By completing the last two milestones, I learned how to use functions and declare their prototypes so that each function can be used to receive the contact’s name, address, and phone numbers respectively. I also learned not only how to use a struct to hold the user-defined types such as Name, Address, and Numbers but also how to pass its member by address when calling the related function.

The reason why I used a struct named Contact to include three user-defined structs as data members is that we can store the contact’s information in a single object. In this way, the code can be more readable, condensed, and easily maintainable. For example, when I wanted to display the first name of the contact, I was simply able to access the information using a variable named contact.name.firstName.

It would be a very poor design if a pointer to type Contact is used as an argument when calling the functions, getName, getAddress, and getNumbers. By doing so, the functions would have access to the addresses of all members and, thus, have unnecessary information to perform their tasks. This extra information may cause serious problems when the function caused accidental changes to those values. Therefore, by limiting the amount of accessible information to the functions, each function can be highly cohesive and exhibit low coupling.