CENG 112 – DATA STRUCTURES
Homework 3

March 24, 2015

Due Date: 7 April 2015

Prog. Assignment 1.1 Stack using Linked Lists

Modify the stack implementation in “stacks/stack.c” , so that it uses linked lists instead of an array. Compile and run the RPN calculator using your linked list based stack implementation. You can modify “stack.c” in any way you like. You are **not** allowed to modify “stack.h” or the RPN calculator main program source. *Hint:* You should first understand the **queue** implementation and use a similar technique to solve this question.

Prog. Assignment 1.2 Double-Ended Queues

The queue data structure given in “queues/queue.h” and “queues/queue.c” only allows data to be put at one end and pulled from the other end. A double-ended queue (called deque) allows insertions and removals from both ends.

Implement a deque ADT similar to the queue that has the following four methods instead of put/get:

- `void put_front(struct Deque *deq, void *elem)` should insert a new element at the front.
- `void get_front(struct Deque *deq, void *elem)` should remove an element from the front.
- `void put_back(struct Deque *deq, void *elem)` should insert a new element at the back.
- `void get_back(struct Deque *deq, void *elem)` should remove an element from the back.

Place your deque implementation in “deque.h” and “deque.c”. Test your deque implementation by writing a test program “deque_test.c” that is similar to “queue_test.c” but includes actions for `put_front`, `put_back`, `get_front`, `get_back`.