Objective: Work with pointers and doubly linked lists.

C++ has an STL called map (also called dictionary). It stores a key-value pair. In this assignment, you are going to design your own map class. This map class is very simple. It's key-value pairs are all strings.

Assignment:
Design a class called **Dictionary** that works like a dictionary/map. You only need to create a single file "dictionary.h" that includes all the code for your dictionary class. The file "dictionary.h" may include other header files. The class should keep track of a key-value pairs. All the keys must be unique. For a key, uppercase and lowercase letters should be treated the same. Your class should include a single default constructor that initializes an empty dictionary. The class should have the following functions (Do not change the names of the functions):

- **get** – returns the value stored at specified key. If not found return the string `"NOT_FOUND"`
- **put** – add a key-value pair to the dictionary. If it already exists, it should replace its value
- **keys** – returns all the keys in the dictionary in a *vector* of type string. They should be in the same order as the entries in the dictionary.
- **values** – returns all the values in the dictionary in a *vector* of type string. They should be in the same order as the entries in the dictionary.
- **hasKey** – returns true if a key is in the dictionary
- **hasValue** – returns true if a value is in the dictionary
- **size** – returns the number of key-value pairs in the dictionary
- **removeKey** – removes a node from the list that matches a key
- **removeValues** – removes all the matched values from the dictionary

The dictionary must implement with an ordered doubly linked list of key-value pair. The linked list must always be sorted by key.

Do not use any STL for this assignment other than the vectors returned in functions **keys** and **values**.

Write a main program to test all the above functions. You are also provided with unit tests to test these functions. All the functions must be implemented to run the unit tests.
A green checkmark on GitHub indicates that all the unit test have passed correctly. You may run the unit tests locally using the makefile provided:

```
make run_tests
```

Add a rule to the provided **Makefile** (at the top) to compile your program with a sample main program.

Hints:

- Start early
- Design the doubly linked list first with all the necessary functionalities.
- Test your doubly linked list before implementing the dictionary
- Always work with properly formatted code. To format your code in VS Code use the command (shift-Alt-F) or (shift-option-F)

Grading:

Programs that contain syntax errors will earn zero points.
Programs that use global variables, other than constants, will earn zero points.

- 5 points – get function
- 10 points – put function
- 5 points – size function
- 10 points – keys function
- 10 points – values function
- 5 points – hasKey
- 5 points – hasValue
- 10 points – removeKey
- 10 points – remove values
- 5 points – documentations and style.
  Follow the coding style outline on the class's website on GitHub:
  https://github.com/nasseef/cs3610/blob/master/docs/coding-style.md

Submission:

Submit a link to your repository on Blackboard before the deadline.