

Assignment #5 – C-style Doubly Linked Lists

Due: Sunday, 06/05/16, 11:59pm

(90 pts) 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!!!** You will modify test_list.c and write your own list.c implementation for all the following functions in the list.h file. Reminder: This last assignment is not demoed for a grade. The TAs will compile, run, and grade your program based on what was submitted to TEACH.

list.h

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

struct list {
    struct node *head;
    struct node *tail;
};

int length(struct list); //return number of nodes in list
void print(struct list); //print the values in list
void push_front(struct list *, int); //push to front of list
void push_back(struct list *, int); //push to end of list

void clear(struct list *); //remove all nodes from list
void remove_val(struct list *, int); //remove nodes w/ int as val
void sort_ascending(struct list *);
void sort_descending(struct list *);

//insert into a location in the list, start at 0 for front
void insert(struct list *, int, int);
```

test_list.c

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

int main (){
    char filename[20];
    struct list l;
    FILE *fileptr;
    char num[3]; //We will read positive integers 0-99

    printf("Enter filename: ");
    scanf("%s", filename);
    fileptr = fopen (filename, "r");
```

```

//continue reading until we are at the end of the file
while (fscanf(fileptr, "%s", num)!=EOF)
{
    printf("number is: %d", atoi(num));
    //push to front of list
    //push to back of list

}
//print length of list
//print the contents of the list

//sort in ascending order
//print the contents of the list

//sort in descending order
//print the contents of the list

//prompt user for the value/integer to remove from list
//remove specific value/integer from list
//print the contents of the list

//prompt user for item/int to add and location in list
//insert int into specific location
//print the contents of the list

//clear list - no memory leaks

fclose (fileptr);

return 0;
}

```

(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/spring2016/cs162-001/162_style_guideline.pdf

```

/*****
** Program: test_list.c
** Author: Your Name
** Date: 06/03/2016
** Description:
** Input:
** Output:
*****/

```

In order to submit the files, you will be creating a bzipipped 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 -cvf cs162_hwx_username.tar test_list.c list.c list.h
```

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