

Carleton University
Department of Systems and Computer Engineering
SYSC 2100 — Algorithms and Data Structures — Winter 2021

Lab 6 - Recursive Functions

Submitting Lab Work for Grading

Remember, you don't have to finish the lab by the end of your lab period. The deadlines for submitting solutions to cuLearn for grading are listed in the *Wrap Up* section at the end of this handout. Solutions that are emailed to your instructor or a TA will not be graded, even if they are emailed before the deadline.

Please read *Important Considerations When Submitting Files to cuLearn*, on the last page of the course outline.

References

Problem Solving with Algorithms and Data Structures Using Python, Chapter 5, Sections 5.1 - 5.8.

Getting Started

Download `recursive_functions.py` from the *Lab Materials* section of the main cuLearn course page.

Exercise 1: Calculating x^n , where n is a nonnegative integer, can be specified using this recursive formulation:

$$x^0 = 1$$

$$x^n = x * x^{n-1}, n > 0$$

Read the docstring for function `power`. Replace the `raise` statement with a recursive implementation of the function. Your function cannot have any loops and it cannot use Python's `**` operator or call the `pow` function in Python's `math` module.

Use the shell (or write a short script) to test `power`. Remember to test the base case and some recursive cases.

Exercise 2: You're going to write a function that returns the number of digits in integer n , $n \geq 0$. If $n < 10$, it has one digit, which is n . Otherwise, it has one more digit than the integer $n // 10$. For example, 7 has one digit. 63 has two digits, which is one more digit than $63 // 10$ (which is 6). 492 has three digits, which is one more digit than $492 // 10$, which is 49.

Define a recursive formulation for `num_digits`. You'll need a formula for the recursive case and a formula for the stopping (base) case.

Read the docstring for function `num_digits`. Replace the `raise` statement with a recursive

implementation of the function. Your function cannot have any loops. Your function must be $O(n)$.

Use the shell (or write a short script) to test `num_digits`. Remember to test the base case and some recursive cases.

Exercise 3: Read the docstring for function `count`. Replace the `raise` statement with a recursive implementation of the function. Your function cannot have any loops. Your function must be $O(n)$.

Use the shell (or write a short script) to test `count`. Remember to test the base case and some recursive cases. To help you test your code, functions `build_linked_list` and `to_string` have been provided in `recursive_functions.py`.

Exercise 4: Read the docstring for function `last`. Replace the `raise` statement with a recursive implementation of the function. Your function cannot have any loops. Your function must be $O(n)$.

Use the shell (or write a short script) to test `last`. Remember to test the base case and some recursive cases.

Exercise 5: Read the docstring for function `copy`. Replace the `raise` statement with a recursive implementation of the function. Your function cannot have any loops. Your function must be $O(n)$.

Use the shell (or write a short script) to test `copy`. Remember to test the base case and some recursive cases.

Wrap Up

Please read *Important Considerations When Submitting Files to cuLearn*, on the last page of the course outline. The submission deadlines for this lab are:

Lab Section	Lab Date/Time	Submission Deadline (Ottawa Time)
L5	Tuesday, 11:35 - 13:25	Thursday, Feb. 25, 23:55
L2	Thursday, 9:35 - 11:25	Saturday, Feb. 27, 23:55
L4	Thursday, 12:35 - 14:25	Saturday, Feb. 27, 23:55
L3	Friday, 9:35 - 11:25	Sunday, Feb. 28, 23:55
L1	Friday, 14:35 - 16:25	Sunday, Feb. 28, 23:55

To submit your lab work, go to the cuLearn page **for your lab section** (not the main course page). Submit `recursive_functions.py`. Ensure you submit the version of the file that contains your solutions, and not the unmodified file you downloaded from cuLearn! You are permitted to make changes to your solutions and resubmit the file as many times as you want, up to the deadline. Only the most recent submission is saved by cuLearn.

Last edited: Feb. 21, 2021