Prelab 5 (for February 23rd)

For this prelab you will create and implement an ADT to emulate a simple grocery list. Here's how we want users to be able to declare variables of type GroceryList.

```
GroceryList list;
```

The documentation for the ADT will say that variables of this type need to be initialized using the following interface function:

```
GroceryList GroceryListInit(int maxLength, int *errorCode)
```

where maxLength defines the maximum number of items that can be stored in the list. The following shows how a user might initialize their variable list:

```
list = GroceryListInit(100, &ec); // Create empty grocery list that can hold up to 100 items
```

where ec is an int variable the user has chosen for the error code. Next we have an interface function for adding a new item (a character string) to the end/tail of the list:

```
GroceryList GroceryListAppend(GroceryList glist, char * itemName, int * errorCode)
```

which the user can call as:

```
list = GroceryListAppend(list, “Lettuce”, &ec);
```

Next we have an interface function for accessing the kth item on a list:

```
char * GroceryListGetItem(GroceryList glist, int k, int *errorCode)
```

which can be called as follows to obtain the 3rd item on the list:

```
item = GroceryListGetItem(list, 3, &ec);
```

Next we have the function: int GroceryCount(GroceryList), which returns the number of items on the list, and the function: void GroceryListPrint(GroceryList), which prints the items on the list.

Lastly, we have the function: void GroceryListDelete(GroceryList), which frees any memory allocated for the list.

Really the only thing that needs to be determined is the information required to be stored in the GroceryList struct. Clearly an array will have to be allocated to store items placed on the list, and we'll need to store the length of that array. What else will be needed? Note that the list will initially be empty, i.e., zero items on the list.

As I've discussed in lecture, the best approach for designing an ADT is to begin by defining the desired set of interface functions (i.e., most intuitive and useful for users) without regard to how they might be implemented. Once you have that, then it should be clear what information will need to be maintained in your stuct.