 ... you can't break the program, so **explore!** ... you can't break the program, so **explore!** ... you can't break the program, so **explore!** ... you can't break the program, so **explore!** ... you can't break the program, so **explore!**

640:192:03 **Part I: playing with arithmetic on `maple`** **10/14/96**

What I'd like you to type at the terminal will be in *this typeface* with numbers that look like this *1 4 7 9* and these suggestions will be indented about an inch. I'll ask you to hit the "enter" key (new line) with the request RET. Now, please, log in to `eden` and get a prompt in an x-window. Then type

> *xmaple &* RET

The system should respond with a `maple` screen on your display. There are standard ways for you to move or resize the sceen, and various `maple`-specific command possibilities. `maple` is a huge program with a great many capabilities. We'll just explore a few of them. Right now I'd like you to move your mouse into the `maple` window. You'll this, which is the command line:

> $>|$

The symbol for your cursor is | and it is currently at an input line, indicated by the > sign. Please type

> *3+2* RET

Note that **nothing happens** except that you get another input line! Move your cursor back (with the arrow keys) to the first input line, and then move the cursor to the the end of the line. The input line should look like

> $> 3 + 2|$

Then continue your typing with a semicolon followed by a return:

> *;* RET

Something should happen. You should get 5 and a new input line. You can move your cursor up and down. Do this, and see the word "Input" on the `maple` screen change in succession to "Separator" and then to "Output" and then to "Input" when your cursor gets back to the original line.
Now move your cursor back to your new input line. Type

> *17*3;* RET

and see what the result is. At the next input line type

> *"+5;* RET

and explain the result. What do you think the meaning of the symbol *"* is?
Now type the following to learn what ˆ means.

> *2ˆ3;* RET

But ... I made a mistake! I wanted you to calculate the 300^{th} power of 2. Please do the following: move your cursor back to the input line with 2ˆ3 and position it in the following place: > 2ˆ3|; and now type *00* and immediately hit RET. What happened? Please compute 3^{300} in the same fashion by moving your cursor and changing the input line. What are the first 5 and last 5 decimal digits of this number?
A little more: please type (look carefully here – I'm asking for a **colon**, *not* a semicolon!)

> *5+6:* RET

You should immediately get another input line. Type (for example)

> *"+7;*

and deduce what `maple` does when an input line ends with a *:* (that is, a colon). Note that computations might and do occur which have results that are huge and silly to print out if you don't need them — 2ˆ(2ˆ(2ˆ(2ˆ2))), for example. Try that sometime on your own, please, with ";" rather than a ":" and see what happens (hah!). Onward: please type

> *20;* RET

and see the result. Go back with your cursor and put a space between the 2 and the 0 and hit RET. What's the result?
Now let's try

> *2*3+7;* RET

and observe that `maple` follows the usual rules of precedence. Can you put parentheses in so that `maple` will compute two times the sum of three plus seven instead? Remember to hit RET after you make the alterations. You should have gotten 20 as your answer, of course. If you did **not** make an error inserting the parentheses, go back and take one out (create an intentional **error**!) and then hit RET. What happens? <u>You haven't broken anything</u>. Let's keep exploring. Please get a new input line and try these commands in succession to learn how to do more arithmetic and to explore more features of `maple`.

OVER

2/3; **RET**

`maple` computes "exactly" and can do some (fourth grade?) arithmetic:

*"*300;* **RET**

Now try

sqrt(2); **RET**

and now

"^2; **RET**

so `maple` knows the "meaning" of fractions and square roots – or at least how to manipulate them. And now try (remember, if you mess up with a parenthesis or something else, just go back and do it again – nothing is broken!):

(sqrt(2)-1)^5; **RET**

This result is puzzling. Sometimes `maple` is lazy. Let's urge it to work by writing

expand("); **RET**

That's better. But what if we want or need decimal approximations? Try

evalf(sqrt(2)); **RET**

Parentheses need to be matched – always a source of anxiety as more and more complex expressions and commands are typed.

What if we want more digits of $\sqrt{2}$? We can coax *evalf* to do this with more informed use. To see how, type

help(evalf); **RET**

Another screen should pop up. When I use `maple` I tend to need a lot of help so the help screens accumulate on my display. You can "click" them on and off, and eliminate them entirely (click on "File" in the upper left corner of the screen to see how). Read the evalf screen (I usually skip to the examples on any help screen first!) until you can figure out how to get the first 100 digits of $\sqrt{2}$. What is the one-hundredth digit after the decimal point? Can you tell me the three-hundredth digit after the decimal point of $17^{1/3}$? Use lots of parentheses, even in exponents, to inform `maple` clearly what you want.

Now try

1400/24; **RET**

and we learn that `maple` knows how to factor integers automatically. Can you get `maple` to factor your social security number? How would you find a factoring command in `maple`? If the first thing you try with the *help* command doesn't work, look at the references on the SEE ALSO line and check one of them.

We can go on to try some algebra. But notice that you can stop your `maple` session at any time in a variety of ways. One way that is polite to the system and also simple for you is to type

quit **RET**

and your `maple` window will disappear. Of course all of your work will also vanish, but at some other time you can explore the possibilities of saving your work using one of the options gotten by clicking on the File button.

Disclaimer! Non-advertisement!! Important information!!!

Symbolic manipulation programs such as `maple` are becoming increasingly available. Other popular programs with about the same capabilities are `derive` and `mathematica` and there are many special purpose programs in various fields of science, engineering, and mathematics which have extensive "intelligence" to analyze models. We're considering `maple` here because Rutgers has a site license for this program, and it should be generally available on Rutgers systems. The specific instructions won't be the same from program to program, but many of the same ideas will be present. Students should expect to have a machine do tiresome or elaborate symbolic computations as well as numerical computations.