Homework 2

Due: 5:00pm, February 8, 2008

Total Points: 30 points

1. (6 points) Show the final doubly-linked list with sentinels, similar to the illustration in Figure 3.9, after performing the following operations on an empty list.

```c
push_front(21);
push_front(22);
pop_front();
push_front(23);
push_front(24);
pop_front();
push_front(25);
```

2. (4 points) Show the final stack, similar to the illustration in Figure 3.24, after performing the following operations on an empty stack.

```c
push(21);
push(22);
pop();
push(23);
push(24);
pop();
push(25);
```

3. (4 points) Show the final queue, similar to the queue illustration in the lecture notes, after performing the following operations on an empty queue.

```c
enqueue(21);
enqueue(22);
dequeue();
enqueue();
enqueue(23);
enqueue(24);
dequeue();
enqueue(25);
```
4. (10 points) Suppose we need a method for the **List** class that reverses the list by using only a constant amount of extra memory.

   a. Write a version of the *reverse* method that only moves the *data* elements between *Node* structures, but does not modify any pointers.

   b. Write a version of the *reverse* method that only modifies the pointers of the *Node* structures, but does not move any *data* elements.

5. (6 points) Consider the following code segment.

   ```cpp
   List<int> myList();
   myList.push_front(1);
   myList.push_front(2);
   myList.push_front(3);
   List<int>::iterator itr = myList.begin();
   cout << *itr << endl;
   myList.reverse();
   cout << *itr << endl;
   ```

   a. Assuming *reverse* is implemented as described in problem 4a, what will be the output of this code segment? Explain your answer.

   b. Assuming *reverse* is implemented as described in problem 4b, what will be the output of this code segment? Explain your answer.