Reflection

The code was approached using the black box method of testing. A table of inputs and outputs was created to observe the errors and incorrect outputs. Starting off the very first issue that I notice that occurred is too much being written into the char variable “yesNo”. To take the input “%s%\*c” was used as a place holder. The issue with this is that %s will take an input until the null and includes the terminator \0. Normally the %\*c is used to ignore the null however since the %s is used it will be stored into the char variable which will cause overflow of the char variable that contains y or n with ‘\0’. To correct this %c%\*c is used to take the y or n and ignore the \0. In the second test the program ran however the following incorrect outputs occurred:

1. Middle initial: “N. O.Flint Rubble” instead of “N. O.”
2. Did not ask for apartment number and home number even though “y” was inputted

I approached these issues by placing a break point before the inputs of yesNo for the conditions and at the condition checks for the variables. The first issue was due to the size of “middleInitial” in the name struct being only 5. The input N. O. is 5 chars however this does not include the terminator that causes the overflow and to solve this the size was increased. The next issue was solved by placing an extra printf before the do While loop and observed the failed condition check. The problem with the check is the use of “ “ instead of ‘ ‘. The final issue was in a similar manner however was due to an and condition being used for checking for if “y” or “Y” was inputted.

There were several key topics that were introduced in this assignment. The first is the use of %\*c when getting input for a character. This is done as mentioned by the first error so that when entering a single character, the enter to finish the input is ignored. In the case of multiple characters, %[^\n] was used in the stead of %s. This is due to %s requiring a ‘\0’ or null terminator to complete the string. In the case of a scanf this would mean the user would need to manually enter the null which is not ideal. By using %[^\n] this will take input until next line which is the enter input.

When using the struct, an embedded struct that is a struct within a struct. This will use the embedded struct in a similar fashion to how other types such as an int being allocated 4 bytes the compiler will define based on the struct initialization.

Finally, within the code, one repetitive task that was done is the condition check for yes or no. One way this could have been done is to pass the char to a function as a parameter. This will read and convert to a single true or false and return to an if statement as a 1 or 0 (true or false).