Using Numerical Integration to Approximate $\int_A^B f(x)\,dx$

The following programs for the TI-82/83 and TI-85/86 compute three different approximations of the integral.

Program Input:*

A,	the lower limit of integration
B,	the upper limit of integration
N,	the number of segments of equal length into which $[A, B]$ is divided (except for Simpson's Rule; see footnote **)
$f(x)$,	the function being integrated.

Program Output:

a list of three numbers: MP, TR and SI, where
$MP =$ the approximation using the Midpoint Rule;
$TR =$ the approximation using the Trapezoidal Rule;
$SI =$ the approximation using Simpson's Rule.

The Program Itself:

for the TI-82/83

```
PROGRAM: NUMINT
:Prompt A,B,N
:(B-A)/N → H
:0 → S
:0 → T
:For(I,1,N,1)
:S + Y1(A+I*H) → S
:T + Y1(A+(I-.5)*H) → T
:End
:H*T → U
:H*(S+.5*(Y1(A)-Y1(B))) → V
:Disp U
:Disp V
:Disp (2*U+V)/3
:Stop
```

for the TI-85/86

```
PROGRAM: NUMINT
:InpSt ''enter function'',STRING
:St>Eq (STRING,Y1)
:prompt A,B,N
:(B-A)/N → H
:0 → S
:0 → T
:For(I,1,N,1)
:evalF(Y1,x,A+I*H) → F
:S + F → S
:evalF(Y1,x,A+(I-.5)*H) → G
:T + G → T
:End
:H*T → U
:evalF(Y1,x,A) → C
:evalF(Y1,x,B) → D
:H*(S+.5*(C-D)) → V
:(2*U+V)/3 → W
:Disp U
:Disp V
:Disp W
:Stop
```

Comments:

1. After entering the program, it is essential to test it for typographical errors by calculating some special cases (try $\int_0^1 x^n\,dx$ for $n = 1, 2, 3$).

---

* On the TI-82/83, you will be prompted for A, B and N during the execution of the program, but *before executing the program* you must enter the function f in the usual way as `Y1=f(x)` in the function editor. On the TI-85/86, you will be prompted for A, B, N and f during the execution of the program.

** For Simpson's Rule, the actual number of segments will be twice the value of N which you enter. If you enter $N = 20$ or 25, for example, the program will produce MP for a subdivision of $[A, B]$ into 20 or 25 subintervals, respectively. Likewise for TR. However, it will give SI for a subdivision into 40 or 50 subintervals, respectively.

2. This program is not designed as production-perfect software; no attention is paid to error checking, roundoff error or efficiency. The program's purpose is to provide rough answers to the question "What happens when you apply the `@#!$%&!` Rule to estimate the integral?"
3. If you have some other calculator, you will have to figure out how to modify these programs.

Entering the program on the TI-82*:

Push the `PRGM` button. The following menu will appear:

`EXEC EDIT NEW`

`EXEC` is used to execute a program that has already been written, `EDIT` is used to modify an existing program, and `NEW` is used to write a new program. Accordingly, use the cursor keys to move the cursor over to `NEW` and push `ENTER`. Then, on the program screen, `NAME =` will appear. Type in the program name, say `NUMINT`. (Note: Normally when you wish to enter a letter, you must first push the `ALPHA` key. However, when you type a program name, the calculator sets the `ALPHA` key automatically, and so you need only type in the letters.) After typing in the program name, press `ENTER` and begin writing the body of the program.

You enter the program commands from two menus. To gain access, push the `PRGM` button. Then on the screen will appear:
`CTL I/O EXEC`

The first menu, `CTL`, (for control) has the commands: `For(...` and `End`, as well as many others. The second menu, `I/O`, (for input/output) gives the commands: `Prompt` and `Disp`.
For example, to enter the initial line:

```
:Prompt A,B,N
```

first push the `PRGM` button. Move the cursor across to `I/O` and push `ENTER`. Take the cursor down to `2:Prompt` and again push `ENTER`. On your program screen `Prompt` will appear. Then type in `A,B,N` and push `ENTER`. Remember to push the `ALPHA` key before you attempt to type a letter. Pushing the `ENTER` key automatically moves you to the next blank line with the beginning colon supplied by the calculator.

If you leave the program editor, by accident or intention, to return back to the program that you were writing, push the `PRGM` button. Use the cursor keys to take the cursor across to `EDIT` and push `ENTER`. Take the cursor down to the program name, in this case `NUMINT`, and push `ENTER` again.

Many of the remaining symbols are letters, numbers, punctuation marks, or symbols for arithmetic operations and are directly visible on the calculator keyboard. However, some are not and are found as follows:

$\rightarrow$ is the `STO` key. (2nd from bottom on left)

`Y1` is the function name. To enter, first push the `[2nd]` key (blue button) followed by the `VARS` key. A menu will appear. Since a function name is desired and the cursor is already on `1:Function`, push `ENTER`. A menu will appear with the cursor in `1:Y1`. Push `ENTER` again and `Y1` will appear on the program screen.

Running the program on the TI-82

First use the `Y =` button to enter `Y1`. Then press `PROG`, choose `EXEC` and choose `NUMINT`. You should be prompted with `A=?`; enter the value and push `ENTER`, and continue.

Testing your Understanding of the Programs

Question 1: If the TI-82/83 program said `H * S` $\rightarrow$ `U` at line 9 instead of `H * T` $\rightarrow$ `U`, then what would the output `U` represent? Illustrate for the example $f(x) = x^2$, $A = 1$, $B = 2$ and $N = 4$, by drawing the graph of f and the related figure whose area is `H * S`.
Question 2: Three long sums are computed in this program, although the `FOR` loop only is computing two sums. Explain why the third one, Simpson's Rule, comes out so simply as `(2 * U + V)/3`.

---

* The procedure is only slightly different on the TI-83. On the TI-85 and 86 there are more significant differences.