

Assignment #5 – C-style Linked Lists

Due: Sunday, 06/07/15, 11:59pm

(80 pts) Read **Chap. 17** on linked lists in your book. To begin getting ready for CS 261, you will write a C program that fills and sorts a linked list of integers. Make sure your program compiles using **gcc and the following list.h and test_list.c files on the ENGR server!!!**

list.h

```
struct node {
    int val;
    struct node *next;
};

//Implemented in Lab #9, but now return new head address
int length(struct node *);
void print(struct node *);
struct node * push(struct node *, int); //put at front
struct node * append(struct node *, int); //put at back

struct node * clear(struct node *);
struct node * remove(struct node *, int);
struct node * sort_ascending(struct node *);
struct node * sort_descending(struct node *);
//insert into a specific location in the list
struct node * insert_middle(struct node *, int, int);
```

test_list.c

```
#include "list.h"
#include <stdio.h>
#include <stdlib.h>

int main (){
    char ans[2];
    int num;
    struct node *head = NULL;
    do {
        do {
            printf("Enter a number: ");
            scanf("%d", &num);
            head = push(head, num); //Can change to append

            printf("Do you want another num (y or n): ");
            scanf("%1s", ans);
        } while (ans[0] == 'y');
    }
```

```

        printf("Sort ascending or descending (a or d)? ");
        scanf("%1s",ans);

        if(ans[0] == 'a') head=sort_ascending(head);
        else if(ans[0] == 'd') head=sort_descending(head);

        print(head, length(head));

        printf("Do you want to do this again (y or n)? ");
        scanf("%1s",ans);
        if(ans[0] == 'y') head = clear(head);

    } while(ans[0] == 'y');

    return 0;
}

```

For example:

```

Enter a number: 100
Do you want another num (y or n): y

Enter a number: 30
Do you want another num (y or n): y

Enter a number: 50
Do you want another num (y or n): y

Enter a number: 10
Do you want another num (y or n): n

Sort ascending or descending (a or d)? a

Your linked list is:
10 30 50 100

Do you want to do this again (y or n)? n

```

Extra Credit (10 points): You will change your implementation to work with having a tail pointer as part of your data structure, i.e. not your node but your linked list data structure.

- Add a tail pointer to main, i.e. struct node *tail;
- Specifically change the **append** to take advantage of this extra feature to **make the complexity of that function constant**, rather than $O(n)$.

- How do these functions change?

(10 pts) **Program Style/Comments**

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line! http://classes.engr.oregonstate.edu/eecs/spring2015/cs162-001/162_style_guideline.pdf

```

/*****
** Program: test_list.c
** Author: Your Name
** Date: 06/06/2015
** Description:
** Input:
** Output:
*****/

```

(10 pts) **Design for Assignment #5 changes/Testing**

- (5 pts) How did your design for the Linked List in Assignment #5 change during implementation?
- (5 pts) What were the actual values from your testing? Did these match your expected values? What did you do to make sure you get the expected values?

Please see the template for this document: [Polya template.pdf](#)

You need to have a table with these headings:

Input Values	Expected Output	Did Actual Meet Expected?
t	Error message for choice 't' because that isn't an option. Reprompt for choice again.	Yes

In order to submit the files, you will be creating a bziped tar ball. In order to do this, you will use the following command, adding all the source files to the end of the command:

```
tar -cjvf cs162_hwx_username.tar.bz2 file1 file2 file3...
```

This tar ball (replacing username with your ENGR username), and only this tar ball, will be submitted via TEACH.