CS106B
Spring 99-00

Handout #21
April 24, 2000

Section Handout #4

Same as last week, for the following linked-list questions, assume that the following definition exists:

```
   typedef struct cellT {
       int value;
       struct cellT *link;
} cellT;
```

Problem 1: Two ways for Length

Write a function that computes the length of a linked list. First, write the function without using recursion. Then, re-write it using recursion.

Problem 2: Iterative or Recursive Split

On your first assignment, you were asked to write an function to split a linked list of numbers into two linked lists that contain the odd and even numbers of the original list. For example, if the starting list was `(4 2 5 3 14)`, then when `Split(list, oddList, evenList)` ends, `oddList` is `(5 3 1)`, and `evenList` is `(4 2 4)`.

Now, it is possible to write this function both iteratively and recursively. Discuss both solutions, and see if you find the recursive one simpler. For this exercise, don't make copies of the cells, just use the elements of the original list to make up the two new lists.

Problem 3: The Philosophy of ADTs

a) What is an ADT? What are some ADT's that you've already used?

b) What is meant by "breaking the wall" of an ADT, and why is it a bad thing?

c) Why do we keep the implementation of an ADT separate from its interface?

d) Why do we say that it's sometimes okay to use globals inside of an ADT?

Problem 4: Midterm!

Ask away at your section leader any questions you have about the material to help you prepare for the midterm on Friday. =)