Laboratory 5

1. #This is a well known function which always cycles to 4,2,1 within a
   # a reasonable number of iterations.
   f:= proc(n) if n=1 then RETURN; elif n mod 2 = 0 then f(n/2) else
   f(3*n+1); fi; end;
   find for n=2,..1000 which ones take the longest to stop. Also plot the iterations(counts) in the y-axis and the numbers in the x-axis. The file 3np contains the function. You call the function as f(10) etc.

2. *Demonstrating the principles of Chaos and Dynamic Systems using Maple*

   One phenomenon which occurs frequently in mathematics and the sciences is a function where the value at time t+1 is a function of the value at time t. For example, the population of deer in an area in a given year might be determined by a function based on the number of deer in the previous year. One of the features of many such functions is that they behave in a quite orderly and predictable fashion within some parameter ranges, but rather abruptly start behaving chaotically when the value of a parameter is changed slightly.

   Consider this function:

   $x_{t+1} = c * x_t * (1 - x_t)$

   Suppose that x is initialized to some suitable value, and then the function is run through, say, 50 iterations. It turns out that the performance of this function is very much dependent on the value of the constant c. The next homework investigates this.

   (a) Write a Maple procedure which accepts a single parameter, the constant c in the above formula. It should then iterates through the above function fifty times. On each iteration, it should display the iteration number and the current value of x. The initial value of x should be 0.75;

   Run this function a number of times, using the following parameters;

   2.5, 2.75, 3.0, 3.25, 3.5

   When your function is run with the first two values (2.5 or 2.75), it converges nicely to a single value. When it is run with the next two values (3.0 and 3.25), it oscillates between two values. If this were a simulation of the population of deer in an area, we could consider this to be boom and bust cycles. Note that they get further apart as the parameter increases. The next value (3.5) results in oscillations between four different points. At this point you should modify your function so that it performs 100 iterations, but only displays the results of the last 20 iterations. Run this revised function with the following parameters.

3.55, 3.6, 3.7

If you study the output, you will see that with the parameter 3.55, the function oscillates between eight values, with the parameter 3.6 it oscillates between 16 different values, and at 3.7 there is no discernible pattern to the output.

(b) Modify the function which you wrote for previous problem so that it displays the results graphically, with the iteration number on the x axis and the value of the function on the y axis. To do this, you should store the results in an array, and then convert the array to a list. This procedure will only work if you create the array of exactly the right size. Start by creating an array of size 100, for fifty points.

Then call the plot function, using the array as the first parameter. Experiment with the appearance of the plot (i.e. style=POINT; change the range of the vertical axis) to get a plot which is both pleasing and informative.