



CSE 015: Discrete Mathematics Laboratory 4

Fall 2021

Preliminary Notes

- **This lab must be solved individually.** You can discuss your ideas with others, but when you prepare your solution you must work individually. Your submission must be yours and yours only. No exceptions, and be reminded of the CSE academic honesty policy discussed in class.
- Your solution must be exclusively submitted via CatCourses. Email submission will not be accepted. Pay attention to the posted deadline because **the system automatically stops accepting submissions when the deadline passes. Late submissions will receive a 0.** You can upload one or more `.py` files.
- Start early.

Introduction

This lab is all about using recursion as a problem solving tool. All solutions you provide *must* be recursive.

Download the file `recursion.py` from the folder associated with this lab on CatCourses. It contains definitions and test cases for several functions. None of the functions in the file have been implemented. Your task is to implement the functions and upload your completed `recursion.py` file to the appropriate CatCourses assignment.

Once again, your solutions have to be recursive. This means no loops of any kind. Points will be deducted if your solutions are not recursive or if you use built-in libraries to implement your functions.

Exercises

At the start of the file `recursion.py`, there is a call to set the recursion depth limit to 100000. Python typically sets this to 1000, which is too small. The exercises for this lab are as follows:

1. `factorial(n)` - This is the factorial function.
2. `fib(n)` - This is the Fibonacci number function. Each Fibonacci number is defined as the sum of the previous two Fibonacci numbers. The zeroth Fibonacci number is 0, and the first one is 1.
3. `equal(A, B)` - This function checks if two strings are equal. They have to be exactly the same. Your solution must be recursive, so do not just say `return A == B`.

4. `addup(list)` - This function takes as input a list of numbers and adds them all up. Once again, the solution must be recursive.
5. `reverse(A)` - This function takes in a string and returns the string in reverse. As before, recursive solutions only, no loops.