These must be completed and shown to your lab TA either by the end of this lab, or by the start of your next lab. The bonus question is worth an extra 0.25 points ($1/8^{th}$ of a lab), but only if you successfully finish the rest of the lab. You cannot get bonus points until you finish the lab.

1. Complete and debug the `CDate` class (available under Lab 3 on the course web). If you complete the code correctly, you should see the following output:

```
25/5/2000
0/0/0
0/0/0
0/0/0
29/2/2000
0/0/0
31/12/2000
0/0/0
30/11/2000
```

2. Complete the Linked List program (available under Lab 3 on the course web page). You will need to complete the following functions:

```
// This function deletes the last element in the linked list.
// Pre-condition: The head of a linked list is provided.
// Post-condition: The last element of that linked list has been removed.
void delete_last_element( Node*& head );

// This function inserts a key after a node with a given key.
// If there is no node with the given key, no action.
// Pre-condition: The head of a linked list,
// a key to indicate where to insert,
// and a new key to insert are provided.
// Post-condition: If a node with key is found, the linked list
// contains a new node (newKey) after that node.
void insert_after( Node* head, int key, int newKey );

// This function merges two linked lists.
// Pre-condition: Two linked lists (list1 and list2) are provided.
// Post-condition: A new linked list is returned that contains the keys
// of list1 and list2, starting with the first key of list1, then the
// first key of list2, etc. When one list is exhausted, the remaining
// keys come from the other list.
// For example: [1, 2] and [3, 4, 5] would return [1, 3, 2, 4, 5]
Node* interleave( Node* list1, Node* list2 );
```

   If you complete the code correctly, you should see the following output:

```
<A> List 1: [3, 2, 1]
<B> List 2: [6, 7, 8, 9, 10]
<C> List 1: [3, 2]
<D> List 1: [3]
<E> List 1: []
```

```
<F> List 1: []
<G> List 1: [11, 12]
<H> List 1: [11, 12]
<I> List 4: [11, 6, 12, 7, 8, 9, 10]
<J> List 4: [11, 12]
<K> List 4: []
```

3. **(Bonus)** If you used a recursive approach to implement the `interleave` method, create another implementation using an iterative approach. If you used an iterative approach, now use a recursive approach.