Things I learned in A2MS4:

Firstly, I learned that what pass by address in a function means and what are its benefits. Pass by address in a function means that we are passing a particular variable’s address into a function declaration, where a pointer is declared which will directly point to the variable’s address, and ultimately has access to the variables’ value that is passed as a function argument. If we do not want that function to change the value of the variable passed, we declare the pointer as const, so that any changes to the value of the variable that it is pointing to is prohibited. One major benefit of pass by address is that let us say, we want to return more than one value in a function to the caller of the function, we can do so with the help of pointer concept. Since the pointer points to an address of variable, as changes are made to its value, they will be reflected onto the original variable and the value of the variable will be updated accordingly, as long as the pointer is not const. In this way we can return as many values as possible, regardless of one return statement constraint in c language.

Secondly, I learned, sorting algorithm and how to sort a string in ascending order. The sorting algorithm’s primary functionality is to sort an array of numbers in ascending order. It can also be done in descending order, but this assignment focuses on sorting the numbers ascendingly. If we want to sort an array of let us say integer number we can do so easily, however, sorting a string is tricky. To solve this problem, we can use a very helpful function in string library called strcmp, which compares two strings and returns 0 if identical, a negative value if the first string is smaller than second string, according to the ASCII’s collating sequence & a positive value if the first string is bigger than second string, according to the ASCII’s collating sequence. The main condition of the function in A2MS4, which the whole sorting algorithm is dependant on is, if strcmp returns a value greater than 0, that means swapping is necessary since the string at previous index is bigger in number than the string in next index.

Thirdly, I learnt what are string library functions and its applications. In A2MS4, strlen and strcmp are widely used and has made it easier for me to code. strlen returns the index of the null byte in a string whereas, strcmp returns 0 if identical, a negative value if the first string is smaller than second string, according to the ASCII’s collating sequence & a positive value if the first string is bigger than second string, according to the ASCII’s collating sequence. An example of the application of strlen function in this assignment would be, displayContact function. In this function I had to print the middle initial only if that array is populated. Since, strlen returns the index of the null byte in a c string, the function helped me to print the middle initial only if the value returned by strlen is greater than 0. This way I can check if a c string is empty. If it is empty, then do not print the middle initial. An example of the application of strcmp function in this assignment would be, sortContacts function as descried in the previous paragraph about sorting a string in ascending order.

How findContactIndex and getTenDigitPhone function help with overall maintenance and readability.

To begin with, let us talk about the functionality of these two functions. The findContactIndex function helps us find the index in the Contacts array. This function requires another parameter, cellNum to match at what index does the string cellNum appear which is entered by the user. It uses strcmp function to match whether both the strings are identical, if they are, then return the index found otherwise return a different value. On the other hand, getTenDigitPhone function helps us with validation if the user entered a correct cell number. It checks for two main conditions, one, if the string entered is exactly 10 in length and two, if there are any characters other than 0 to 9 is the string entered by the user. These two functions are used in widely used throughout the program, for example in searchContacts, addContact, updateContact and deleteContact.

An example of application of findContactIndex & getTenDigitPhone function is seen in searchContacts function. searchContacts’ primary goal is to search and find a contact that the user entered that matches with a contact in the database. A user is asked to enter a ten-digit number with the help of getTenDigitPhone function which makes sure that the number entered is a valid string. Once a valid string is received, that string is passed onto the findContactIndex function to check whether the number entered by the user matches with the numbers in database. If the number matches with the numbers in the database, findContactIndex will return that index and then searchContacts will display the contact at that index with the help of displayContact function.

Since our program is totally based on cell phone number and to match the cell number entered by the user with the one in the database and perform various functionalities like search a contact, add a contact, delete a contact and update a contact, it is very critical for us as a programmer to make sure that user enters a valid cell phone number. After that, we must check if the number entered by the user is present in database as required by functions like searchContacts so that we can provide the user with the matched number and then it is up to them if they want to delete or update a contact.

In this way these two functions provide us with code maintenance and readability since most of the functions in the program rely on these two functions. If we were to do some changes in the code regarding the cell phone number, we only have to do that in these two functions which avoids unnecessary breakage of code and also helps us with clarity in reading the code. Once we call these functions and we pass the right arguments to these functions, they will perform their tasks reliably and provide us the result that we need.

Why is load/save a piece of information from/to a text file is important?

Firstly, text files help store/update the data permanently until that file is not deleted or corrupted. To give you an example, let us say we are not using the text file, rather we are using the array declared in the program whose size is 50 and we run the program to update information of 5 contacts, 45-50. Obtaining all the information of those 5 people is impossible after we terminate the program because after termination when the program runs again it will not have those information held at the respective indexes since the memory used at the time of execution of code is RAM which is a volatile storage. Whereas, when the data is stored into a specific text file, the data will be stored successfully and the next time we run the program, the value of those variables would not have been lost since text file writes the data onto HDD/SSD, which is a non-volatile storage. Apart from that, if we were to use this information stored in a file to another program, we can do that easily by importing the contents of this file into the newer program and now we have access to the same 50 contacts from the original program and perform newer functionalities like data analysis for number of people living in a certain city whose age is between 25 and 40.

Secondly, we can use text files to input the data that is not already connected to the program. Let us say, we have 50 more contacts to be added to the program. This functionality can be easily added to the program by importing the file that is updated with a total of 100 contacts rather than hard coding the values into a source code or adding those values to the program for one time only at the time of program execution rather than adding those values permanently only to lose that information after the program is terminated. This also helps in modifying the data efficiently rather than updating the data into a source code which makes the code hard to read, understand and debug. Also, hard coding values will make the look program lengthier since numerous lines are taken by those values in the source code.

By using text files for data input or output, modifying data would become very easy, since we can update the data straight into a text file and the program will keep a track of the variables updated, along with its values reflected into the text file, if they were to be updated.

Basically, using text files helps us separate the source code with the data stored at the variables used in a contacts program if the data is relatively larger. Thus, for these reasons it is more feasible to write the data on a text file and import that file into a program and use it to update the information as well.