

Problem 8

Here is an excerpt from the definition of a singly-linked list class. A `LinkedList` object represents a singly-linked list of integers. The implementation dynamically allocates the nodes it needs, uses no dummy node, and indicates the end of the list by the last node having `NULL` as its `m_next` data member. The empty list is represented by `m_head` and `m_tail` being `NULL`.

```
class LinkedList
{
    public:
        LinkedList(); // creates an empty list
        ...
        void push_back (int v);
        void unique();
        bool dominates (LinkedList& other) const;
        { return dom(m_head, other.m_head); }
    private:
        struct Node
        {
            Node (int v, Node* n) : m_value(v), m_next(n) {};
            int m_value;
            Node* m_next;
        };
        Node* m_head;
        Node* m_tail;
        bool dom (const Node* p1, const Node* p2) const;
};
```

For this problem, we will ask you to write some function implementation. Be sure your code is syntactically correct as well as functionally correct, and leaks no memory. Notice that the `Node` type has no default constructor.

- a. The `push_back` function appends to the end of the list a node whose value is `v`. Write an implementation of this function.
- b. For each group of consecutive nodes with identical values, the `unique` member function deletes all but the first node of that group, leaving the list containing nodes where each node's value is different from the value of the following node. For example, if the `LinkedList` `a` contained nodes with values `5 3 3 3 8 2 2 5 5 9 7 7 2 2 2`, then after the call `a.unique()`, the list `a` would contain `5 3 8 2 5 9 7 2`.
Write an implementation of the `unique` member function.

- c. A sequence s dominates a sequence t if s has at least as many elements as t , and for all k , the k th element of s is greater than or equal to the k th element of t (if t has a k th element). Every sequence (including the empty sequence) dominates the empty sequence. As example,

the sequence	2 5 3 5 7	the sequence	2 5 3 5 7
dominates	1 3 3 4 7	does not dominate	1 4 4 4 (because $3 < 4$)

Write a recursive implementation of the member function named `dom`, which accepts two Node pointers, each pointing to a linked list of Nodes, and returns a bool. The function returns true if the first linked list dominates the second. The body of `dom` function is no more than 15 statements long and DO NOT use `while`, `for`, `goto`, or `static`.