Lab #5

CS-2050 - Section D

Week of February 22, 2021

1 Requirements

This lab is intended to test your ability to work with structures and interface functions. You will not be provided with a main file in your starter code, and any testing code you produce will not be graded in this lab. In this lab, you will produce 6 *interface functions* (3 getters and 3 setters). All of your other required functions must use these interface functions in order for you to receive full credit, and **must not** directly access individual structs.

```
typedef struct {
        float squareFeet;
        int baths;
        short houseNumber;
} House;
```

1.1 Getters and Setters

> **Info:** You must write one getter and one setter function for each member of the struct type defined in your starter code. Unlike the prelab, these functions should treat the struct pointer passed to them as pointing to a single struct, and not a struct array.

1.2 printHouseArray

```
void printHouseArray(House *houses);
```

> **Info:** This function takes an array of **House** structs, and prints it out with each struct on a newline, and struct members labeled and separated by commas, like so:
>
> ```
> F2: 450.58, Baths: 2, HN: 144
> F2: 152.08, Baths: 1, HN: 772
> ```

1.3 loadHousesFromFile

```
House* loadHousesFromFile(const char *filename);
```

> **Info:** This function will read **House** structs in from a file, and store them in a dynamically allocated array which saves the size of the array in the pre-index location as with last lab. You are encouraged to reuse your createArray(), freeArray() and getArraySize() functions, but is not required. The first number in the file should be interpreted as the number of structs in the file, and the struct members are stored in the same order as they appear in the struct definition, separated by commas. Depending on how you write the function, this format specifier may be useful for you:
>
> ```
> "%f, %d, %hd"
> ```

1.4 getHouseWithMostBaths

```
House* getHouseWithMostBaths(House *houses);
```

Info: This function takes an array of **House** structs, and returns a pointer to the *House with the most baths*.

1.5 getHouseByHN

```
House* getHouseByHN(House *houses, short HN);
```

Info: This function takes an array of **House** structs, and returns a pointer to the *House with the specified house number*.

1.6 getAverageNumberOfBaths

```
float getAverageNumberOfBaths(House *houses);
```

Info: This function takes an array of **House** structs, and returns *the average number of baths in the array*.

2 Notice

Grading: Total 25 points

1. Write required *getter and setter* functions
   - * 6 points *total*
2. Write required *print array* function
   - * 3 points
3. Write required *load from file* function
   - * 8 points
4. Write required *getHouseWithMostBaths* function
   - * 3 points
5. Write required *getHouseByHN* function
   - * 2 points
6. Write required *getAverageNumberOfBaths* function
   - * 3 points

Notice:

1. All of your lab submissions must compile under GCC using the $-Wall$ and $-Werror$ flags to be considered for a grade.

2. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab policy document.