CS 61B Homework 10
Due 3pm Wednesday, May 3, 2000

This homework will give you practice writing linked list code in C++.

Copy the Homework 10 directory by doing the following, starting from your home directory.

```
mkdir hw10
cd hw10
cp $master/hw/hw10/* .
```

The file slist.cc defines the SortedList class, and slist.h declares it. A SortedList is a list of ints sorted from lowest to highest. The implementation is similar to the singly-linked list that you used in Homework 3.

Your task is to write three methods for manipulating lists. The file testlist.cc is provided to help you test your list implementation. Compile the program with the command

```
g++ -g -Wall -o testlist *.cc
```

You can run the test program by typing "testlist", and you can use the gdb debugger as you did in Lab 14. Because C++ pointer code is hard to debug, you should reconcile yourself to getting used to gdb (or some other debugger).

Part I (3 points)

Fill in the body of the isIn() method. Your isIn() method must run in linear time--O(n) where n is the length of the list. Uncomment the line in testlist.cc that calls the test function isInTest(), and run the test.

Part II (3 points)

Fill in the body of the remove() method. Note that remove()'s parameter is an element, not a node. remove() should remove ALL instances of the given element, and have no effect if the element is not in the list. Your remove() method must run in linear time.

Uncomment the line in testlist.cc that calls the test function removeTest(), and test.

Part III (4 points)

Fill in the body of the merge() method. Your merge() method must run in linear time--O(n + m), where n and m are the lengths of the two lists. Your implementation should not create or destroy any ListNodes; it should reuse the nodes from the input lists. Upon completion, this list contains all the nodes from both input lists, sorted in ascending order, and the list "sl" is empty.

Uncomment the line in testlist.cc that calls the test function mergeTest(), and test.

C++ notes

There are several C++ constructs in the file that you do not need to understand to do this assignment. In particular, the constructors use the following syntax for initializing fields:

```
SortedList::SortedList(): size(0), head(NULL) {
}
```

"fieldname(value)" is a constructor call (C++ has built-in constructors for primitive values), and is equivalent to the assignment "fieldname = value;". The above constructor could also be written

```
SortedList::SortedList() {
  size = 0;
  head = NULL;
}
```

C++ has no garbage collector, but the programmer can explicitly delete any objects that are no longer in use. (When your program terminates, all of these objects are cleaned up if you haven't already deleted them.) "~SortedList" is a destructor which is called when a SortedList is deleted; it deallocates all the nodes in the linked list (by calling the recursive procedure "destroyAll").

```
SortedList::~SortedList() {
  destroyAll(head);
}
```

Submitting your solution

Change (cd) to your hw10 directory, which should contain slist.cc. Your entire solution should be in this file. Include your name, login, and lab section at the top of slist.cc. From your hw10 directory, type "submit hw10".