TRAPRULE, Problem Set 1-4, Problem 5 (pages 22–23)

This program evaluates the definite integral of a given function between lower and upper limits of integration using the trapezoidal rule with any desired number of increments. Before you run the program, store the function for the integrand as Y_1. When you run the program, the grapher will prompt you to enter the lower and upper limits of integration, *A* and *B*, and the number of increments, *N*. The grapher then computes and sums the successive *y*-values, using half the first and half the last, and saves the sum as *S*. The grapher displays the approximate value of the integral, which it has saved internally as the variable *I*. Along the way the grapher displays the successive *y*-values to give students a sense of how rapidly it is calculating. You may want to omit this instruction so that the program will run faster, particularly for larger numbers of increments.

TRAPRULE (TI-83, TI-83+, and TI-84+)

```
:Prompt A
:Prompt B
:Prompt N
:A→X
:0→S
:(B-A)/N→D
:0→C
:Lbl 1
:Y1+S→S
:X+D→X
:Y1+S→S
:IS>(C,N-1)
:Goto 1
:SD/2
:Disp "INTEGRAL"
:Disp Ans
```

TRAPRULD, Problem Set 1-4, Problem 6 (page 23)

The name comes from "trapezoidal rule from data." This program evaluates an integral of a function for which *y*-values are given, assuming that the *x*-values are evenly spaced. Before running the program, store the given *y*-values in list L_1. Upon running the program, the grapher will prompt you to enter the number of increments (which must be one less than the number of data points). Then it will prompt you to enter the width of each increment, *DX*. The grapher then runs the program and displays the approximate integral, which it has stored in its memory as *I*. Note that the *x*-values themselves appear nowhere in the program.

TRAPRULD (TI-83, TI-83+, and TI-84+)

```
:Disp "INCREMENTS"
:Input N
:Disp "DX"
:Input D
:(D/2)*(2*sum(L1)-L1(1)-L1(N+1))→I
:Disp I
```