VIDEO VIEWING GUIDE

1. Approximately how many bears are there in Pennsylvania?

2. For what reasons do people dislike bears?

3. What is the primary means of controlling the bear population?

4. What tool is used by wildlife managers like Gary Alt to answer questions about bear population growth?

5. How many bears are removed from the population each year?

6. Does current removal hold the population steady?

7. What natural factor has the greatest effect on bear population growth?

8. What does a wildlife biologist do if the model doesn't give an accurate prediction?

USING TI-83 LISTS TO CHECK RECURSIVE FORMS

In Unit 4, *Prediction*, you learned how to define one list in terms of another list. Two other list operations will be very helpful as you examine functions proposed as models in this unit.

SEQ():

The first of these operations is the seq() command, found under the LIST OPS menu. It is extremely useful in creating a list of values for a particular closed-form function. (The difference between tables entered as lists instead of using the Table menu is that you may do calculations on the numbers in lists; you can't with Tables.)

The syntax for seq() looks like this:

seq(formula, variable, start, end, step)

For example, to create in L1 a list of values from 0 to 7, by ones, place the cursor on the header for L1 and enter seq(X,X,0,7,1). **Figure 1** illustrates the cursor placement and command.

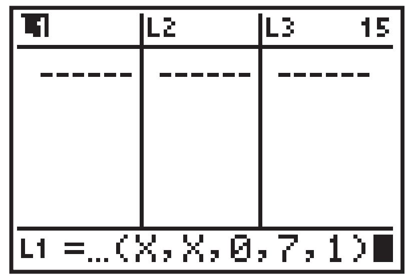

Figure 1.
Entering a seq() command for a list.

In this example, the formula was just X itself. To place in L2 a list of the squares of the integers from 0 to 7, place the cursor on the header of L2 and then enter the command seq(X^2,X,0,7,1). The result is shown in **Figure 2**.

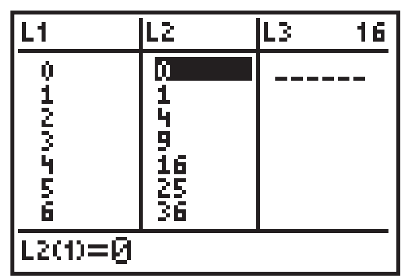

Figure 2.
Integers and their squares.

Of course, since L1 was already present, you also could have entered the command L1^2 in the header of L2 to accomplish the same task.

CHANGE:

The other operation that is extremely useful in checking data against recursive definitions is the ability to examine and graph the next value and the change between values.

Suppose list L2 contains your data values. To get the next values matched with current values, first define L3 to be exactly the same as L2; place the cursor on the L3 header and enter L2. Then place the cursor on the first entry of L3 and press DEL once to erase just that value. Since that makes L3 have one value fewer than L2, go to the bottom of L2 and press DEL to remove its last entry (which is still available as the last entry of L3). Now lists L2 and L3 are the current and next values of your data, as shown in **Figure 3.**

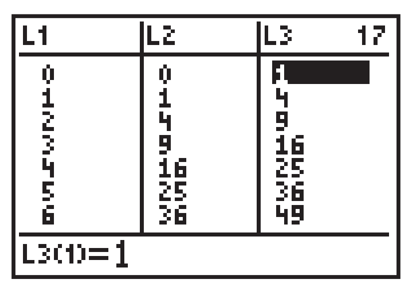

Figure 3.
Current and next values in lists L2 and L3.

You may obtain the values for the change between values by entering L3 – L2 in the header of L4, for example, as shown in **Figure 4.**

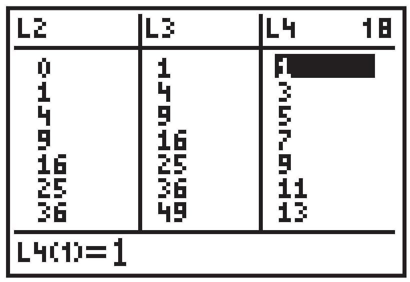

Figure 4.
Lists of current, next, and change values.

If you do not need the next values but do need the change values, the TI-83 has a special command just for that; you do not need to create L3. Suppose again that L2 contains your data values. To place the change values in L3, place the cursor on the L3 header and select ΔList (from the LIST OPS menu) and enter the command ΔList(L2). See **Figure 5**.

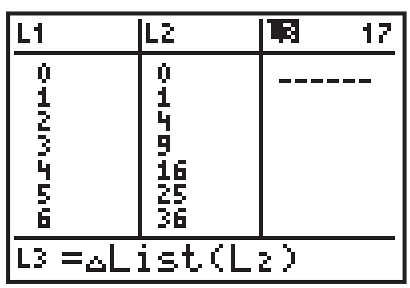

Figure 5.
Entering the ΔList command.

USING SEQ MODE TO DEFINE RECURSIVE RELATIONS ON THE TI-83

You are familiar with recursive definitions of functions, and you know how to enter closed-form equations into your calculator for graphing, making tables, and creating lists. The TI-83 also provides a way to use recursive definitions.

In order to have a complete recursive definition, you need a recursion equation and an initial value. In addition, you have to specify what "initial" means; does your counter start at 0 or 1 or somewhere else? It's easy to set these values. Here's how.

First, press MODE and select Seq (see **Figure 1**).

Next, press Y=. You now see an entirely different kind of screen.

Enter as the value of nMin the starting value for your counter. For most of the examples in this unit, that value has been 0. See **Figure 2**.

Next, enter the formula for the values. Just as the calculator always requires you to name your closed-form equations something like Y1 or Y2, it also requires you to name recursively defined functions u, v, or w. The parentheses indicate function notation, not multiplication. Since recursive function values are defined in terms of the prior values, the value of $u(n)$ must be defined in terms of the value before it, namely, $u(n - 1)$. Remember, $n - 1$ is the integer right before n, no matter what n happens to be. The u, v, and w keys are 2nd 7, 8, and 9, respectively. The n key is labeled"X,T,θ,n" and is next to the green ALPHA key.

Thus, to define the additive model $p_{next} = p_{current} + 17$, with $p(0) = 25$ as the initial value, the screen would look like that shown in **Figure 3**. Note, however, that the braces around the 25 in the screen do not have to be entered; the calculator will supply them if you just type the initial value.

This process completes the definition of the recursive model. However, in order to see it you need a table, a graph, or a list. Here are ways to see each of those.

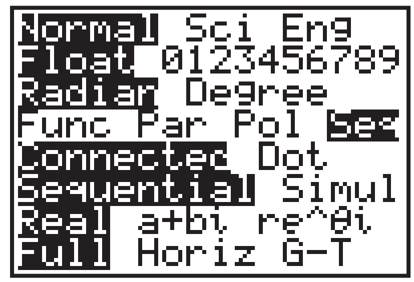

Figure 1.
Selecting Seq mode.

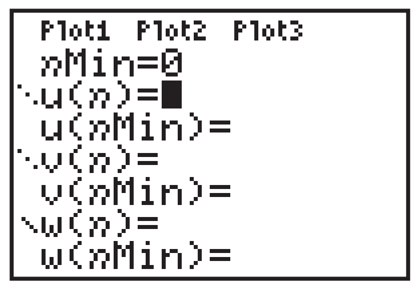

Figure 2.
The Y= screen in Seq mode.

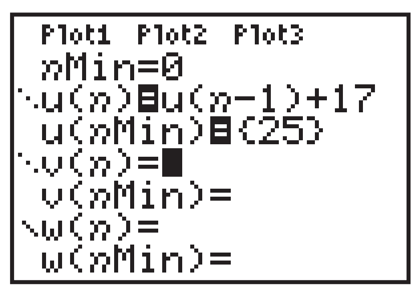

Figure 3.
Entering an additive model in Seq mode.

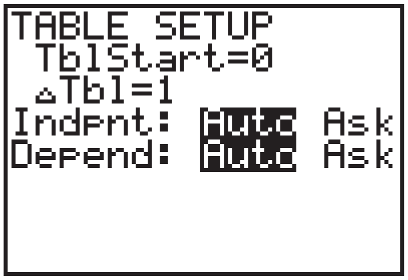

Figure 4.
Table setup menu.

TABLE:

Press the TBLSET key (2nd WINDOW). Select TblStart = to match your choice for *n*Min when you defined your function. Select ΔTbl = 1. Choose Auto for the remaining items. See **Figure 4.**

Press TABLE (2nd GRAPH).

GRAPH:

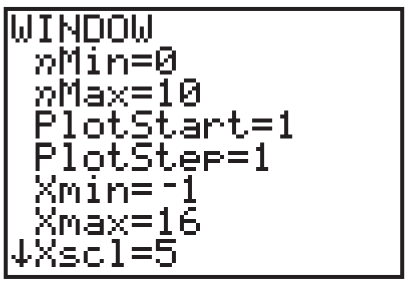

Figure 5.
Setting the window for a recursively defined function.

To see the graph you must define the window, just as with closed-form equations. However, with a recursively defined function, it is necessary to specify the size of the counter steps and its starting and ending values in addition to the actual edges of the window.

Remember, too, that it's good to make the window a little larger than what you plan to graph. Thus, for the previous example, a reasonable setting for the window size is [–1, 16] x [0, 200]. **Figure 5** shows other selections for that example.

Press GRAPH to see the result.

LISTS:

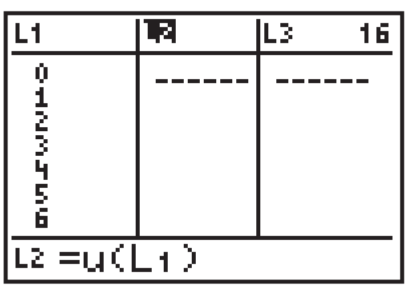

Figure 6.
Entering a recursively defined list.

To be able to take advantage of the ΔList command, you need to be able to get the values into a list. Handout H6.2 showed how to do this for closed-form equations. The main difference for recursively defined functions is that you must enter the counter values into a list before getting the function values.

First, use the seq() command (discussed in Handout H6.2) to create in L1 a list of the values you wish to use for the counter.

Next, place the cursor in the L2 header. Enter the formula *u*(L1). Remember, *u* is 2nd 7. See **Figure 6**.

INITIAL MOOSE POPULATION BY AGE AND SEX.

Age	Females	Males
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
Totals		

Female moose population by age and year.

Age	1993	1994	1995	1996	1997	1998
Females						
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
Total females						

H6.5
page 2 of 2

Male moose population by age and year.

Age	1993	1994	1995	1996	1997	1998
Males						
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
Total males						
Total males and females						

STEP-BY-STEP DIRECTIONS FOR A PROBABILISTIC SIMULATION

Be sure your calculator is in floating-point mode.

The following instructions refer to the sample data in the activity.

1. Select the male in the 1993 column and age-0 row. You need to determine whether he survives to next year:
   - Check the survivability chart for an age-0 male; it is 75%.
   - Using your calculator, generate a random number between 0 and 1.
   - If the number generated is less than or equal to 0.75, this moose survives the year, so make a tally mark in the 1994 column, age-1 row (this moose is now a year older) and circle the tally mark in 1993 (indicating that you have dealt with this moose). If the random number is greater than 0.75, this moose dies (circle the mark that represents the moose to indicate that this moose has been considered).

2. Next consider the age-0 female. Follow the same procedure as for the age-0 male. (Since females do not start bearing young until age 2, you do not need to consider births until the age-2 row.)

3. Note that there are no moose of age 1 in the sample chart. Go to the age-2 row. There is one male moose. Run the simulation for that male. The survivability rate is 95% for a male of this age. Generate a random number for this moose. If the calculator number is less than or equal to 0.95, the moose survives. Make a tally mark in 1994 column in the age-3 row (let the moose have a birthday), and circle the 1993 tally mark so that you will remember that you have dealt with him. If the calculator number is greater than 0.95, the moose fails to survive (circle the mark).

4. Consider the age-3 females next. They could possibly give birth (the birth rate is greater than 0% for this row). Therefore, you have a few more steps to follow.

   - First, test for survival of the female in the usual way. If she dies, circle the mark and go on to the next moose. If she lives, make a mark in the 1994 column in the age-4 row, circle her 1993 tally, and then see if she had a calf (see the next step).
   - If the female lived, determine whether she gave birth by generating a random number and comparing it to the tabulated chance of giving birth (90%). If the random number is greater than 90%, she does not have a calf, and you can go on to the next moose. If it less than or equal to 90%, she has a calf. You need to find the gender of the calf (see the next step).
   - If the female gave birth, generate a random number. If it is less than or equal to 50%, the calf is male. If the random number is greater than 50%, the calf is female. Make a tally mark in the 1994 column in the age-0 row under Male or Female, as appropriate, to represent this calf.

5. Continue until all the 1993 moose have been considered, both for survival and for having calves.

6. You now have filled in the 1994 column. Find the sex totals and total moose. Fill in these numbers at the bottom of the 1994 column. Repeat the simulation through 1998 to complete the table.