



Maxima 5.27.0 <http://maxima.sourceforge.net>  
 using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (a.k.a. GCL)  
 Distributed under the GNU Public License. See the file COPYING.  
 Dedicated to the memory of William Schelter.

## Jewish Problem 7 (Tania Khovanova)

---

Pour l'origine, les détails, et la compilation de ces *killer problems*, voir :

[Article sur le site arXiv.org \(Cornell University Library\)](#)

```
▷ load("../gdd.mac")$
```

**Problème 7** — On considère un cercle dont on a représenté un diamètre. Par un point quelconque, hors du cercle et de son diamètre, tracer la perpendiculaire au diamètre passant par ce point en utilisant seulement une règle.

```
▷ A:Origine;
5:   point(0,0)

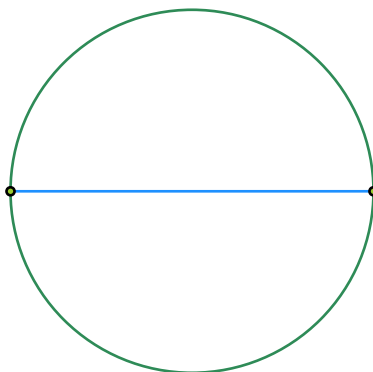
▷ B:Point(4,0);
6:   point(4,0)

▷ C:Point(3,4);
7:   point(3,4)

▷ c:CercleDiametre(A,B);
8:   cercle (point(2,0),2)

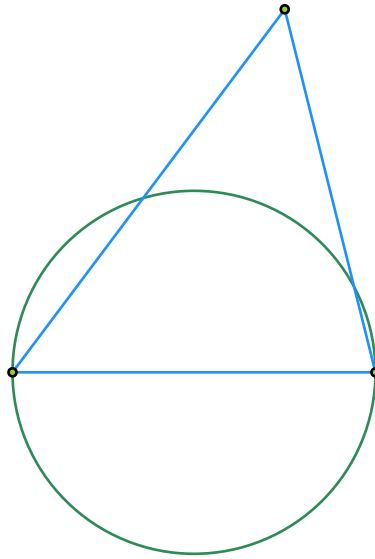
▷ Figure(A,B,C,'c,Segment(A,B));
```

•



On construit les segments joignant les extrémités du diamètre au point considéré.

▷ `Figure(Segment(A,C),Segment(B,C));`



Marquons les points d'intersection de ces segments avec le cercle.

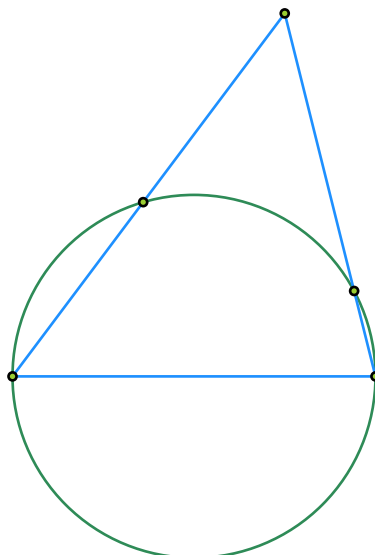
▷ `I:(It:Intersection(Droite(A,C),c),if is(It[1]=A) then It[2] else It[1]);`

11: `point`  $\left(\frac{36}{25}, \frac{48}{25}\right)$

▷ `J:(Jt:Intersection(Droite(B,C),c),if is(Jt[1]=B) then Jt[2] else Jt[1]);`

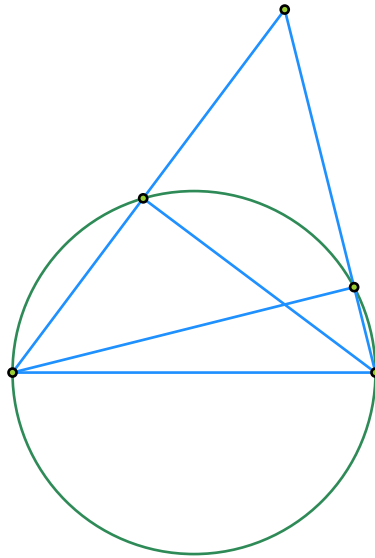
12: `point`  $\left(\frac{64}{17}, \frac{16}{17}\right)$

▷ `Figure(I,J);`



Construisons maintenant les segments  $AJ$  et  $BI$ .

▷ `Figure(Segment(A,J),Segment(B,I));`

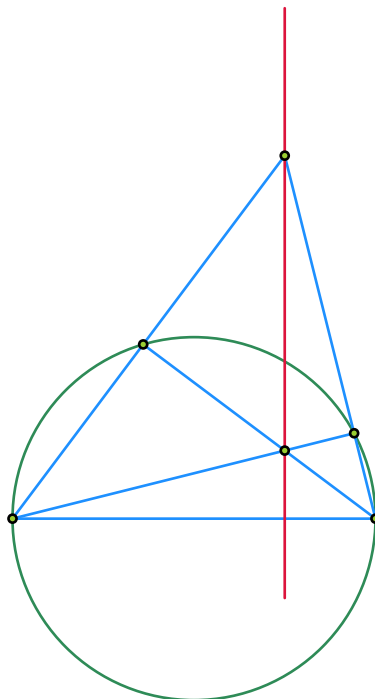


Les segments que nous venons de tracer sont des hauteurs du triangle  $ABC$ . La troisième va nous donner la droite recherchée.

```

> H:Intersection(Droite(A,J),Droite(B,I));
15:   point  $\left(3, \frac{3}{4}\right)$ 
> d:Droite(C,H);
16:   droite  $\left(\text{point}(3,4), \text{point}\left(3, \frac{3}{4}\right)\right)$ 
> Figure(H, 'd');

```

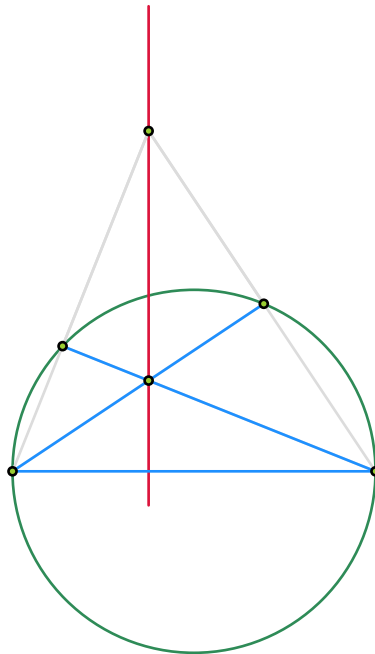


Faisons de tout cela une macro avec points initiaux  $A$ ,  $B$  et  $C$ .

```

> k7(A,B,C) := block([c,d,I,J,H,It,Jt,s,t],
  c:CercleDiametre(A,B),
  It:Intersection(Droite(A,C),c),
    if SontEgaux(It[1],A) then I:It[2] else I:It[1],
  Jt:Intersection(Droite(B,C),c),
    if SontEgaux(Jt[1],B) then J:Jt[2] else J:Jt[1],
  H:Intersection(Droite(A,J),Droite(B,I)),
  d:Droite(C,H),
  s:[Segment(A,B),Segment(A,C),Segment(B,C),
    Segment(C,I),Segment(C,J)],
  t:[Segment(A,H),Segment(B,H),Segment(I,H),
    Segment(J,H),Segment(A,I),Segment(B,J)],
  Figure(A,B,C,I,J,H,'c','d','t',s)
);
> (NouvelleFigure(figb), k7(A,B,Point(3/2,1)));

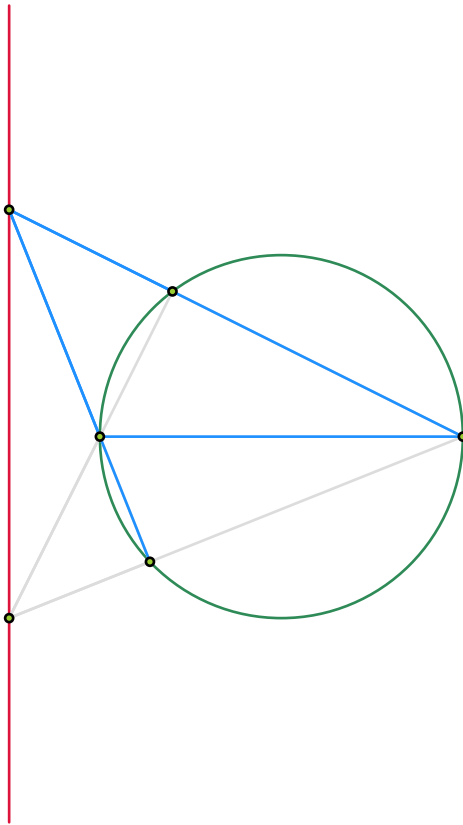
```



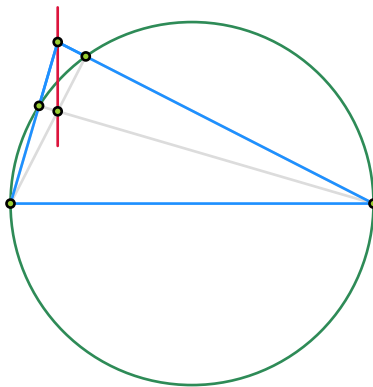
```

> (NouvelleFigure(figc), k7(A,B,Point(-1,5/2)));

```



▷ (NouvelleFigure(figd), k7(A,B,Point(0.52,1.78)));



**TODO :** (9 mai 2013) — Affiner le tracé des droites.