

Homework 6: Recursion

Assignment Due on 12/10/2016 at 11:59PM [HARD DEADLINE]

Academic Honesty Policy:

You are expected to submit **your own work**. Assignments are to be completed individually, NOT in groups. No collaboration is permitted, unless otherwise specified. Please do not include any of your code snippets or algorithms in public Piazza posts. You cannot not use solutions from any external source. If you have any questions at all about this policy please ask the course staff before submitting an assignment.

For this, the last homework, we will not have any main Piazza post. You may simply post questions and tag them appropriately.

Please note that this is an optional homework; you can only earn points here, not lose them. However, you will not be able to use any late hours on this assignment.

For Section A:

Please write your answers to each question in a file named `short_answers.txt` and include it in your submission folder.

For Section B:

- We will not be providing skeletons for this homework. It is up to you to design functions that fulfill the requirements stated in each problem. Make sure that you name your functions properly, save them in files whose names are identical to the functions they contain (i.e. `summation.py`).
- Make a comment at the top of each file explaining the high-level logic of your code. Make in-line comments detailing specific decisions and logic where necessary.
- You will be graded on your code style. Please refer to the Python style guide available on Courseworks in the resources folder.
- Your submission should contain the following files:
 - `short_answers.txt`
 - `summation.py`
 - `is_prime_recursive.py`

All code should be written in python 3.5. When submitting, put all files into a single folder named `UNI_hw6`. Compress your `UNI_hw6` folder into a zip file named `UNI_hw6.zip` (eg. `abc1234_hw6.zip`). Make sure to use an underline, NOT a hyphen. Submit that zip file via courseworks.

Section A: Short Answers

Write your answers to these questions in `short_answers.txt`

1. What will `fun(25)` print? What is the significance of this/what does this represent? (2 points)

```
def fun(n):  
    if (n == 0):  
        return  
    print(n%2)  
    fun(n//2)
```

2. What does the following function do? (assume `x` and `y` are integers). Give a short description of the operation this function performs on `x` and `y`. (3 points)

```
def mystery(x, y):  
    if (y == 0):  
        return 0;  
    else:  
        return (x + mystery(x, y-1))
```

Implement the following functions as recursive functions. That means that you may not use any loops (`while`, `for`).

1)

Implement the following summation with a recursive function ***summation(n)***:

$$\sum_{k=1}^n k^3 = 1^3 + 2^3 + 3^3 + \dots + n^3$$

Summation should take a positive integer `n` and return the value of the summation for that `n` as defined above. (8 points)

2)

Write a function, ***is_prime_recursive***, which takes some integer `n` as input and returns `True` if `n` is prime and `False` otherwise. (12 points)

NOTE: since we are not providing skeleton code, you may want to add additional function parameters as necessary. Make sure, in your comments, that you clarify exactly what each of your function parameters represent (i.e. which one is `n`, the purpose of the additional parameters) and how to properly call your function on some value of `n`. Feel free to write your own test cases to demonstrate what your function returns for different values of `n` (i.e. -4, 1, 7, 25).