
CMSC 201 Spring 2016 Homework 8 – Recursion

Assignment: Homework 8 – Recursion
Due Date: Monday, April 25th, 2016 by 8:59:59 PM
Value: 40 points

Homework 8 is designed to help you practice using recursion to solve problems. (Remember, a recursive function must have at least one base case and at least one recursive case!)

Remember to enable Python 3 before you run your programs:
`/usr/bin/scl enable python33 bash`

Instructions

Each one of these exercises should be completed 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 Blackboard under “Course Documents” in a file titled “CMSC 201 - Python Coding Standards.”

You should be commenting your code, and using constants in your code (not magic numbers or strings). You should also have a function header comment for every function that is not `main()`!

Re-read the coding standards!

You will **lose major points** if you do not following the 201 coding standards.

A very important piece of following the coding standards is writing a complete **file header comment block**. Make sure that each file has a comment block at the top (see the coding standards document for an example).

NOTE: You must use `main()` in each of your files.

Details

Homework 8 is broken up into three parts. **Make sure to complete all 3 parts.**

Each part must use a recursive function to solve the given problem.

NOTE: Your filenames for this homework must match the given ones exactly.

And remember, filenames are case sensitive.

hw8_part1.py

(Worth 10 points)

For this part of the homework, you will write a recursive function that takes in a list of positive integers as a parameter, and prints out the maximum value in that list. Your recursive function should be called `maxInt()`.

The list of positive integers should be obtained by prompting the user to enter numbers, allowing them to use “-1” to stop adding integers to the list.

You may **NOT** use the `max()` function to calculate the maximum.

Here is some **sample output** for `hw8_part1.py`, with the user input in blue. Your output does not need to be identical, but should be similar.

```
bash-4.1$ python hw8_part1.py
Enter a number to append to the list, or -1 to stop: 15
Enter a number to append to the list, or -1 to stop: 20
Enter a number to append to the list, or -1 to stop: 13
Enter a number to append to the list, or -1 to stop: 30
Enter a number to append to the list, or -1 to stop: 22
Enter a number to append to the list, or -1 to stop: -1

The list you entered is: [15, 20, 13, 30, 22]
The maximum value in the list is: 30
```

hw8_part2.py

(Worth 12 points)

For this part of the homework, you will write a recursive function that prints a hollow square. Your recursive function should be called `hollowSquare()`, and must take in three arguments: an integer for the square height, the character the user has chosen to “draw” the square with, and a counter variable to keep track of “where” you are when drawing the square.

Your program should prompt the user for these inputs, **in exactly this order**:

1. The height of their square
2. The symbol the square will be made of

You cannot assume that the provided height will be a valid value! The user will always provide an integer, but you must perform basic **input validation** to ensure that the height you accept is greater than or equal to 1. Remember to prompt the user with what you will accept as valid input. (Don’t just tell them their choice is incorrect; tell them what the acceptable values are as well.)

Here is some sample output, with the user input in blue.

```
bash-4.1$ python hw8_part2.py
Please enter the height of your square: -3
Please enter the height of your square (must be > 0): 0
Please enter the height of your square (must be > 0): 10
Please enter a character for your square: #

#####
#           #
#           #
#           #
#           #
#           #
#           #
#           #
#           #
#####
```

(Because characters are taller than they are wide, your “square” won’t appear to be square – this is fine!)

hw8_part3.py

(Worth 14 points)

Finally, you will be given an input file, from which you will build a list of all *unique* characters in that file, using a recursive function.

Your recursive function should be called `newChar()` and should take in *at least* two parameters: the string from the file and a list to store all of the unique characters.

You will not be asking the user for the name of an input file. It will always be named `input.txt`, and should be able to contain any type of text.

Your program should print out the list of all unique characters. (Hint: you can achieve this by using `print(listNameGoesHere)` in your code.)

Here is the sample output for `hw8_part3.py`.

```
bash-4.1$ python hw8_part3.py
['T', 'h', 'e', ' ', 'q', 'u', 'i', 'c', 'k', 'b', 'r',
'o', 'w', 'n', 'f', 'x', 'j', 'm', 'p', 'd', 'v', 't',
'l', 'a', 'z', 'y', 'g', '"', 's', '.', '\\n', "'", 'F']
```

Here is text in the `input.txt` file used to create the above sample output:

```
The quick brown fox jumped over the lazy dog's tail.
"For realz."
```

Submitting

Once all three parts of your Homework 8 are complete, it is time to turn them in with the `submit` command.

Don't forget to complete the header block comment for each file! Make sure that you updated the header block's file name and description for each file.

You must be logged into your GL account, and you must be in the same directory as the Homework 8 files. To double check this, you can type `ls`.

```
linux1[3]% ls
hw8_part1.py hw8_part2.py hw8_part3.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW8`. Type in (all on one line) `submit cs201 HW8 hw8_part1.py hw8_part2.py hw8_part3.py` and press enter.

```
linux1[4]% submit cs201 HW8 hw8_part1.py hw8_part2.py
hw8_part3.py
Submitting hw8_part1.py...OK
Submitting hw8_part2.py...OK
Submitting hw8_part3.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 **double-check that all three homework files were submitted** by using the `submitls` command. Type in `submitls cs201 HW8` and hit enter.

And you're done!