CS 61B Lab 3
February 2-4, 2000

Goal: This lab will give you practice with linked lists.

Copy the Lab 3 directory by starting from your home directory and typing:

```
mkdir lab3
cd lab3
cp $master/lab/lab3/* .
```

Getting Started

Please make sure you have a partner for this lab.

The files in the lab3 directory contain classes to implement a singly-linked list. The classes include an encapsulated SList and the underlying SListNode class (which is akin to a Scheme cons cell). Compile SList.java (using "javac -g List.java".) Run the resulting test code with

```
java SList
```

The main() method of SList includes test code, which can be used to help debug the list code before SLists are used in other programs.

Items in SLists are indexed starting from 1, unlike Java arrays.

Part I: Using SLists (1 point)

In the main() method, construct a list that looks like:

```
[ 6 9 12 ]
```

and print the resulting list.

Add more lines to change this list to:

```
[ 3 6 9 12 15 ]
```

and print the resulting list.

Part II: insertAt() on an SList (1 point)

Write a method called insertAt that takes an Object and a position in the list, and inserts the Object between items position - 1 and position. After the insertion, the new Object is the item at position "position". For any n >= position, the item that previously was at position n is now at position n + 1.

Uncomment the call to testAfterInsertAt() in the main() method so that your method will be tested. For example, give a list [1 2 3], inserting the Integer object 4 at each of the following positions would produce the list shown:

```
position = 1:  [  4  1  2  3  ]
position = 2:  [  1  4  2  3  ]
position = 4:  [  1  2  3  4  ]
```

Part III: Adding to the End of a SList (2 points)

A method called insertEnd() already exists, but it runs in linear time, because every time it is called, it walks down the list to find the end. Without changing the meaning of this method or any other, modify the representation of a SList and whatever methods are necessary to make insertEnd run in constant time. Your SList class will need to continually maintain a record of the last (tail) SListNode in an SList, and all SList's methods will have to ensure that this record stays current.

Check-off

Show your TA or Lab Assistant your main() and insertEnd() methods and run the program.

1 point: Show your main() method, and show that it is printing the proper output for Part I.
1 point: Show that your program is printing the proper output for Part II.
2 points: Show your insertEnd() method, and explain how you got it to work in constant time. Which other methods had to be modified?