

---

## CMSC 201 Fall 2016

### Homework 7 – More Functions

**Assignment:** Homework 7 – More Functions

**Due Date:** Wednesday, November 2nd, 2016 by 8:59:59 PM

**Value:** 40 points

**Collaboration:** For Homework 7, collaboration is allowed. Make sure to consult the syllabus about the details of what is and is not allowed when collaborating. You may not work with any students who are not taking CMSC 201 this semester. If you work with someone, remember to note their name, email address, and how you collaborated at the top of your file.

If you did not work with anyone else on this assignment, your collaboration statement should state that

**I did not collaborate with anyone on this assignment.**

If you did work with someone else (or multiple other people), your collaboration statement should state something *similar* to the following:

**I collaborated with Fox Mulder (fmulder1@umbc.edu); I helped him understand the loop.**

Make sure that you have a complete file header comment at the top of your file, and that all of the information is correctly filled out.

```
# File:      FILENAME.py
# Author:    YOUR NAME
# Date:      THE DATE
# Section:   YOUR DISCUSSION SECTION NUMBER
# E-mail:    YOUR_EMAIL@umbc.edu
# Description:
#   DESCRIPTION OF WHAT THE PROGRAM DOES
# Collaboration:
#   COLLABORATION STATEMENT GOES HERE
```

Homework 7 is designed to help you practice using all of the structures we have learned so far. You will also get even more practice with creating and calling functions.

Remember to enable Python 3 before running and testing your code:

```
scl enable python33 bash
```

## **Instructions**

In this homework, you will be doing a series of exercises designed to help you practice creating different functions, and calling them to perform tasks as needed. Each one of these exercises will be in a **single python file**. For this assignment, you may assume that all the input you get will be of the correct type (e.g., if you ask the user for a whole number, they will give you an integer).

**For this assignment, you'll need to follow the class coding standards**, a set of rules designed to make your code clear and readable. The class coding standards are on the website, linked at the top of the “Assignments” page. You can also access them directly via this URL (<http://goo.gl/yEoGfC>).

**At the end, your Homework 7 file must run without any errors.**

## **Details**

Homework 7 is broken up into seven separate parts. **Make sure to complete all 7 parts.**

Although Homework 7 is broken up into separate parts, you will be turning in a **single file** at the end, with all of your functions contained inside it. The single file should be called `hw7.py`.

**NOTE: Your filename for this homework must match the given one exactly.**

And remember, filenames are case sensitive!

## Questions

Each question is worth the indicated number of points. Following the coding standards, having complete file headers, and having the correctly named file is worth 4 points.

For Homework 7, you will be creating a **single file**, called `hw7.py`, that contains seven separate functions:

- `createIntList()`
- `getValidInt()`
- `printMenu()`
- `allTheSame()`
- `allDifferent()`
- `sorted()`
- `main()`

Each function is described in detail below. **Adhere to the coding standards by including function header comments for each of the functions** (other than `main`). Follow the instructions and example provided in the coding standards document when creating your function header comments. Failing to include function header comments will lose you points.

You are highly encouraged to test each function fully before moving on to the next. It is recommended not to complete the looping functionality of `main()` until all of your other functions work correctly.

Using a “temporary” `main()` to interact with each function as you develop and test it may prove useful.

Pay close attention to the sample output provided when testing your own code.

**Part1 - createIntList()**

**(Worth 2 points)**

The `createIntList()` function takes in a sentinel value (an integer), and returns a list of integers. The function prompts the user to create a list of integers, using the provided sentinel value as the number the user enters to exit the creation of the list. Sample output for this function can be found at the end of the assignment description.

*This function should behave identically to the one from your Homework 6 assignment. You can use the same code as you submitted for HW6.*

**Part2 - getValidInt()**

**(Worth 2 points)**

The `getValidInt()` function takes in a minimum and a maximum (integers), and returns an integer between those two (inclusive). The function should reprompt the user until they enter a choice between the minimum and maximum. Sample output for this function can be found at the end of the assignment description.

*This function should behave identically to the one you completed for the assignment in Lab 8. You can use the same code you wrote for Lab 8.*

**Part3 - printMenu()**

**(Worth 4 points)**

The `printMenu()` function takes in no parameters, and returns no value. It simply prints out the menu options to the user, so they know what choices are available to them. Sample output for this function can be found at the end of the assignment description.

NOTE: You should use constants for the menu choices! Also, `printMenu()` should not call `getValidInt()`; that should be handled separately by `main()`.

**Part4 - allTheSame()**

**(Worth 5 points)**

The `allTheSame()` function takes in a list, and returns a Boolean value based on the contents of the list. If all elements of the list are identical, the function returns `True`. If even one element is different, the function returns `False`. Sample output for this function can be found at the end of the assignment description.

NOTE: You are allowed to use `in` (the membership test) for this function.

**Part5 - allDifferent()**

**(Worth 6 points)**

The `allDifferent()` function takes in a list, and returns a Boolean value based on the contents of the list. If all elements of the list are different, the function returns `True`. If two elements (or more) are the same, the function returns `False`. Sample output for this function can be found at the end of the assignment description.

NOTE: You are allowed to use `in` (the membership test) for this function.

**Part6 - sorted()**

**(Worth 5 points)**

The `sorted()` function takes in a list, and returns a Boolean value based on the contents of the list. If all elements of the list are ordered from smallest to largest, the function returns `True`. If any elements are out of order (including if the list is ordered from largest to smallest), the function returns `False`. Sample output for this function can be found at the end of the assignment description.

NOTE: You are not allowed to use `sort()` for this function. Doing so will earn you no points for this function.

**Part7 - main()**

**(Worth 12 points)**

The final `main()` function uses the six functions that you have already written to implement a simple list information checker program. The program should:

- Present the user with a menu
- As long as the user does not choose to quit:
  - Ask the user for their menu choice
  - If they choose to create a list
    - Ask the user for their desired sentinel value
    - Create a lists of integers (using the given sentinel value)
  - If they choose to check if a list is all the same element
    - Check if the list is all the same element
    - Print out the list contents and the result
  - If they choose to check if a list is all different elements
    - Check if the list is all different elements
    - Print out the list contents and the result
  - If they choose to check if a list is sorted
    - Check if the list is sorted (from smallest to largest)
    - Print out the list contents and the result
  - Reprint the menu and reprompt the user for their menu choice
  - ❖ Continue until the user has chosen to quit the program

***(WARNING: It is not advised to attempt this part of the homework until you have completed and fully tested the first six parts.)***

See the following pages for sample output, with the user input in blue.

**We have also highlighted the results in orange**, to make it easier to pick out from all of the menu choices being printed out.

(Yours does not have to match this exactly, but it should be similar.)

```

bash-4.1$ python hw7.py
Welcome to the List Info Checker

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 2
The list [] is all the same element
-----

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 3
The list [] is all unique elements
-----

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 4
The list [] is sorted
-----

```

(Additional sample output on the next page.)

```

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 1
What do you want the sentinel to be? 0
Please enter a number, 0 to stop: 1
Please enter a number, 0 to stop: -9
Please enter a number, 0 to stop: 88
Please enter a number, 0 to stop: 2
Please enter a number, 0 to stop: 6
Please enter a number, 0 to stop: 0
-----

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 2
The list [1, -9, 88, 2, 6] is not all the same element
-----

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 3
The list [1, -9, 88, 2, 6] is all unique elements
-----

```

(Additional sample output on the next page.)



Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 4

The list [1, -9, 88, 2, 6] is not in order

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 1

What do you want the sentinel to be? 0

Please enter a number, 0 to stop: 9

Please enter a number, 0 to stop: 9

Please enter a number, 0 to stop: 9

Please enter a number, 0 to stop: 0

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 2

The list [9, 9, 9] is all the same element

(Additional sample output on the next page.)

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 4

The list [9, 9, 9] is sorted

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 1

What do you want the sentinel to be? -9

Please enter a number, -9 to stop: 7

Please enter a number, -9 to stop: 5

Please enter a number, -9 to stop: 3

Please enter a number, -9 to stop: -9

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 4

The list [7, 5, 3] is not in order

(Additional sample output on the next page.)

```

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 6
Please enter a number between 1 and 5 (inclusive): 0
Please enter a number between 1 and 5 (inclusive): -1
Please enter a number between 1 and 5 (inclusive): 11
Please enter a number between 1 and 5 (inclusive): 88
Please enter a number between 1 and 5 (inclusive): 1
What do you want the sentinel to be? -1
Please enter a number, -1 to stop: 1
Please enter a number, -1 to stop: 2
Please enter a number, -1 to stop: 3
Please enter a number, -1 to stop: 4
Please enter a number, -1 to stop: 5
Please enter a number, -1 to stop: 100
Please enter a number, -1 to stop: -1
-----

Please make a choice from the menu:
1 - Create a list
2 - Check if list is all same
3 - Check if list is all different
4 - Check if list is sorted
5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 2
The list [1, 2, 3, 4, 5, 100] is not all the same element
-----

```

(Additional sample output on the next page.)

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 4

The list [1, 2, 3, 4, 5, 100] is sorted

-----

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 3

The list [1, 2, 3, 4, 5, 100] is all unique elements

-----

Please make a choice from the menu:

- 1 - Create a list
- 2 - Check if list is all same
- 3 - Check if list is all different
- 4 - Check if list is sorted
- 5 - Exit the program

Please enter a number between 1 and 5 (inclusive): 5

Thank you for using the List Info Checker

---

## Submitting

Once your `hw7.py` file is complete, it is time to turn it in with the `submit` command. (You may turn in the file multiple times as you complete each function. To do so, simply submit `hw7.py` each time you complete a part of the homework. Each new `submit` will overwrite the old file.)

You must be logged into your GL account, and you must be in the same directory as your Homework 7 python file. To double-check this, you can type `ls`.

```
linux1[3]% ls
hw7.py
linux1[4]% █
```

To submit your Homework 7 python files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW7`. Type in (all on one line) `submit cs201 HW7 hw7.py` and press enter.

```
linux1[4]% submit cs201 HW7 hw7.py
Submitting hw7.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can check that your homework was submitted by following the directions in Homework 0. Double-check that you submitted your homework correctly, since **an empty file will result in a grade of zero for this assignment.**