Reading Day text Files

Function: Read_File

Input: Day Text File   e.g. Day64.txt

Be in directory with text file.

Create **lists** for all data types.

- E1currentArrays
- E2currentArrays
- cupA_LcurrentArrays
- cupA_McurrentArrays
- cupB_LcurrentArrays
- cupB_McurrentArrays
- cupC_LcurrentArrays
- cupC_McurrentArrays
- cupD_LcurrentArrays
- cupD_McurrentArrays
- E1times
- E2times
- Ltimes
- Mtimes

Use while loop to walk through text file for important information

```
While not at end of File Do
    FirstLine                    Each loop through the first line is time and status info for data
    GarbageLine0                 There are three lines that we do not need and will store in
    GarbageLine1                 temporary variables to be thrown out
    GarbageLine2

    newFirstLine=FirstLine[17:-1]     remove first sixteen chars “OKNTCUR AT TIME”
    splitFirstLine               split first line so that time is first element and status is second
    time = splitFirstLine[0]     extract time info from First line
    status = splitFirstLine[1]   extract status info and remove spaces

    mode=status[0]               first character of status tells us the mode
        First Character          Mode
        (blank space)            L
        1                        L
        2                        L
        3                        L
        4                        E1
        5                        E1
        6                        E1
        7                        E1
        8                        E2
```

9	E2
A	E2
B	E2
C	M
D	M
E	M
F	M

```
Use IF statement to check first character for mode
    IF 1, 2, 3, or ‘’                           mode is L
        Ltimes.add, time                        add time to list
        dummyCurrentA=dblarr(16)                create double array of size 16 to hold currents
        dummyCurrentB=dblarr(16)
        dummyCurrentC=dblarr(16)
        dummyCurrentD=dblarr(16)

         dummyCurrentA                          read line of data and fill dummyCurrent
        garbage line                            read and discard “garbage line”
         dummyCurrentB
        garbage line
         dummyCurrentC
        garbage line
         dummyCurrentD
        For I = 0, 15 do        for loop for each current value
            Check that each value is greater than 1. If not, set to 1. Does A, B, C, and D cups

        cupA_LcurrentArrays.add, dummyCurrentA          add array data to final list
        cupB_LcurrentArrays.add, dummyCurrentB
        cupC_LcurrentArrays.add, dummyCurrentC
        cupD_LcurrentArrays.add, dummyCurrentD

Use IF statement to check first character for mode
    IF 4, 5, 6, or 7                            mode is E1
        E1times.add, time                       add time to list
        dummyCurrent=dblarr(16)                 create double array of size 16 to hold currents
                                                E1 is a single set of 16 current values

         dummyCurrent                           read line of data and fill dummyCurrent

        For I = 0, 15 do        for loop for each current value
            Check that each value is greater than 1. If not, set to 1. Does E cup

        E1currentArrays.add, dummyCurrent           add array data to final list

Use IF statement to check first character for mode
    IF 8, 9, A, or B                            mode is E2
        E2times.add, time                       add time to list
        dummyCurrent=dblarr(16)                 create double array of size 16 to hold currents
                                                E2 is a single set of 16 current values
        dummyCurrent                          read line of data and fill dummyCurrent
        dummyCurrent = dummyCurrent[4:*]         First four values are no good

        For I = 0, 11 do          for loop for each current value
            Check that each value is greater than 1. If not, set to 1. Does E cup

        E2currentArrays.add, dummyCurrent          add array data to final list

Use IF statement to check first character for mode
    IF C, D, E, or F                          mode is M
        Mtimes.add, time                      add time to list
        dummyCurrentA =dblarr(128)            create double array of size 16 to hold currents
        dummyCurrentB=dblarr(128)
        dummyCurrentC=dblarr(128)
        dummyCurrentD=dblarr(128)

        dummyCurrentA                         read line of data and fill dummyCurrent
        garbage line                          read and discard "garbage line"
        dummyCurrentB
        garbage line
        dummyCurrentC
        garbage line
        dummyCurrentD
        For I = 0, 127 do         for loop for each current value
            Check that each value is greater than 1. If not, set to 1. Does for A, B, C, and D

        cupA_McurrentArrays.add, dummyCurrentA          add array data to final list
        cupB_McurrentArrays.add, dummyCurrentB
        cupC_McurrentArrays.add, dummyCurrentC
        cupD_McurrentArrays.add, dummyCurrentD

LtimesArray = Ltimes.toarray(type=STRING)                   convert list of times to strings for
MtimesArray = Mtimes.toarray(type=STRING)                   later use
E1timesArray = E1times.toarray(type=STRING)
E2timesArray = E2times.toarray(type=STRING)


cupA = {L: cupA_LcurrentArrays, M: cupA_McurrentArrays}      structure to hold L & M mode data
cupB = {L: cupB_LcurrentArrays, M: cupB_McurrentArrays}
cupC = {L: cupC_LcurrentArrays, M: cupC_McurrentArrays}
cupD = {L: cupD_LcurrentArrays, M: cupD_McurrentArrays}
Emodes = {E1: E1currentArrays, E2: E2currentArrays}             structure to hold E1 & E2 mode data
times = {L: LtimesArray, M: MtimesArray, E1: E1timesArray, E2: E2timesArray}        time structure

data = {cupA:cupA, cupB:cupB, cupC:cupC, cupD:cupD, Emodes:Emodes, times:times}  big structure

return, data
End
```