This homework illustrates the catastrophic effect of roundoff error, and also provides an introduction to Mathematica.

In 250 B.C.E., the Greek mathematician Archimedes estimated the number $\pi$ as follows. He looked at a circle with diameter 1, and hence circumference $\pi$. Inside the circle he inscribed a square. The perimeter of the square is smaller than the circumference of the circle, and so it is a lower bound for $\pi$. Archimedes then considered an inscribed octagon, 16-gon, etc., each time doubling the number of sides of the inscribed polygons, and producing ever better estimates for $\pi$. Using 96-sided inscribed and circumscribed polygons, he was able to show that $\frac{223}{71} < \pi < \frac{22}{7}$. There is a recursive formula for these estimates. Let $p_n$ be the perimeter of the inscribed polygon with $2^n$ sides. Then $p_2$ is the perimeter of the inscribed square, $p_2 = 2\sqrt{2}$. In general

$$p_{n+1} = 2^n \sqrt{2(1 - \sqrt{1 - (\frac{p_n}{2^n})^2})},$$

$$p_2 = 2\sqrt{2}. $$

Compute $p_n$ for $n = 3, 4, \ldots, 60$. Try to explain the failure in the formula. (This problem was suggested by Alan Cline.)

The formula derived above to estimate $\pi$ fails due to a combination of underflow and catastrophic cancellation. The formula can be improved so that the subtraction is eliminated. First write $p_{n+1}$ as

$$p_{n+1} = 2^n \sqrt{r_{n+1}},$$

where

$$r_{n+1} = 2(1 - \sqrt{1 - (\frac{p_n}{2^n})^2}), \quad r_3 = \frac{2}{2 + \sqrt{2}}.$$ 

Show that

$$r_{n+1} = \frac{r_n}{2 + \sqrt{4 - r_n}}.$$ 

Use the last iteration to calculate $r_n$ and $p_n$ for $n = 3, 4, \ldots, 60$. (This revision was suggested by W. Kahan.)

Eventually, $4 - r_n$ will round to 4, and so the latter formula is still affected by rounding errors for large values of $n$. Should this concern us?
Here are some notes on Mathematica (the material on ComputerArithmetic is not used for the above problem on $\pi$).

(* Illustration of Mathematica for simulating arbitrary precision arithmetic.*)
<<NumericalMath`ComputerArithmetic`
SetArithmetic[5,10,MixedMode->True] (* Sets 5-digit base 10 arithmetic.*)
x = {100.01, 102.01, .01}; y = {1.02, -1.00, .04};
(* Now convert each entry in these vectors to a computer number.*)
x5 = Map[ComputerNumber,x]; y5 = Map[ComputerNumber,y];
(* Here’s the exact result, using the Mathematica default 64-bit arithmetic.*)
x . y;
(* Here’s the result, using 5-digit decimal arithmetic, implied by the use of ComputerNumbers.*)
x5 . y5; (* The semicolon inhibits echoing the result.*)
Clear[x,y,x5,y5]

(*Recursion.*)
Nest[f,t,3]
NestList[f,t,3]
g[t_] := (t^2 + 2)/(2*t)
NestList[g,1.0,5] (*This computes Sqrt[2].*)
(* Hint for Homework # 1. Suppose you needed to compute a recursion of the form $p_{n+1} = f[p_n, n]$; this is a bit subtle, because the recursion depends not only on the previous value, but also a changing index.*)
FoldList[f, p2, Range[2,5]]
Last[FoldList[f, p2, Range[2,5]]]
FoldList[Plus,0,{a,b,c}] (*Just another example of FoldList.*)
(*To print out a specific number of digits, use NumberForm.*)
{N[Pi], NumberForm[N[Pi], 12]} (* N[] produces a numerical value.*)