

Programming Assignment #3

Write a C program that prompts the user to enter some information about up to ten individuals. It should store this information in a **structure**. The program should obtain the information from the structure, and output it in mailing label format as shown in sample run below.

Your program should include a **structure** with a tag name of: “**information**”. It should contain the following data as members:

a character array to store person's name, defined as:	full_name[35]
a character array to store person's street address, defined as:	address[50]
a character array to store person's city, defined as:	city[25]
a character array to store person's state, defined as:	state[3]
a long integer variable to store person's zip code, defined as:	zip_code

You need to define an array of type: **struct information**.

You can call this array: **person[10]**.

The dialog with the user must be as follows:

Welcome to the mailing label generator program.

How many people do you want to generate labels for (0-10)? 2

Enter name: Minnie Mouse
Enter street address: 100 Disney Drive
Enter city: Orlando
Enter state: FL
Enter zip code: 99990

Enter name: Big Bird
Enter street address: 10 Sesame Street
Enter city: Funtown
Enter state: MA
Enter zip code: 01222

Below are your mailing labels:

Minnie Mouse
100 Disney Drive
Orlando, FL 99990

Big Bird
10 Sesame Street
Funtown, MA 01222

Note: The **blue** text represents the "output" from your program and is shown for clarity only here. It is not required that your output be in this color. (Do not even attempt to try!). Also note that what the user types in is indicated in bold above. Again, for clarity only. Data validation/error checking is described below.

Hints/other requirements:

- Use **gets** (or `safer_gets()`) instead of **scanf** or **gets()** to read in more than one word at a time. (Needed for "name", "address" and "city", in this program.)
- Structure member **state** is a 3 element array, not 2. In this field you will be storing a 2 character abbreviation of a state. We need to define the extra (3rd) element to hold the null terminator character. (An optional feature could be to make it capital letters)
- You should use **%di** (instead of **%li**) as the format specifier for the "zip code" because if you enter your zip code starting with the number 0 (zero), C will interpret that number as "octal", which will cause erroneous results. For this reason, I **recommend** that you use **%ld** in the `scanf` statement when prompting the user for the zip code.
- The above hint handles the zip code as it is being *input* using a **scanf** statement. You have a different problem when you want to *output* the zip code using a **printf** statement. **C does not store leading zeros**. So, if the user in fact enters a zip code starting with a zero, you need to do some extra formatting when you *output* the zip code to display leading zeros if any. We have not yet covered this, so I will give you the solution. You have 2 choices: Either: **%.5ld** or **%05ld**. Both of those output format specifiers indicate that you want C to reserve 5 spaces for the output, and if the integer value is less than 5 digits, to pad the integer with leading zeros.
- Be sure to output the 2 character State in uppercase, even if the user enters it in lowercase.
- You do not need to use user-defined functions or string functions in this program (however, feel free to if you'd like to).
- You do not need to use pointers in this program. (That will be in another assignment!)
- Be sure to perform error checking on the "number of people" to enter (0-10). If 0 is entered, simply exit the program. Otherwise, re-prompt.
- Be sure to perform error checking on the zip code (00001 to 99999). If invalid, re-prompt.
- No other data validation on user input is needed.

Good luck!