Exercise: *Due by classtime Monday, February 11th, 4:00 PM*

Journal files: `YourName_ex3_#.#.pro` where `#.#` is actually the number of the exercise the file begins with, e.g., 4.0. There are a lot of parts to this exercise so you should split it into at least two different journal files. Rather than tell you where to split it up, I'll let you pick what feels convenient. The only requirement is that you make the split between book exercises, i.e., not in the middle of one.
You should do this in your `~/ASTR2600_assignments/assignment3/` directory.

Exercise A. The python version of Tutorial 7

Exercise 4.0: Optional parameter in `plot`, using `max`.

Exercise 5.0: Play around with some formatted printing

Exercise 6.0: Writing a text file. (Repeats last part of Exercise 5.0 but writes output to a file.) This says to modify the file with a text editor. Use `vim` or `gedit` for that. (Reminder: Open another terminal window. At the UNIX prompt, `cd` into your `assignment3` directory. Then run `gvim` by typing `gvim powerTable.txt`. This gets you running the `gvim` text editor, with the `powerTable.txt` file.)

Exercise 6.1: Reading the text file created in Exercise 6.0.

Exercise 6.2: Binary files This has you compare two versions of output. You may need to make your terminal window taller by resizing it with the mouse to see both printouts. It then has you "use the operating system" to compare the sizes of two files. Use `ls -l` in UNIX to see the sizes of the files. (Use a separate window).

Exercise 6.3: Saving an image file. If you'd like, you can just restore the file from Exercise 6.4 (i.e., do 6.4 first) instead of redefining the `x & y` arrays. You can view this jpg file by using the window file browser in Linux. Work your way down to your `ASTR2600_assignments/assignment3` directory and double-click on the file `YourName_mySincPlot.jpg`. It should come up in an image viewing program.

Exercise 6.4 (Not in the book): Saving and restoring an IDL save file.

```
help             ; Shows the list of variables IDL is currently aware of (if any)
.reset_session   ; Tells IDL to forget all those variables (& procs & funcs)
help             ; Verify that IDL has forgotten all those variables

; create an array of x values from -10 to 10
x = -10 + 20.*findgen(100)/99.
y = sin(x) / x                    ; sin(x)/x is called a "sinc" function
help                              ; See that you now have x & y arrays defined
plot, x, y                        ; Take a look at it

; Save the x & y variables in an IDL save file:
save, x, y, filename="sinc.idlsave"

.reset_session    ; Tell IDL to forget all about those
help              ; Verify that IDL has forgotten all those variables


; restore the variables from the file
restore, filename="sinc.idlsave"
help                         ; See that IDL now knows x & y arrays
plot, x, y                   ; Plot them
```

Exercise 6.5 (Not in the book): Working with a file whose "endian" convention is different. We have two data files, one with x values, the other with y values. Each contains data in a "self-describing" format. The first data in each file is a long integer that says how many floats follow it. The file then has that many floats. A convenient way to say this is "It contains a long integer, N, followed by N floats." The files are `/home/shared/astr2600/data/assignment3/sinc_x.dat` and `/home/shared/astr2600/data/assignment3/sinc_y.dat` We need to open both, read N for each, then read in the x and y values. Then plot, x, y

```
; open BOTH data files
; We'll do this using some string operators.
; This is just to go through those motions.
; We could just define the filenames in the open statements
path = "/home/shared/astr2600/data/assignment3/"
; the path is something the filenames share
xfilename = path + "sinc_x.dat"        ; build the full filename with path
yfilename = path + "sinc_y.dat"
print, xfilename                       ; verify that the filenames are right
print, yfilename
openr, xlun, xfilename, /get_lun   ; Open the files.  Notice that this LUN is xlun
openr, ylun, yfilename, /get_lun   ; and this LUN is ylun

; Now read in nx and ny from the two files
nx = 0L                                ; "declare" nx and ny to be longs.
ny = 0L                                ; (It doesn't matter what actual value we use)
readu, xlun, nx                        ; read in the number of x values
readu, ylun, ny                        ; read in the number of y values
                                       ; we expect nx is equal to ny

x = fltarr(nx)                         ; create the array for x
                                       ; OOPS!  Something's wrong!
print, nx                              ; check the nx value

; It makes no sense that nx is negative.
; This is our clue that the datafile is big-endian and we are little-endian
; (or the other way around).  Whichever case, we need to swap the bytes in nx

nx = swap_endian(nx)           ; replaces nx with a byte-swapped version of nx
print, nx                      ; Does this value make more sense?
print, ny                      ; verify that ny is similarly goofy
ny = swap_endian(ny)           ; so also need to swap bytes for ny

x = fltarr(nx)               ; NOW we can create our arrays with sensible numbers
y = fltarr(ny)
readu, xlun, x               ; Read in the actual arrays of numbers
readu, ylun, y

free_lun, xlun               ; We're done reading the files so we clos them
free_lun, ylun

x = swap_endian(x)           ; x and y must ALSO be byte-swapped
y = swap_endian(y)

plot, x, y                   ; plot a nice sinc function
```

Exercise 7.0: A sorting exercise.

Exercise 7.1: Comparisons & Booleans.

Exercise 7.2: Using where with comparisons & Booleans.

Exercise 7.3: Finding where the minimum is using where. You've now seen 3 ways to find a minimum (or maximum): Use the min & max functions, use where, and compute the derivative and see where it crosses 0.

Turn in via github:
All journal files of the exercises, `YourName_ex3_X.X.pro`
`testArray.txt` (These files do not have to have your name in their name.)
`testArray.dat`
JPEG image file from Exercise 6.3: `YourName_mySincPlot.jpg`

Whuduzitdo: All Whuduzitdo's from Chapters 4-7.

Graded Homework
Due by midnight the night of Wednesday, February 13th, 2013
Do all of this in your `ASTR2600_assignments/assignment3` directory

Homework 4.0: Journal file: `YourName_hw4.0.pro`. Finding peak of Planck function using max.

Homework 6.0: Journal file: `YourName_hw6.0.pro`. Writing binary file.

You don't need to do Homeworks 6.1 & 6.2 & 6.3

Homework 6.4 (Not in book): Journal file: `YourName_hw6.4.pro`
In the directory `/home/shared/astr2600/data/assignment3/` there is a file, `plotData.dat` It's a binary file containing a long integer, say, N, followed by an array of N floats. Read this file and plot the array. (Don't copy the file into your local directory and read that. Include the full directory path in the filename when you open the file. It's not always practical to make local copies of data files. You need to know how to read in files from other directories.) Since I don't give you an x array, you can assume evenly spaced values, so "plot, y" works.

Homework 7.1 (Not in book): Journal file: `YourName_hw7.1.pro`
In the `/home/shared/astr2600/data/assignment3` directory there are two files:
`lif2bwave.dat` and `lif2bflux.dat`. These are binary files containing ultraviolet spectra data from a star.
`lif2bwave.dat` contains the wavelengths in angstroms.
`lif2bflux.dat` contains the flux in ergs cm^{-2}
The format of these binary files is a long integer, N, followed by N floats.
a) Read these files in and plot the flux vs. wavelength. (Again, don't make local copies of the files. Read them in from where they sit.)
Save an image of this plot as `YourName_fullSpectrum.jpg`
b) The plot has a double peak near 990 angstroms. Plot about a 5-angstrom range roughly centered on this double peak so that we see it clearly. (Use the where function to select the portions of the wave and flux arrays that fit this. This is like "zooming in" on that part of the spectrum.)
Save an image of this plot as `YourName_doublePeak.jpg`

Turn in on github:
Journal files of the homework:
`YourName_hw4.0.pro`
`YourName_hw6.0.pro`
`YourName_hw6.4.pro`
`YourName_hw7.1.pro`
Image files:
`YourName_fullSpectrum.jpg`
`YourName_doublePeak.jpg`