

**CS 100**  
**Lab 2: dynamic allocation and lists**

**Reminder: find a new partner for this lab (50% grade reduction** for working with a person ever worked before).

**You are allowed to use 1 (one) computer per team (50% grade reduction** for using more than 1 computer per team including laptops and tablets).

**Grading:** 30% each program, 10% attendance.

Complete 3 out of 4 problems for full credit.

1. Create a dynamically sized stack (*based on a dynamic array*). The stack should re-size when:
  - i. A push() needs to exceed the current capacity of the stack--double the capacity.
  - ii. A pop() results in the size of the stack being less than half the current capacity--halve the capacity.

Print the size and the capacity every time it changes.

2. Create an ordered *list* whose entries are books with title, author, and year of release. The list should be ordered alphabetically by the title of the book. At a minimum, it must be possible to search entries, add entries, and remove entries.

3. Improve the ordered list in (2). Now, instead of a single list of books, create a list[1] of lists[2]. Each list of type [2] should be an ordered list containing entries for all the books starting with the same letter. The list of type [1] should be an ordered list where each entry is a list of type [2]. Same minimum of list operations apply. Does this provide any kind of improvement over (2)?

4. **Start** with a *singly-linked list of unknown length*. Write a function which will determine the  $n^{\text{th}}$ -to-last element of the list. For example, if the list consists of the elements {1, 2, 3, 4, 5} and  $n = 2$ , the function should return 4. If  $n = 0$  or  $n = 1$  then return the last element (i.e. 5).

**For the purposes of the lab, do not use any previously written code. All code must be written from scratch** (that means no pre-defined data structure libraries (you can still use basic string manipulation libraries)).

For each exercise, make sure to test your programs. **Debug each function separately.**

When finished, demo for the TA, and **submit on iLearn** under assignments. Please upload your

code within one .tar.gz (or .zip) file. You only need to upload the file for one of the partners in your group, but **make sure that the TA recorded the members of your group!**