# AE1PGA Lab 9

## Doubly-linked list

Copy your integer list code from last lab, then modify the copied code to implement a doubly-linked list rather than a singly-linked list. Change the type and function names so they aren't the same as the previous code (for example, by prepending "dll_" for doubly-linked list). You should also add two new functions:

- `dll_next_element` which takes a pointer to a doubly-linked list element and returns a pointer to the next element in the list, or NULL.
- `dll_prev_element` which takes a pointer to a doubly-linked list element and returns a pointer to the previous element in the list, or NULL.

After you have done that, also write the equivalent `prev_element` function for your singely-linked list type (you don't have the `prev` pointer to help you this time).

## Stack

Implement a stack data-type. It should support the following operations:

- `stack_create` which takes no arguments and returns a pointer to a stack data-structure, or NULL on error.
- `stack_push` which takes a stack pointer and an integer and adds that number to the top of the stack.
- `stack_pop` which takes a stack pointer and removes and returns the value from the top of the stack, or indicates an error if there is no numbers left (you can choose how to implement that error).

## Queue

Implement a queue data-type. It should support the following operations:

- `queue_create` which takes no arguments and returns a pointer to a queue data-structure, or NULL on error.
- `queue_enqueue` which takes a queue pointer and an integer and adds that number to the back of the queue.
- `queue_dequeue` which takes a queue pointer and removes and returns the value from the front of the stack, or indicates an error if there is no numbers left (you can choose how to implement that error).

---

*End*