Please Write In C-Free Software Or Write It Out!

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 as shown in sample run below.

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

a character array to store person's name, defined as:                **name[30]**acharacter array to store person's street address, defined as: **address[30]**a character array to store person's city, defined as:                    **city[20]**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**an integer variable to store person's age, defined as:                **age**
a character variable to store person's gender, defined as:        **gender**

You need to define an array of type: **struct info**. You can call this array: **people[10]**.

*The dialog with the user must be as follows:*

**Whatever Title you want Here**

**Enter name: Minnie Mouse
Enter street address: 100 Disney Drive
Enter city: Orlando
Enter state: FL
Enter zip code: 99990
Enter age: 25
Enter gender (M or F): F**

**Enter name: Big Bird
Enter street address: 10 Sesame Street
Enter city: Funtown
Enter state: MA
Enter zip code: 01222
Enter age: 20
Enter gender (M or F): M**

**The information you entered is:**

**Minnie Mouse
100 Disney Drive
Orlando, FL 99990
She is 25 years old.**

**Big Bird
10 Sesame Street
Funtown, MA 01222
He is 20 years old.**

**Note**: The **coral** 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 by the **blue** area above. Again, for clarity only.

**Hints/other requirements**:

* Use **gets** (or safer\_gets() ) instead of **scanf** to read in more than one word at a time. (Needed for "name" , "address" and "city", in this program.)

* Call **fflush(stdin);**   after any **scanf** (or **getchar**) for **gender**. This will remove any extraneous characters (like carriage returns) from the input buffer.

* 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.

* If you use **%li** (instead of **%ld**) as the format specifier for the "zip code" and if you enter your zip code starting with the number 0 (zero), C will interpret that number as "octal" and which will cause erroneous results. The **%i** format specifier says to interpret any number starting with a zero as *octal*. 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. You have 2 choices: Either: **%.5ld** or **%05ld**. Both of those 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.

* You do not need to use functions in this program (however, feel free to if you'd like).

* You do not need to use pointers in this program.
* Gender should be entered as either **M** for male, or **F** for female. Your program should test for "M"ale or "F"emale gender to determine whether to output the age as: He is... or She is... (refer to sample output below for more info.) Remember to test for both uppercase, and lowercase values of M and F. For example, the user may enter "m" or "M" as the gender, in either case, your program should interpret it as male. (***If neither M, m, F, or f is entered for gender, your program should not display any age information***.)

* You need to perform error checking on only the age user input (between 1 and 120, hey who knows?).