For the work below, you should type your program in Sublime, and execute that program at the terminal (i.e., don't use the Python interpreter to enter code directly). All of your work below can go in the same source code file. Remember that function definitions should go at the top of your program, and calls to your functions should all go at the bottom in your “main scope”.

1. Write a non-fruitful function named `printMyName` that has no parameters and that simply prints your name. (You do not need to use a for loop for this.)
   
   Make sure to call your function in the “main scope”, and test your program for correctness before moving on.

2. Write a non-fruitful function named `printSomeName` that has one parameter called `name` and simply prints the name provided as the argument. (You do not need to use a for loop for this.)
   
   Make sure to call your function appropriately in the “main scope”, and test your program for correctness before moving on.

3. Write a non-fruitful function named `printName` that has one parameter called `name` and simply prints the name provided as the argument. For this function, use a for loop to print the characters in the name one at a time all on the same line. Include a separate empty print statement at the end of your function so that a new line will be printed once at the end only.
   
   Make sure to call your function appropriately in the “main scope”, and test your program for correctness before moving on.

4. Write a non-fruitful function named `printNameReversed` that has one parameter called `name`. For this function, use a for loop to print the characters in the name one at a time all on the same line, but in reverse order of that provided. Include a separate empty print statement at the end of your function so that a new line will be printed once at the end only.
   
   Make sure to call your function appropriately in the “main scope”, and test your program for correctness before moving on.

5. Write a non-fruitful function named `printNameUpper` that has one parameter called `name`. For this function, use a for loop to print the characters in `name` one at a time all on the same line, and capitalize (using our typical combination of the `chr()` and `ord()` functions) any lowercase letters. (Your function should not modify the output for any characters that are not lowercase letters.) Include a separate empty print statement at the end of your function so that a new line will be printed once at the end only.
   
   Make sure to call your function appropriately in the “main scope”, and test your program for correctness before moving on.

6. Write a fruitful function named `convertCharToUpper` that has one parameter called `theChar`. This function should return an uppercase version (again, using `chr()` and `ord`) of `theChar` if `theChar` is a lowercase alphabetic character. Otherwise, the function should simply return the given character as is.
   
   When calling your function in the “main scope”, remember that you'll need to print the return value there — i.e., this new function is returning (not printing) a value, so your call in main will need to print the returned value for you to see the result when testing.
7. Write a fruitful function named `convertNameToUpper` that has one parameter called `name`. This function should return an uppercase version of the name, where only lowercase letters are changed.

Hints:

- You should use a for loop to walk through the characters of `name`, and use your `convertCharToUpper` function from above to handle the conversion to uppercase for you.
- Because you'll be using `convertCharToUpper`, your for loop here won't need an if statement.
- You should create a new variable before your for loop initialized to the empty string "".
- Inside your for loop, you can append to your variable using the plus operator on strings, as in:
  
  ```
  newName = newName + convertCharToUpper('x')
  ```

  (but obviously you shouldn't pass 'x' every time, but rather the appropriate character from `name`).
- At the end of your function, just return your new variable, which should now contain the name in appropriate uppercase form.

When calling your function in the "main scope", remember that you'll need to print the return value there — i.e., just like `convertCharToUpper`, this new function is returning (not printing) a value, so your call in main will need to print the returned value for you to see the result when testing.