

**EOSC 211: Practice with loops, if, strings, fopen, fclose, fprintf, logical indexing****Group #:** \_\_\_\_\_ **Name:** \_\_\_\_\_**File handling is a very important capability in data analysis. Here's an example of a common type of situation:**

You have a set of files containing pairs of measurements of magnetic field strength taken 20 times per second continuously. The data are organized such that there is one file per day for every day in the year and each file contains two columns: time in seconds and magnetic field strength in nT. You can call these variables TIME and BMAG. Assume that the files are named as follows *mag001.dat*, *mag002.dat* ... through to the file for Dec 31<sup>st</sup>, *mag365.dat*

You want to do some analyses of data taken each day but only inside a time interval that is specific to each day. e.g., you want to find the maximum magnetic field in a given time interval, and you want to output the day, the maximum field and the time (in seconds) at which this occurs, to a file called *maxmag.out*. The time interval to be analyzed each day is given by a start time and end time that are kept in another file called *crossings.dat*. This has 3 columns of data: the first contains the day-of-the-year i.e. 1 thru 365 (call this DOY), the second contains the start time (call this TIN) and 3<sup>rd</sup> contains the end time (call this TOUT) in seconds.

- A. First write down a procedure/algorithm for doing this. Don't write full matlab code – write a flow chart, or shorthand code or whatever works for you to map out your plan-of-attack.... Remember: What are the inputs / outputs? What are the repetition / selection parts of this problem?

HINT: Often it helps to first solve a simpler problem – e.g. what if you had a data file for one day only? Break the problem down into pieces and see if you can write the algorithm just for this one day of data.

- B. Write the code to load the content of one of these files (e.g. `mag001.dat`) into a variable called `bdata`. Because we are going to do this again and again for different filenames it will be easiest to load a file whose file name is in contained in a string variable `strfil` so first assign the file name to this variable.
- C. Recall: you can turn a number into a string using `num2str`, and you can concatenate 2 strings using `strcat` or `[str1 str2]`. Use a loop to successively replace the contents of `strfil` with the filenames for days 100 thru 365.
- D. Now include the code needed to include days 1 thru 365 in the loop and replace the contents of `strfil` with the filename for any day of the year. Be careful because the filenames for days 1 thru 9 are e.g. *mag001.dat*, and those for days 11 thru 99 are e.g. *mag048.dat*

- E.** Now imagine that all you wanted to do was to find the maximum magnetic field value for each day and the time at which this occurs (i.e. ignore the *crossings.dat* file for now) and write it to a file called *maxmag.out*. Add the code to your loop to open and close the file *maxmag.out* (use *fopen* and *fclose*) and write out the DOY, the maximum field for each day, and the time (in seconds) at which that occurs (use *fprintf*).
- F.** Now add code that will compute the maximum magnetic field for each day ONLY between the times given by TIN and TOUT for that day in the file *crossings.dat*. USE LOGICAL INDEXING. You will first need to load the file *crossings.dat* and to select the appropriate TIN and TOUT for the day.

- G.** You are done! Finally, look at your code and decide if there are any parts of it that could be put into functions. If so write down the function definition line and a short version of the help lines.