HOMEWORK 5 - DATA STRUCTURES AND FUNCTIONS (PART II)
CHALLENGING VERSION

Part I

Please complete all 6 in-class exercises at the end of the week 5 notes. Place the relevant source code in a file named `hw5_1.py` which should be in your home directory under week5 (home directory > python > week5).

Part II

Write a function that stores employee data in a dictionary of dictionaries called `employees`. Each individual employee dictionary should hold the employee's name and title. The key to a particular employee's dictionary in the `employees` dictionary should be the employee's ldap.

`employees` should look something like this at the end:

```
employees = {'alberthwang': {'name': 'Albert Hwang',
                             'title': 'Developer'},
             'satishm': {'name': 'Satish Musurunu',
                         'title': 'Tech Lead'}}
```

Write a subroutine to fill the `employees` dictionary with employee data. Here are the specifications:

- First the user should be prompted with the option to Add or Quit
- If the user wants to add, prompt the user for an employee's ldap, name, and title
- The `employees` dictionary should hold at most 5 employees (but can hold fewer).
- The same employee (ldap) should not be allowed to be added twice (but trying to do so should not stop execution of the script)
- Once the user quits or the `employees` dictionary fills up, it should exit and print all the records

Here is sample output:

```
Add/Quit - add
Employee LDAP: alberthwang
Employee Name: Albert Hwang
Employee Title: Black Smith
alberthwang has been added successfully

Add/Quit - add
Employee LDAP: alberthwang
Employee Name: Albert Hwang
Employee Title: Developer
alberthwang already has a record!

Add/Quit - add
Employee LDAP: smadaan
Employee Name: Saurabh Madaan
Employee Title: Prime Minister
```

```
smadaan has been added successfully

Add/Quit - quit

Here are the Employee Records!

alberthwang's name is 'Albert Hwang' and he/she is a Black Smith
smadaan's name is 'Saurabh Madaan' and he/she is a Prime Minister
```

Part III

Write a program that asks the user to enter a variable number of integers (just like in HW #4). Each integer should be stored in a list called `data`. Once `data` is created, please write a function that **sorts** this list of integers (ascending). Here are the caveats:

- You must write the sorting algorithm yourself
- You cannot use any of the following - `.sort()`, `sorted()`, `max()`, `min()`
- Write the algorithm using only list methods covered in Weeks 4 and 5

Please complete Part I in a file named hw5_1.py , Part II in a file named hw5_2.py, Part III in a file named hw5_3.py. All three files should go inside your week5 directory (home directory > python > week5). I will look for them there. Be sure that the script follows all the style conventions.