.6540778751933247064249595617
      "From algo_test_8"
      "slope A ", 2.74475518052142816955395774362
      "interseccion R-X ", [16.3114224595504796198062272541, 0.]
      "interseccion R-Y ", [0., 44.7708612975247545638522947840]
      "interseccion A-X ", [11.6885775404495203801937727462, 0.]
      "interseccion A-Y ", [0., -32.0822837570752341836585220380]
"interseccion A-R ",
      [14.000000000000000000000000000001, 6.34428877022476019009688637299]
      "res", 2.74475518052142816955395774346
      {x=28.3023302143961855465285483228, y=0.}
      {x=0., y=69.6540778751933247064249595617}
      {y=69.6540778751933247064249595662, x=-0.183028976120622824292919160420 10-26}

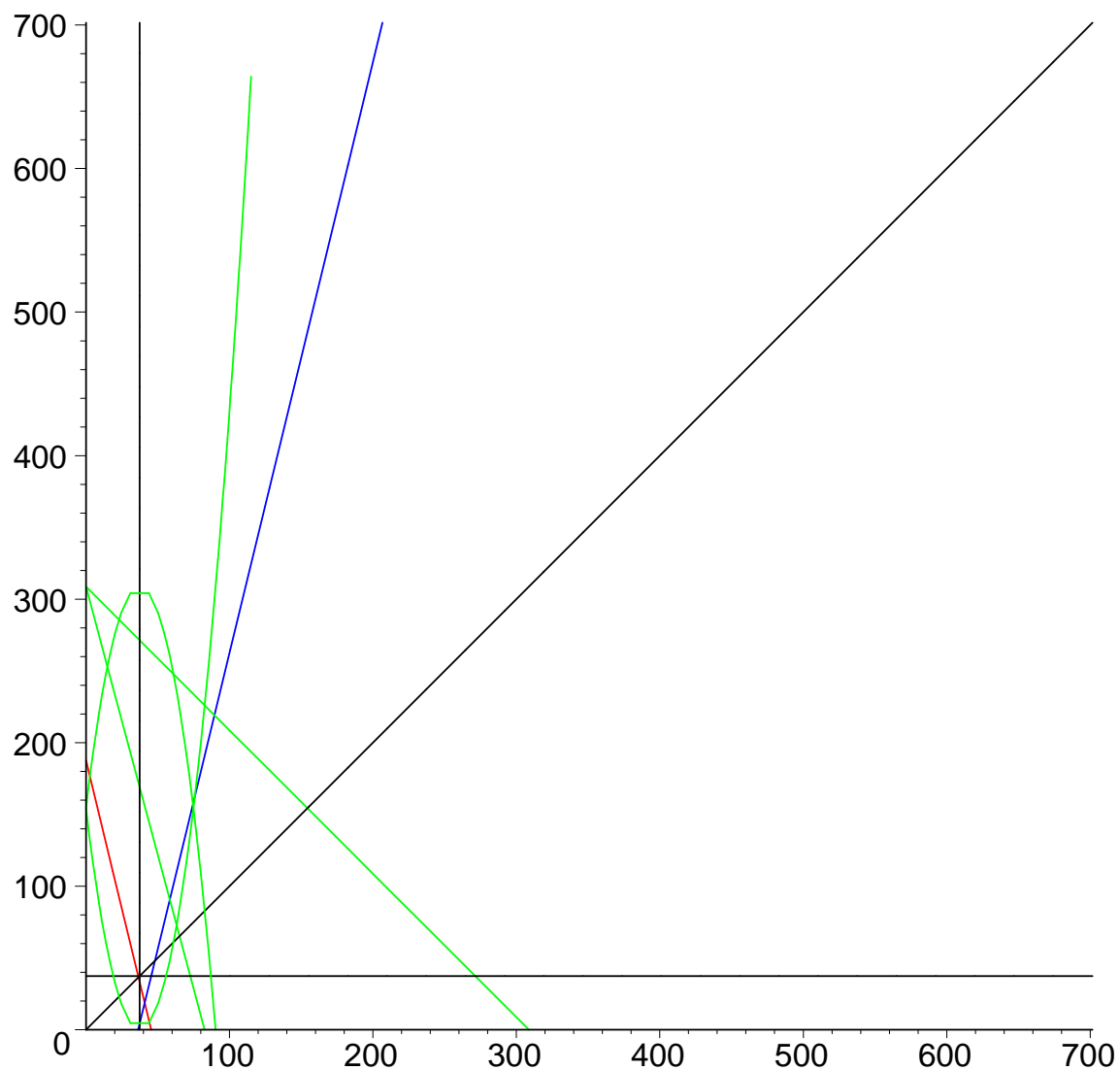
```



```
> tanto_alg3(23,61,23,4);
      23, 61, 1403, 37.4566416006560844472448586154, 74
      [37.4566416006560844472448586143, 2.27167919967195777637757068349]
      {x=23.0000000000000000000000000000, y=285.802404891192483322671846644}
      {x=61.0000000000000000000000000000, y=247.802404891192483322671846644}
      "print block for YY"
      {x=308.802404891192483322671846645, y=-0.}
      "from tanto_slope_6"
      "i", 308.802404891192483322671846955
```

"From algo\_test\_8"  
"slope A ", 4.12213150585534009249900839606  
"interseccion R-X ", [45.5433583993439155527551413840, 0.]  
"interseccion R-Y ", [0., 187.735712540396986016081773544]  
"interseccion A-X ", [36.4566416006560844472448586152, 0.]  
"interseccion A-Y ", [0., -150.279070939740901568836914929]  
"interseccion A-R ",  
[40.999999999999999999999999999997, 18.7283208003280422236224293075]  
"res", 4.12213150585534009249900839647  
{ $x = 82.7404407689854861149677299679$ ,  $y = 0.$  }  
{ $x = 0.$ ,  $y = 308.802404891192483322671846955$  }  
{ $x = 0.113540525672287058902475608928 \cdot 10^{-24}$ ,  $y = 308.802404891192483322671846531$  }





```
> tanto_alg3(59,127,59,6);
      59, 127, 7493, 86.5621164251429333084549727151, 173
      [86.5621164251429333084549727160, 3.21894178742853334577251351322]
      {y = 1104.89181520207266373724317523, x = 59.0000000000000000000000000000}
      {x = 127.0000000000000000000000000000, y = 1036.89181520207266373724317523}
      "print block for YY"
      {x = 1163.89181520207266373724317523, y = -0.}
      "from tanto_slope_6"
      "i", 1163.89181520207266373724317791
```

"From algo\_test\_8"

"slope A ", 6.72287060014632041422418359126

"interseccion R-X ", [98.4378835748570666915450272841, 0.]

"interseccion R-Y ", [0., 661.785153426032944762852376749]

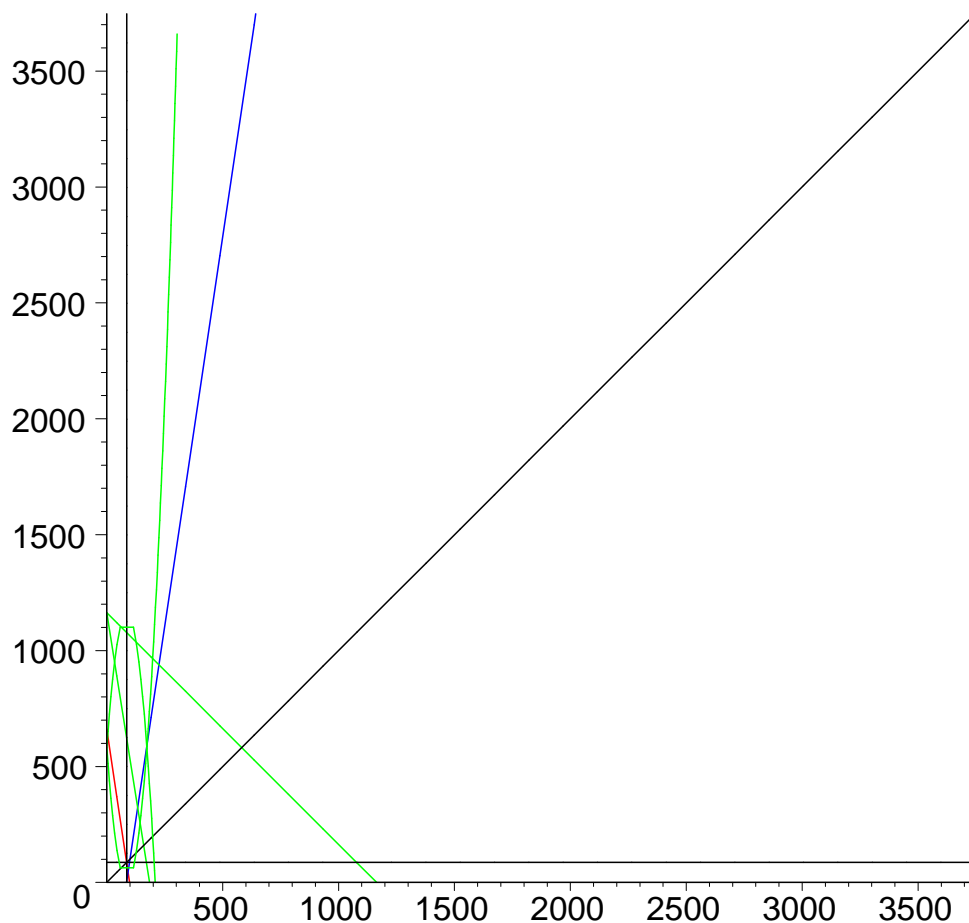
"interseccion A-X ", [85.5621164251429333084549727152, 0.]

"interseccion A-Y ", [0., -575.223037000890011454397404038]

"interseccion A-R ",

[illegible]

```
"res", 6.72287060014632041422418359188
```

$$\{x = 184.729019597098009908918190509, y = 0.\}$$
$$\{x=0., y=1163.89181520207266373724317791\}$$
$$\{y = 1163.89181520207266373724317472, x = 0.505734804239948456106316232086 \cdot 10^{-24}\}$$


```
> tanto_alg3(1237,9743,1237,2018);
```

1237, 9743, 12052091, 3471.61216151804607080526185951, 6943



```
[ > #Digits:=200;tanto_alg3(p1,p2,p1,p3);  
[ > #sq:=1762787066122860943811705736776226385093299156422107330123170  
327876780025762012912130622946534606.21986183297712292148540810932  
050491402973780493054318632555472963195545494700626908418852823987  
93458885:div:=175.737518909765632285740130530324088377149251584216  
834771662694705173882029859106595597097379251238875372374312275020  
337408876457754090613982262008110324599362241324639027211340738701  
73684332511856169:sq/div/2;  
[ >
```