Useful procedures are posted on our lecture page, at
http://www.math.utah.edu/~korevaar/4530spring05/4530lectures.html

1) Rework exercise 1.4.7 using Maple; this was a messy hand calculation to compute the curvature, torsion, T,N,B for a specific curve which was not parameterized by arclength. In "curtor" are already written procedures to compute curvature and torsion. You can modify these to write procedures which will compute T, N, B. Note, that since it is easy to compute T, and also B, as the normalized cross product of the first and second derivatives of the curve, you can get N by taking B cross T.
   After all those computations, use Maple to draw a nice picture of the curve, say for t between zero and one. You can decide whether you prefer the "plot3d" or "tubeplot" commands. (See below for an example of what "tubeplot" creates.)

2) Exercise 1.7.8, for kap3 and kap4: Use the plane-curve recreation template which I've posted, or create your own procedure, to recreate nice pictures of these two curves.

3) Exercises 1.7.7 and 1.5.15: Again use the plane-curve recreation template which I've posted or your own procedure, to recreate a curve with curvature $\kappa(s) = \dfrac{1}{1+s^2}$, so that at s=0 the curve is horizontal and at the origin. Now, the real fun begins. Using the plane- Frenet system which we talked about in class (see also exercise 1.5.13 on page 41), you can find an explict formula for the curve by antidifferentiation (this is the exercise 1.5.15 part). Prove that the curve you've created is actually a catenary (graph of y=cosh(x) or Euclidean motion of such)! (We saw the catenary before as the shape of a hanging cable.)

4) Use the procedure "recreate3dview", which is one of the files posted on our web page, to create a 3-d curve which is beautiful, with your own choice of curvature and torsion functions.

5) Here is a picture of part of an a=2, b=1 helix, sitting on a radius 2 cylinder:

```
> with(plots):
> cylinder:=
  plot3d([2*cos(theta),2*sin(theta),z],theta=0..2*Pi,z=0..2*Pi,
  style=wireframe,color=black):
  #this is a parametric way of drawing the cylinder
> helix2:=tubeplot([2*cos(t),2*sin(t),t],
  t=0..2*Pi,radius=0.1,color=black):
  #this is a corresponding piece of the helix
> display({cylinder,helix2},scaling=constrained,axes=boxed);
  #here they are displayed together.
```
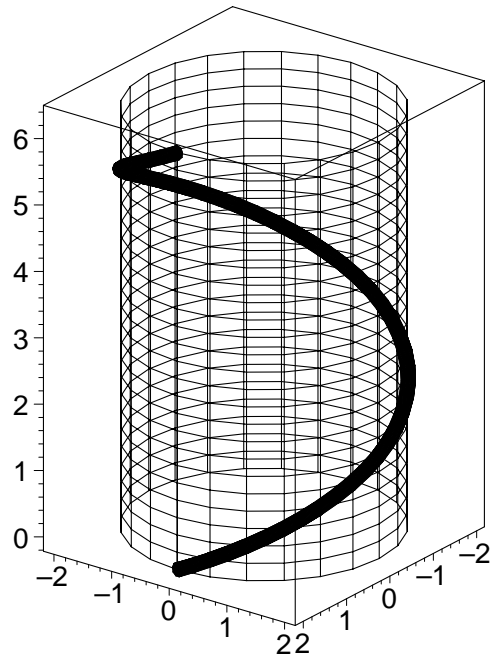
Your job is to find the osculating circle to the helix at the point when t = Pi , at $P = [-2, 0, \pi]$.
 (Recall exercise 1.3.28 from last week.) Add this osculating circle to the display above.