

### CHALLENGES

1. Create a data file called `friends.dat` using any text-based editor and enter at least three records storing your friends' first and last names. Make sure that each field in the record is separated by a white space.
2. Using the `friends.dat` file from challenge number one, build another program that uses the `fscanf()` function for reading each record and printing field information to standard output until the end-of-file is reached. Include an error-handling routine that notifies the user of any system errors and exits the program.
3. Create a program that uses a menu with options to enter student information (name, ID, GPA), print student information, or quit the program. Use data files and `FILE` pointers to store and print information entered.
4. Modify the Phone Book program to allow the user to enter multiple entries without quitting the program.
5. Continue to modify the Phone Book program to allow a user to modify or delete phone book entries.

### CHALLENGES

1. Create a structure called `car` with the following members:

- `make`
- `model`
- `year`
- `miles`

Create an instance of the `car` structure named `myCar` and assign data to each of the members. Print the contents of each member to standard output using the `printf()` function.

2. Using the `car` structure from challenge number one, create a structure array with three elements named `myCars`. Populate each structure in the array with your favorite car model information. Use a `for` loop to print each structure detail in the array.
3. Create a program that uses a structure array to hold contact information for your friends. The program should allow the user to enter up to five friends and print the phone book's current entries. Create functions to add entries in the phone book and to print valid phone book entries. Do not display phone book entries that are invalid or `NULL (0)`.

5. save your programs in a folder called `Unit_6`. Zip it. Email to [eagenest@madison.k12.wi.us](mailto:eagenest@madison.k12.wi.us) by \_\_\_\_\_ . One submission per pair of partners