

Computer Science I – Summer 2011 Recitation #2: Linked Lists

For each question use the following struct definition:

```
struct ll {
    int data;
    struct ll* next;
};
```

1) Write a function that takes a pointer to the front of a linked list and changes the list by adding an integer *n* (passed in as a parameter) to each node of the list.

```
void addN(struct ll* list, int n);
```

2) Write a function that deletes the first node in a linked list and returns a pointer to the new front of the list. If there are no items in the original list, NULL should be returned.

```
struct ll* deleteFirst(struct ll* list);
```

3) Write a function that makes a copy of an input list and returns a pointer to it. Note: This function should call malloc once for each node in the original list.

```
struct ll* copy(struct ll* list);
```

4) *p* contains the elements 66, 9, 14, 52, 87, 14 and 17, in that order. Consider running the following line of code:

```
p = question4(p);
```

where *question4* is the function defined below. Show the contents of *p* *after* the function call.

```
struct ll* question4(struct ll *list) {

    struct ll* a = list;
    struct ll* b = list;
    struct ll* c;

    if (a == NULL) return NULL;

    while ( a->next != NULL)
        a = a ->next;
    a->next = b;
    c = b->next;
    b->next = NULL;
    return c;
}
```