CHAPTER 10 Calculator Notes for the fx-9750G Plus and CFX-9850GC Plus

Note 10A • List of Random Integers

There are several ways to generate a list of random integers within an interval. Here we will combine the three commands **Seq**, **Int**, and **Ran#** to generate a sequence of up to 255 terms. We will also look at a short program that will allow you to repeat the procedure quickly. You will see that with slight modifications, you can apply the commands and program to a number of familiar situations.

Integers from 1 through 100

The following procedure will generate a sequence of four random integers from 1 through 100.

From RUN mode:

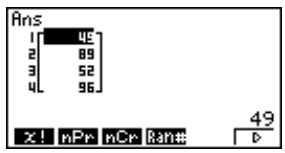
a. Press OPTN F1 (LIST) F5 (**Seq**).

b. Press OPTN F6 (▷) F4 (NUM) F2 (**Int**).

c. Press (OPTN F6 (▷) F3 (PROB) F4 (**Ran#**).

d. Press x 1 0 0) , X,θ,T , 1 , 4 , 1) + 1.

After completing **steps a–d,** press EXE. Press EXE EXE EXE and so on. Notice that each time you press EXE, a new sequence is generated.



By changing the 100 in **step d** to 50 and the 4 to 10, the same procedure would generate a sequence of ten random integers from 1 through 50. To make changes, press EXIT until you see the Run screen displaying **Done**. Then, press the left REPLAY arrow key. Use the arrow keys, DEL, and SHIFT [INS] to access the formula and change it.

If you add another step to the procedure, your calculator can store the sequence into a list. (See **Note 10B.**) Add the following step:

e. Press → OPTN F1 (LIST) F1 (**List**) 1.

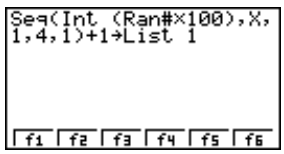
Now when you press EXE, you do not see the sequence, but you see **Done**. Press MENU, select STAT or LIST, and look at **List 1**.

Unfortunately, when you change modes to look at the list and then return to RUN mode, your screen is blank and you need to reenter the steps. You can avoid this inconvenience by storing your work in **Function Memory** and recalling it when you need it again. Add the following step to the procedure:

f. Press OPTN F6 (▷) F6 (▷) F3 (FMEM) F1 (STO) F1 (**f1**).

(continued)

The Function Memory screen shows what you have stored in each function key. Press EXIT twice to return to the Run screen, or press MENU to go elsewhere. Now if you leave RUN mode and then return to it, you can recall the stored procedure by repeating **step f,** substituting F2 (RCL) for F1 (STO).



When you recall your formula, it is "active" and you can use the REPLAY arrow keys, DEL, and SHIFT [INS] to go back into the formula and change it. For example, you can change the interval, send it to a list or not, change the list you send it to, and so on. You do not have to reenter the entire formula each time you want a variation. You can store different versions of the formula with different Function Memory numbers. If you turn your calculator off and then back on, your formulas will still be stored in memory and can be accessed by following **step f** with the F2 (RCL) substitution.

An alternative to the preceding procedure is to use a program called RNDINT. Download, link to, or manually enter the RNDINT program. (See **Notes 0F** and **0G.**) The program allows you to set the interval and to decide how many random integers you want in a sequence within the interval. The program stores the sequence in List 1 and displays it on the Run screen.

```
File Name:RNDINT
ClrText
"WHAT IS THE LOWEST"
"VALUE"?→L
ClrText
"WHAT IS THE
"HIGHEST"
"VALUE"?→H
ClrText
"HOW MANY VALUES"?→N
Seq(Int (Ran#×(H-L+1))+L,X,1,N,1)→List 1
List 1
```

To use the RNDINT program to generate a sequence of four random integers from 1 through 100, follow the steps below.

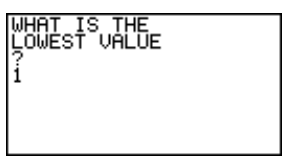
a. Press MENU and select PRGM. Arrow down to RNDINT and press F1 (EXE), or press EXE.

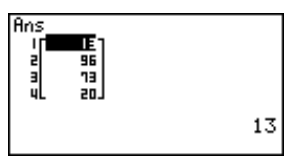
b. In response to the question WHAT IS THE LOWEST VALUE?, press 1 EXE.

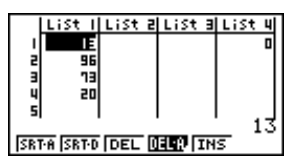
c. In response to the question WHAT IS THE HIGHEST VALUE?, press 1 0 0 EXE.

d. In response to the question HOW MANY VALUES?, press 4 EXE.

e. The Run screen shows a sequence of four integers. The same sequence is in List 1. You can check List 1 by going to STAT or LIST.







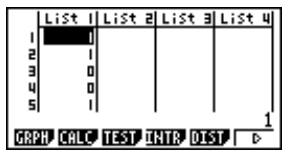
(continued)

Two Possible Outcomes: Coin Toss/True-False Test

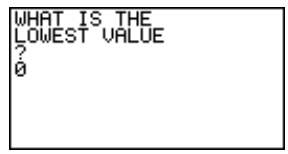
Suppose you want to simulate the results of 100 coin tosses with 0 representing tails and 1 representing heads. Use the REPLAY arrow keys, DEL, and SHIFT [INS] to change **step d** in the **Integers from 1 through 100** section to the following:

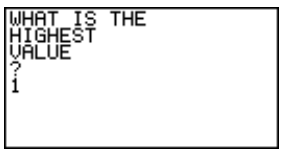
d. Press x 2) , X,θ,T , 1 , 1 0 0 , 1).

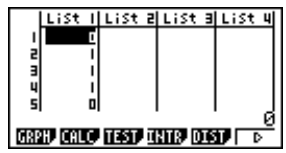
The results of this sequence of commands can be stored in a list. (See **step e** in **Integers from 1 through 100.**) You will use this formula later. You might want to store it in Function Memory. (See **step f** in **Integers from 1 through 100.**)



You can also use the RNDINT program to simulate the results of 100 coin tosses. Run the program and answer 0 to the question WHAT IS THE LOWEST VALUE?. Answer 1 to the question WHAT IS THE HIGHEST VALUE?, and answer 100 to the question HOW MANY VALUES?. The results appear on the Run screen and in List 1.







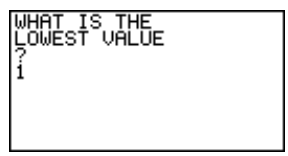
Six Possible Outcomes: Tossing a Die

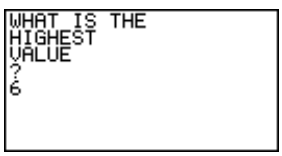
Suppose you want to simulate the results of four throws of a die. Use the REPLAY arrow keys, DEL, and SHIFT [INS] to change **step d** in **Integers from 1 through 100** to the following:

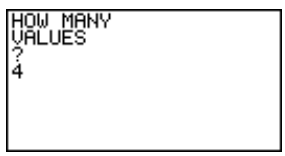
d. Press x 6) , X,θ,T , 1 , 4 , 1) + 1.

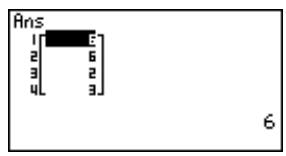
The results of this sequence of commands can be stored in a list. (See **step e** in **Integers from 1 through 100.**) You might want to store it in Function Memory. (See **step f** in **Integers from 1 through 100.**)

You can also use the RNDINT program to simulate the results of throwing a die four times. Run the program and answer 1 to the question WHAT IS THE LOWEST VALUE?. Answer 6 to the question WHAT IS THE HIGHEST VALUE?, and answer 4 to the question HOW MANY VALUES?. The results appear on the Run screen and in List 1.









(continued)

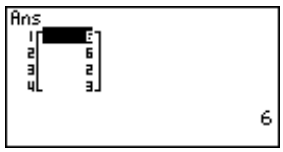
Two Possible Outcomes: Random Walk

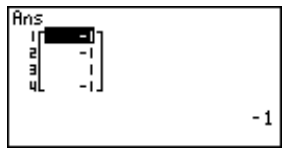
Suppose you want to simulate a random walk back and forth on a road by generating a sequence of four −1's and 1's. Use the REPLAY arrow keys, DEL, and SHIFT [INS] to change **steps a** and **d** in **Integers from 1 through 100** to the following:

a. Press 2 OPTN F1 (LIST) F5 (Seq).

d. Press x 2) , X,θ,T , 1 , 4 , 1) − 1.

The results of this sequence of commands can be stored in a list. (See **step e** in **Integers from 1 through 100.**) You might want to store it in Function Memory. (See **step f** in **Integers from 1 through 100.**)





To use the RNDINT program to simulate the random walk, generate a list of four 0's and 1's as in the **Coin Toss/True-False Test** procedure. That sequence is stored in List 1. Press MENU and select STAT. Define List 2 as 2List1−1. (See **Note 1K.**) Your sequence of −1's and 1's is in List 2. Remember, however, that each time you run the program, you will have to redefine List 2.

As an alternative, you can carefully edit the RNDINT program to give your desired results. (See **Note 0G.**)

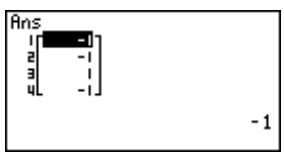
a. Press MENU and select PRGM.

b. Arrow down to RNDINT and press F2 (EDIT).

c. Arrow to the next-to-last command line that begins with Seq. Use SHIFT [INS] to insert 2 before Seq and add −1 after the last closing parenthesis.

d. Press EXIT and then F1 (EXE) to execute the altered program. Use 0 as the lowest value, 1 as the highest value, and 4 as the number of values.

Your sequence is in List 1. Every time you execute the program, you generate a new sequence. When you are finished, change the program back to its original form.

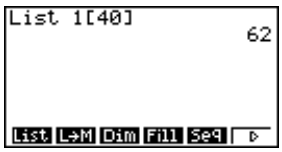


Viewing an Entry in a List

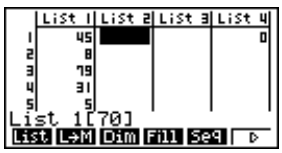
To avoid having to scroll through a list to see a distant entry, you can view any term directly on the Run screen. If in List 1 you have 50 random

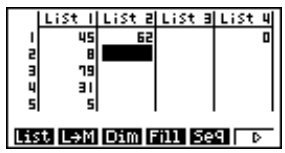
(continued)

integers from 1 through 100 and you want to see the fortieth term, from the RUN mode press OPTN F1 (LIST) F1 (List) 1 [[ ] 4 0 [ ]] EXE.



You can use the same command in any mode. For example, in LIST you can enter the seventieth term of List 1 as the first term of List 2.





Errors

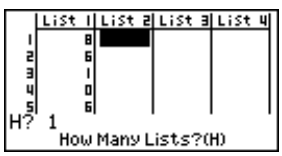
The longest sequence you can make has 255 entries. If you get an Arg ERROR, a Mem ERROR, or a Dim ERROR message, you are probably trying to construct a sequence that is too long, or you are referring to a term of a list that does not exist.

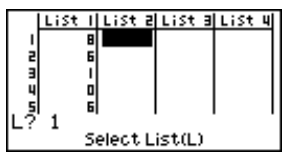
Note 10B • List Operations

(See **Note 1B** for instructions on entering a list.)

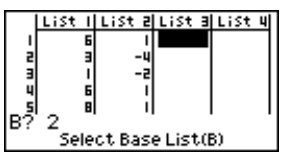
Sorting a List

When working with long lists, it is often more convenient to order the numbers in the list rather than to scan the list for the smallest, greatest, middle, or any other value. Press MENU and select LIST, then press F1 (SRT-A) for ascending order or F2 (SRT-D) for descending order. In response to the prompt How Many Lists?(H), press 1 EXE if you want to sort only one list. In response to the prompt Select List(L), press 1 EXE if you want to sort List 1.





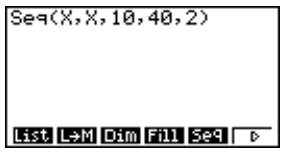
Suppose that you have two related lists and you want to sort List 2 in ascending order, but you want each entry in List 1 to stay with its corresponding entry in List 2. From the Main Menu, select LIST and press F1 (SRT-A). Press 2 EXE when you see the prompt How Many Lists?(H). Press 2 EXE when you see the prompt Select Base List(B), and press 1 EXE when you see the prompt Select Second List(L). Notice that in the screen on the far right below, the entries in List 2 are in ascending order and the original pairings have not changed.



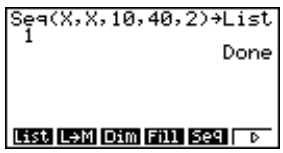
(continued)

Filling a List with a Sequence

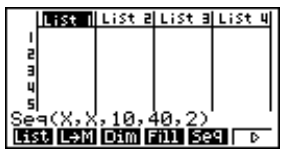
The sequence command Seq(X,X,1,50,1) will generate the sequence of integers from 1 through 50. The command Seq(X,X,10,40,2) will generate the sequence of even numbers from 10 through 40. To access Seq, select RUN from the Main Menu and press OPTN F1 (LIST) F5 (Seq). Then complete the command and press EXE.

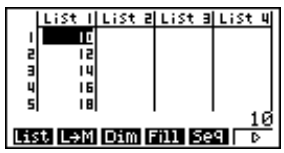


To fill List 1 with your sequence, press EXIT once or twice until you see your commands followed by Done. Press the left REPLAY arrow key to reactivate the commands and clear Done from the screen, and then press → OPTN F1 (LIST) F1 (List) 1 EXE. (See **Note 2C** for more on instant replay.)



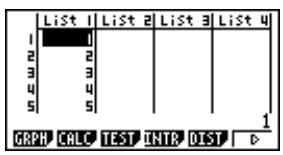
You can also fill a list with a sequence in LIST or STAT by defining the list name with the Seq command as in the screen below. (See **Note 1K.**)

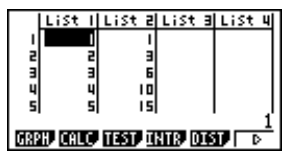




Cumulative Sum of a List

This calculator has a list function that will find the cumulative sums of a list and enter those sums into another list. For example, if List 1 is the sequence of integers from 1 through 10, then to generate the cumulative sums of List 1 and enter them into List 2, press OPTN F1 (LIST) F6 (▷) F6 (▷) F3 (Cuml) F6 (▷) F1 (List) 1 → F1 (List) 2 EXE.





Other List Functions

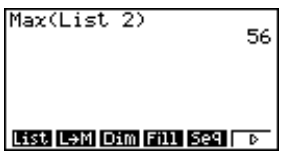
In RUN mode, you can find the mean, median, sum, and other important numbers associated with a list by pressing OPTN F1 (LIST) F6 (▷). Press F6 (▷)

(continued)

to see more options. Always specify the list name and close any parentheses. (Also see **Note 1C.**)

min	the minimum value in a list
max	the maximum value in a list
mean	the mean of the list entries
median	the median of the list entries
sum	the sum of the list entries

To get the screen below, press
OPTN F1 (LIST) F6 (▷) F2 (Max) OPTN F1 (LIST) F1 (List) 2) EXE.



Note 10C • Calculator Coin Toss

100 Trials

a. Enter the sequence of integers from 1 through 100 into List 3. This list will number the trials. (See **Note 10B.**)

b. Enter 100 randomly generated 0's and 1's into List 1. List 1 will represent the 100 coin tosses. Let 0 represent tails and 1 represent heads. (See **Note 10A.**)

c. Use Cuml to calculate the cumulative sums of List 1 and store them in List 2. (See **Note 10B.**)

d. Calculate the ratio $\frac{\text{List 2}}{\text{List 3}}$ and store the results in List 4. (See **Note 1K.**)

e. Make a scatter plot using List 3 for the x-values and List 4 for the y-values. Use the small dot as the mark. Set appropriate view window parameters. (See **Note 1F.**)

f. Enter the probability of tossing a head into Y1 in the Graph Func screen in GRAPH mode. Graph this equation along with the scatter plot. (See **Note 1J.**)

200 Trials

By changing 100 to 200 in **steps a** and **b** in the previous section, you can modify 100 coin tosses to work for 200 tosses.

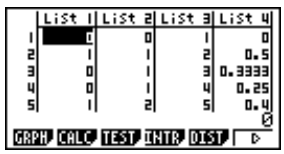
You can also enter the short program that follows so that you can rerun the simulation without reentering the commands each time. This program only enters data into lists. You then need to make a scatter plot and graph the probability as in **steps e** and **f** in the preceding section.

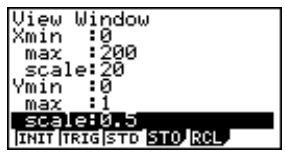
Y1 can be changed from $\frac{1}{2}$ to the probability of an event that does not have outcomes that are equally likely.

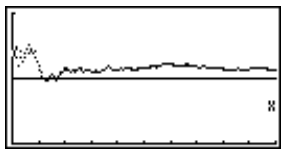
(continued)

```
File Name:PROBSIM
Seq(X,X,1,200,1)→List 3
Seq(Int (Ran#×2),X,1,200,1)→List 1
```

```
Prog "CUMSUM"
List 2÷List 3→List 4
```



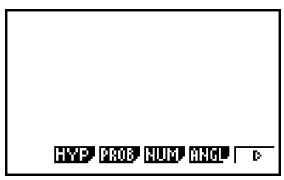




Note 10D • Permutations

To find numbers of permutations, use the nPr command. To find the nPr command, go to the Run screen, and press OPTN F6 F3 (PROB) F2 (nPr). First, enter the value of *n,* which is the number of objects. Then, enter the nPr command, and enter the value of *r,* which is the number of objects chosen. Then press EXE.

For example, to find the number of arrangements of 5 objects chosen 3 at a time, enter 5 nPr 3. The answer shows that there are 60 arrangements.



Note 10E • Combinations

To find numbers of combinations, use the nCr command. To find the nCr command, go to the Run screen, and press OPTN F6 F3 (PROB) F3 (nCr). First, enter the value of *n,* which is the number of objects. Then, enter the nCr command, and enter the value of *r,* which is the number of objects chosen. Then press EXE.

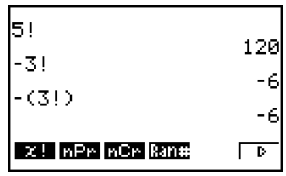
For example, to find the number of groupings of 5 objects chosen 3 at a time, enter 5 nCr 3. The answer shows that there are 10 different groupings.

Note 10F • Factorials

To find the factorial command, press OPTN F6 F3 (PROB) F1 (*x*!). For example, to evaluate 5!, press 5 OPTN F6 F3 F1 EXE.

(continued)

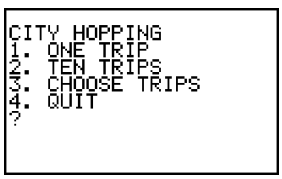
In the order of operations, factorial has higher precedence than negation, so –3! is equivalent to –(3!).

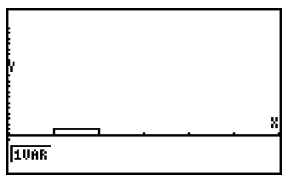


Note 10G • CITIES Program

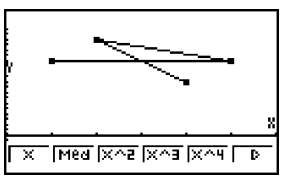
The CITIES program will simulate random travel among six cities, stopping when a previously visited city is revisited.

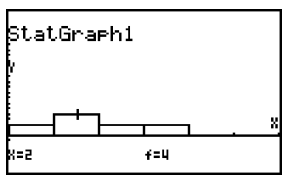
Run the program and choose whether you'd like to view one, ten, or some other number of trips. If you choose 1. ONE TRIP, you'll see travel among the six cities labeled A–F, and the histogram will show you the total number of cities visited on this trip. When the program stops, retrieve the histogram by choosing STAT from the Main Menu, and then pressing F1 (GRPH) F1 (GPH1). Then, press EXE to simulate another trip, or quit.





If you choose 2. TEN TRIPS, you'll see ten trips simulated, and the histogram will show how many cities were visited during the ten trips. As the program runs, you will see a map of the six cities and a trace of the route taken during each trip. When the program stops, retrieve the histogram by choosing STAT from the Main Menu, and then pressing F1 (GRPH) F1 (GPH1). Selecting F2 (GPH2) instead will allow you to see the tenth route taken before the program finished. The first screen shown below indicates that four cities were visited during the tenth trip. The second screen shows that in the ten trips, two trips visited one city, four trips visited two cities, two trips visited three cities, and two trips visited four cities.





If you choose 3. CHOOSE TRIPS, you can choose any number of trips up to the limit of your calculator's available memory. The program won't show each trip, but it will show a histogram with the number of trips that visited 1, 2, 3, 4, 5, and 6 cities. After you run this option, press SHIFT [QUIT] EXE to continue.

(continued)

```
Filename:CITIES
ClrList↲
ClrGraph↲
Lbl 0:1→N↲
"CITY HOPPING"↲
"WHEN DONE PRESS MENU,STAT TO VIEW
  GRAPHS"↲
"1. ONE TRIP"↲
"2. TEN TRIPS"↲
"3. PICK N"↲
"4. QUIT"↲
?→A↲
ClrText↲
A=1⇨Goto 1↲
A=2⇨Goto 2↲
A=3⇨Goto 3↲
A=4⇨Goto 4↲
Lbl 2:10→N↲
Lbl 1↲
{0,0,0,0,0,0,0,0,0,0}→List 1↲
{0,0,0,0,0,0}→List 3↲
{0,0,0,0,0,0}→List 4↲
S-Gph1 DrawOn,Hist,List1,1↲
S-Gph2 DrawOn,xyLine,List3,List4,1,
  Square↲
ClrGraph↲
S-Gph3 DrawOn,Scatter,List 5,
  List 6,1,Cross↲
{2,5,6,5,2,1}→List 5:{4,4,2.5,1,
  1,2.5}→List 6↲
ViewWindow 0,7,1,0,5,1↲
For 1→J To N↲
{0,0,0,0,0,0}→List 2↲
1→K↲
Int (Ran#×(6))+1→C↲
List 5[C]→List 3[K]↲
List 6[C]→List 4[K]↲
S-Gph2 DrawOff↲
DrawGraph↲
Plot 2,4↲
Plot 5,4↲
Plot 6,2.5↲
Plot 5,1↲
Plot 2,1↲
Plot 1,2.5↲
Do↲
1→List 2[C]↲
F-Line List 3[K],List 4[K],List
  3[K],List 4[K]↲
F-Line List 3[K],List 4[K],List
  3[K],List 4[K]↲
F-Line List 3[K],List 4[K],List
  3[K],List 4[K]↲
Int (Ran#×(6))+1→C↲
K+1→K↲
List 5[C]→List 3[K]↲
List 6[C]→List 4[K]↲
F-Line List 3[K-1],List 4[K-1],List
  3[K],List 4[K]↲
LpWhile List 2[C]=0↲
K-1→List 1[J]↲
S-Gph1 DrawOn↲
S-Gph2 DrawOn↲
DrawGraph↲
Plot 2,4↲
Plot 5,4↲
Plot 6,2.5↲
Plot 5,1↲
Plot 2,1↲
Plot 1,2.5↲
ClrGraph↲
ViewWindow 0,7,1,0,5,1↲
Next↲
Goto 0↲
Lbl 3↲
S-Gph2 DrawOff↲
S-Gph3 DrawOff↲
"N"?→N↲
Seq(0X,X,1,N,1)→List 1↲
ViewWindow 0,6,0.5,0,N÷2,1↲
For 1→J To N↲
{0,0,0,0,0,0}→List 2↲
0→K↲
Int (Ran#×(6))+1→C↲
Do↲
K+1→K↲
1→List 2[C]↲
Int (Ran#×(6))+1→C↲
LpWhile List 2[C]=0↲
K→List 1[J]↲
Next↲
S-Gph1 DrawOn,Hist,List 1,1↲
DrawStat↲
Lbl 4↲
```


Note 10H • Infinite Sums

To find the sum of terms of a recursive sequence, you can use a recursive routine. Rather than finding the *terms* of a recursive sequence, as you did in Chapters 0 and 3 (See **Notes 0D and 3A**), here you'll find the *sum* of the terms of a recursive sequence.

For example, to find the sum of

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \ldots$$

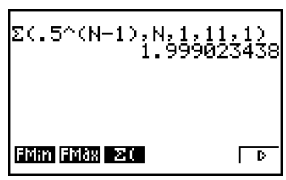
you create the sequence 1, 1.5, 1.75, and so on. This sequence shows the sum of the first one term, the first two terms, the first three terms, and so on. Using the sum feature, Σ(, you can look at what happens to this sequence as you add more terms of the sequence. To find the sum after any number of terms has been added, write a formula that will generate each term. Each term is defined by $\left(\frac{1}{2}\right)^{n-1}$, where n is the term number.

To enter this formula into your calculator, follow these steps:

a. From the Run screen, press OPTN F4 (CALC) F6 F3 (Σ().

b. Now, you are ready to build the formula. Press ALPHA [N] to enter the letter N as the index variable.

The formula is followed by values that tell the calculator how to evaluate the sum. If, for instance, you want to find the sum of the first 11 terms, you enter N, 1, 11, 1. This tells the calculator to sum the expression with N terms, starting at $N = 1$ and ending at $N = 11$, increasing by one unit each time. The sum appears to be approaching 2. Test this conjecture by finding the sum of more terms.



Now, consider the sum

$$1 \cdot \left(\frac{1}{3}\right) + 2 \cdot \left(\frac{2}{3}\right)\left(\frac{1}{3}\right) + 3 \cdot \left(\frac{2}{3}\right)^2\left(\frac{1}{3}\right) + 4 \cdot \left(\frac{2}{3}\right)^3\left(\frac{1}{3}\right) + 5 \cdot \left(\frac{2}{3}\right)^4\left(\frac{1}{3}\right) + \cdots$$

Notice that one factor in each term is the counting number, another factor in each term is always $\frac{1}{3}$, and the remaining factor is a power of $\frac{2}{3}$, beginning with $\left(\frac{2}{3}\right)^0$. So an expression for the terms is $n \cdot \frac{1}{3} \cdot \left(\frac{2}{3}\right)^{n-1}$. Enter the formula into the calculator as described before and evaluate the sum after some number of terms. In this case, the sum appears to approach 3. Experiment by taking the sum of a greater number of terms to confirm this guess.