

## Programming Assignment 2 - Part I

### List-Based Implementations of merge sort and quick sort

**DUE: Monday Feb 29 by 11:59PM**

---

You will write two programs:

- A linked list implementation of Merge Sort
- A linked list implementation of Quick Sort

Both programs behave in the same manner:

They simply read a sequence of integers from standard input. When the input is complete, the values are sorted using the appropriate algorithm and the resulting sorted sequence is printed to standard output.

Framework/Tips and Rules:

- Each program should use the linked list ADT implementation you wrote for project-1. You should find that you have already done most of the work! These programs will be clients of the linked-list module.
- In case you are unsure of about the correctness of your linked-list functions from project-1, we have provided an implementation for your use/reference. We believe it to be bug-free.
- You are free to add new functions to the linked list module, but you shouldn't need much (if anything). You might find that your quick sort implementation will be a little cleaner if you have a function like this:

```
ElemType lst_peek_front(LIST *lst);
```

This function returns the value of the first element of the list without removing it from the list. (If the list is empty, I suppose some arbitrary value can be returned).

- Write your sorting programs as *clients* of the linked list ADT (probably in files msort.c and qsort.c)
- Absolutely no arrays can be used -- directly or indirectly.
- Reading from standard input:

The recommended method for reading the input is to repeatedly call scanf attempting to parse an integer each time until scanf fails or returns EOF.

This also means that if the input contains 10 integers followed by some text, the program will sort the 10 integers and be done with it.

Simple and sensible.

- Submission: you will zip a directory containing all necessary source files and a makefile. When we unzip your directory we should be able to simply type the following to create your programs.

```
> make msort  
> make qsort
```

---

## **Deliverables**

You will zip your source files along with the Makefile described above and submit to Blackboard.

Note: part II will have a separate submission link.