- I'm making the height of the sentinel towers static at 5, though in practice you could make this change at run time. The top level should have no items in it, the bottom level should have every item.
- numItems() returns the value of the private variable numInList, which should track the number of values in the bottom level of the skiplist. The Boolean isEmpty() checks the same value to see if the list is empty. A successful insertion or deletion should modify numInList, of course. However, insertion of an item already in the list should not produce a duplicate of that value (ie: no insertion occurs) and should not do a separate numInList call to check for the item!
- The public method isInList() checks to see if a particular value is in the list; it uses the private SkiplistSearch() method we talked about in class to do the work.
- You can substitute for the getRandom any random number generator.
- When you print a skiplist, you should print what is at each tower level in the skiplist. However, you do not need to worry about lining them up with one another. So the following example is fine:

```
*head*   14    36    90   *head*
*head*   14    18    22    36    44    51    68    90    97    *head*
```

- You get to define the types for item, skiplist, and itemval. The methods Skiplist() and ~Skiplist() are a constructor and destructor respectively for a skiplist. One should initialize the towers in the constructor.

Grading: This assignment is worth up to 10 points. For full credit, be sure to comment (3 points), give a complete printout of your code as well as the source code so the grader can check your implementation (3 points), and do a substantial test run where you insert and delete items, and periodically print the skiplist (4 points). The last should be sufficient to demonstrate to the grader that your code works properly on all the cases outlined above. For submission of the actual code I will either use a lab to test it, or set up a HackerRank depository, so please check back in class so I can let you know if that works!

You are free to discuss implementations with each other, but I am asking you to please *produce your own code*. You may have to do this in a job interview under time pressure, so you need practice.

Assignment 4

Due next Monday, April 11

Welcome to HumbugCorp! We know you will enjoy our software design division, but first we have some programming for you to get you warmed up on. In this assignment you will implement the SkipList class. The description of the class is as follows. You may implement this in C++ or Java.

```
class SkipList
{
public:
    SkipList();
    ~SkipList();

    // Public interface methods
    bool isInList(*skiplist* and *itemval*);
    void SkiplistInsertion(*skiplist* and *itemval*);
    void SkiplistDeletion(*skiplist* and *itemval*);
    bool isEmpty();
    int numItems();
    void printList();

    // (Private) Head array
    Node<T> *head;
    static const int maxHeight = 5;

    // (Private) utility methods and data
    * *item* SkiplistSearch(*itemvalue*);
    double getRandom();
    int numInList;
};
```

A few important notes on the class:

- You can modify things like types, but you should implement all of the methods and data above. Work from the pseudocode given out in the lecture.