CMSC 201 Fall 2016

Homework 6 – Functions

Assignment: Homework 6 – Functions
Due Date: Wednesday, October 26th, 2016 by 8:59:59 PM
Value: 40 points

Collaboration: For Homework 6, collaboration is not allowed – you must work individually. You may still come to office hours for help, but you may not work with any other CMSC 201 students.

Make sure that you have a complete file header comment at the top of each 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 6 is designed to help you practice using all of the structures we have learned so far. You will also get significant 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 should be in a **separate 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 6 file must run without any errors.

Details

Homework 6 is broken up into four separate parts. **Make sure to complete all 4 parts.**

Although Homework 6 is broken up into four parts, you will be turning in a **single file** at the end, with all four of your functions contained inside it. The single file should be called **`hw6.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 6, you will be creating a **single file**, called **`hw6.py`**, that contains four separate functions:

- **`summation()`**
- **`multiply()`**
- **`createIntList()`**
- **`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. 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.

Part 1 – `summation()` (Worth 7 points)

The **`summation()`** function should take in a list of integers and return their sum. You **may not** use Python's built-in functions to calculate the sum. If the list of integers is empty, the sum should be 0.

Sample output for this function can be found at the end of the assignment description.

Part2 – `multiply()` (Worth 7 points)

The **`multiply()`** function should take in a list of integers and return their product (the number computed by multiplying all of the numbers together). You **may not** use Python's built-in functions to calculate the product. If the list of integers is empty, the product should be 0.

Sample output for this function can be found at the end of the assignment description.

Part3 – `createIntList()` (Worth 9 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.

Part4 – `main()` (Worth 13 points)

The final **`main()`** function uses the three functions that you have already written to implement a simple math helper program. The program should:

- Ask the user how many lists they want to create
- For each list:
  - Tell the user which number list they are creating (starting at 1)
  - Ask the user for their desired sentinel value
  - Create a lists of integers (using the given sentinel value)
  - Calculate the summation
  - Calculate the product
  - Print out:
    - The entire list
    - The summation of the list
    - The product of the list
  - ❖ Continue until the desired number of lists have been gathered, and their information has been printed to the user

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

See the following pages for sample output, with the user input in blue. (Yours does not have to match this exactly, but it should be similar.)

```
bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? 0
Thank you for using the Simple Math Helper



bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? 1

You are creating list #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: 2
Please enter a number, 0 to stop: 3
Please enter a number, 0 to stop: 4
Please enter a number, 0 to stop: 5
Please enter a number, 0 to stop: 6
Please enter a number, 0 to stop: 7
Please enter a number, 0 to stop: 8
Please enter a number, 0 to stop: 9
Please enter a number, 0 to stop: 10
Please enter a number, 0 to stop: -9
Please enter a number, 0 to stop: -3
Please enter a number, 0 to stop: -1
Please enter a number, 0 to stop: 0
For the list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -9, -3, -1]
The summation is 42
The product is -97977600
Thank you for using the Simple Math Helper
```

(Additional sample output on the next page.)

```
bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? 3

You are creating list #1
What do you want the sentinel to be? 0
Please enter a number, 0 to stop: 0
For the list []
The summation is 0
The product is 0

You are creating list #2
What do you want the sentinel to be? -1
Please enter a number, -1 to stop: 5
Please enter a number, -1 to stop: 6
Please enter a number, -1 to stop: 7
Please enter a number, -1 to stop: 8
Please enter a number, -1 to stop: 0
Please enter a number, -1 to stop: -2
Please enter a number, -1 to stop: -1
For the list [5, 6, 7, 8, 0, -2]
The summation is 24
The product is 0

You are creating list #3
What do you want the sentinel to be? -999
Please enter a number, -999 to stop: -88
Please enter a number, -999 to stop: 50
Please enter a number, -999 to stop: 2
Please enter a number, -999 to stop: 10
Please enter a number, -999 to stop: 1
Please enter a number, -999 to stop: 3
Please enter a number, -999 to stop: 4
Please enter a number, -999 to stop: -999
For the list [-88, 50, 2, 10, 1, 3, 4]
The summation is -18
The product is -1056000
Thank you for using the Simple Math Helper
```

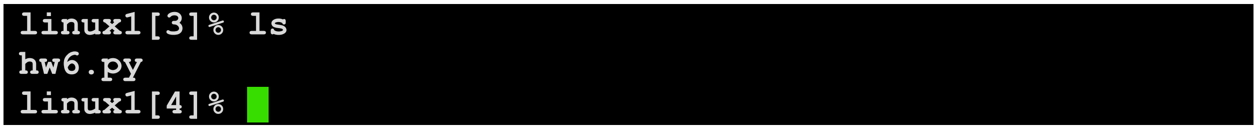
(Additional sample output on the next page.)

```
bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? 1

You are creating list #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: 0
For the list [1]
The summation is 1
The product is 1
Thank you for using the Simple Math Helper



bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? -1
Thank you for using the Simple Math Helper




bash-4.1$ python hw6.py
Welcome to the Simple Math Helper
How many lists would you like to create? -2
Thank you for using the Simple Math Helper
```

Submitting

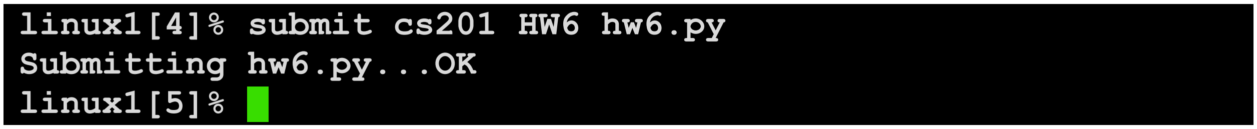
Once your **`hw6.py`** file is complete, it is time to turn it in with the **`submit`** command. (You may turn in the file multiple times as you complete each function. To do so, simply submit **`hw6.py`** each time you complete a part of the homework. Each new **`submit`** will overwrite the old file.)

You must be logged into your GL account, and you must be in the same directory as your Homework 6 python file. To double-check this, you can type **`ls`**.



```
linux1[3]% ls
hw6.py
linux1[4]%
```

To submit your Homework 6 python files, we use the **`submit`** command, where the class is **`cs201`**, and the assignment is **`HW6`**. Type in (all on one line) **`submit cs201 HW6 hw6.py`** and press enter.



```
linux1[4]% submit cs201 HW6 hw6.py
Submitting hw6.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.**