Josh Harshman
cscd240
hw4

Function unit testing outputs

Unit Testing for extract function & showList function
Output for extract and showList is as expected.
Extract function extracts valid english words out of a line and show list function displays the list

```
./a.out testfile1 sort1.log sort2.log
|--------------------------------|
|          word   | count        |
|--------------------------------|
|a                | 3            |
|--------------------------------|
|basically        | 1            |
|--------------------------------|
|calling          | 1            |
|--------------------------------|
|each             | 1            |
|--------------------------------|
|file             | 3            |
|--------------------------------|
|for              | 1            |
|--------------------------------|
|from             | 2            |
|--------------------------------|
|fscanf           | 1            |
|--------------------------------|
|function         | 1            |
|--------------------------------|
|i                | 1            |
|--------------------------------|
|in               | 1            |
|--------------------------------|
|my               | 1            |
|--------------------------------|
|scanning         | 1            |
|--------------------------------|
|string           | 1            |
|--------------------------------|
|strings          | 1            |
|--------------------------------|
|the              | 1            |
|--------------------------------|
|then             | 1            |
|--------------------------------|
|using            | 1            |
|--------------------------------|

./a.out testfile1 sort1.log sort2.log
|--------------------------------|
|          word   | count        |
|--------------------------------|
|a                  | 2          |
|--------------------------------|
|basically       | 1           |
|--------------------------------|
|calling           | 1           |
|--------------------------------|
|case              | 1           |
|--------------------------------|
|different        | 1           |
|--------------------------------|
|each              | 1           |
|--------------------------------|
|file                | 3           |
|--------------------------------|
|for                 | 1           |
|--------------------------------|
|from              | 2           |
|--------------------------------|
|fscanf            | 1           |
|--------------------------------|
|function         | 1           |
|--------------------------------|
|i                    | 1           |
|--------------------------------|
|in                  | 1           |
|--------------------------------|
|my                 | 1           |
|--------------------------------|
|scanning        | 1           |
|--------------------------------|
|string             | 1           |
|--------------------------------|
|strings           | 1           |
|--------------------------------|
|the                 | 1           |
|--------------------------------|
|then                | 1           |
|--------------------------------|
|using             | 1           |
|--------------------------------|

./a.out testfile1 sort1.log sort2.log
|--------------------------------|
|           word   | count      |
|--------------------------------|
|a                   | 11         |
|--------------------------------|
|basically        | 1           |
|--------------------------------|
|calling            | 1           |
|--------------------------------|
|each               | 1           |
|--------------------------------|
|file            | 3         |
|------------------------------|
|for             | 1         |
|------------------------------|
|from            | 2         |
|------------------------------|
|fscanf          | 1         |
|------------------------------|
|function        | 1         |
|------------------------------|
|i               | 1         |
|------------------------------|
|in              | 1         |
|------------------------------|
|my              | 1         |
|------------------------------|
|scanning        | 1         |
|------------------------------|
|string          | 1         |
|------------------------------|
|strings         | 1         |
|------------------------------|
|the             | 1         |
|------------------------------|
|then            | 1         |
|------------------------------|
|using           | 1         |
|------------------------------|
```

Unit Test for isLetter() function.
Output is as expected

```
./a.out testfile1 sort1.log sort2.log
output:
1 <- tested whether 'A' is a letter. Came back true
1 <- tested whether 'b' is a letter.  Came back true
0 <- tested whether ',' is a letter.  Came back false
```

Unit Test for addWord() function:

```
./a.out testfile1 sort1.log sort2.log
```
The output is as expected, for the first test I passed in a null head and a single word "word" a new head was created with the word in it. As for the second one, the head was not NULL and I passed in a different word and checked the count on the third pass.

```
output:
word
different
1
```

----
unit test for hasRepeat

Output is as expected. First test I pass in the same word that is in the list and it returns with that position. Second pass I pass a different word and it returns null. Third pass i try a repeat again and it returns with the position of the repeat just like the first pass

output:
```
repeat at node containing word
temp is null, no repeat in list
repeat at node containing word
```

----
unit test for addPos() function
The output is as expected. The function returns the pointer to the node before which the new node should be added. In the calling function, if the return pointer is checked and if it is pointing to the head node, the addFirst() method is called.

output:
```
passed in word b is less than current position containing a.  insert after current pos.

passed in word a is greater than current position containing b.
current position points to head node, add as new head node

passed in word b is greater than current position containing a.  insert after current pos.
```

AddFirst() function unit testing

output is as expected. In the first case, the passed in head value is null and it returns the pointer to the second parameter as the start of the list. in the second case, the head is not null, and it adds the new node to the beginning of the list and returns a new head node pointer. The third test shows the same as the second.

output:
```
old head value is null
new head value is word

old head value is word
new head value is bob

old head value is bob
new head value is adam
```

CreateNode function unit testing

output is as expected. the function is very simple. 1st test gives a word and a value of 1. 2nd test gives a word and a value of 2. 3rd test gives a word and a value of 50.

output:
```
this 1
is 2
cool 50
```

----
the writelist() function outputs the list pointed to by the head pointer parameter into the file pointed to by the file pointer parameter
The output is as expected. It formats the list in a nice format and writes it to a file using fprintf. At the end of the function the file is closed

output:
Here is the content of the file that was written.
the filename was called sort1.log
```
|--------------------------------|
|          word   | count        |
|--------------------------------|
|a                | 3            |
|--------------------------------|
|basically        | 1            |
|--------------------------------|
|calling          | 1            |
|--------------------------------|
|each             | 1            |
|--------------------------------|
|file             | 3            |
|--------------------------------|
|for              | 1            |
|--------------------------------|
|from             | 2            |
|--------------------------------|
|fscanf           | 1            |
|--------------------------------|
|function         | 1            |
|--------------------------------|
|i                | 1            |
|--------------------------------|
|in               | 1            |
|--------------------------------|
|my               | 1            |
|--------------------------------|
|scanning         | 1            |
|--------------------------------|
|string           | 1            |
|--------------------------------|
|strings          | 1            |
|--------------------------------|
|the              | 1            |
|--------------------------------|
|then             | 1            |
|--------------------------------|
|using            | 1            |
|--------------------------------|
```

```
./a.out testfile1 sort1.log sort2.log
|--------------------------------|
|          word   | count        |
|--------------------------------|
|a                | 2            |
|--------------------------------|
|basically       | 1             |
|--------------------------------|
|calling         | 1             |
|--------------------------------|
|case            | 1             |
|--------------------------------|
|different       | 1             |
|--------------------------------|
|each            | 1             |
|--------------------------------|
|file            | 3             |
|--------------------------------|
|for             | 1             |
|--------------------------------|
|from            | 2             |
|--------------------------------|
|fscanf          | 1             |
|--------------------------------|
|function        | 1             |
|--------------------------------|
|i               | 1             |
|--------------------------------|
|in              | 1             |
|--------------------------------|
|my              | 1             |
|--------------------------------|
|scanning        | 1             |
|--------------------------------|
|string          | 1             |
|--------------------------------|
|strings         | 1             |
|--------------------------------|
|the             | 1             |
|--------------------------------|
|then            | 1             |
|--------------------------------|
|using           | 1             |
|--------------------------------|

./a.out testfile1 sort1.log sort2.log
|--------------------------------|
|         word   | count         |
|--------------------------------|
|a               | 11            |
|--------------------------------|
|basically       | 1             |
|--------------------------------|
|calling         | 1             |
|--------------------------------|
|each            | 1             |
|--------------------------------|
|file            | 3             |
|--------------------------------|
|for              | 1        |
|--------------------------------|
|from             | 2        |
|--------------------------------|
|fscanf           | 1        |
|--------------------------------|
|function         | 1        |
|--------------------------------|
|i                | 1        |
|--------------------------------|
|in               | 1        |
|--------------------------------|
|my               | 1        |
|--------------------------------|
|scanning         | 1        |
|--------------------------------|
|string           | 1        |
|--------------------------------|
|strings          | 1        |
|--------------------------------|
|the              | 1        |
|--------------------------------|
|then             | 1        |
|--------------------------------|
|using            | 1        |
|--------------------------------|
```

----
nodeCopy() unit test
the output of the function is as expected. the function is fairly simple. it clones a node as well as all its member variables.

output:

```
node to be cloned contains:
word: clone
count: 2

copied node contains:
word: clone
count: 2

node to be cloned contains:
word: this
count 1

copied node contains:
word: this
count: 1

node to be cloned contains:
word: awesome
count: 500
```

copied node contains:
word: awesome
count: 500

----

sortedCount function unit test

Output is as expected. the sortedCount function duplicates the list and then preforms a selection sort on the duplicated list.

output: Please excuse the formatting. Apple's text editor doesn't copy the format over very well.

```
|--------------------------------|
|       word    | count        |
|--------------------------------|
|a              | 2            |
|--------------------------------|
|from           | 2            |
|--------------------------------|
|calling        | 1            |
|--------------------------------|
|each           | 1            |
|--------------------------------|
|file           | 1            |
|--------------------------------|
|for            | 1            |
|--------------------------------|
|basically      | 1            |
|--------------------------------|
|function       | 1            |
|--------------------------------|
|i              | 1            |
|--------------------------------|
|in             | 1            |
|--------------------------------|
|my             | 1            |
|--------------------------------|
|scanning       | 1            |
|--------------------------------|
|string         | 1            |
|--------------------------------|
|strings        | 1            |
|--------------------------------|
|the            | 1            |
|--------------------------------|
|then           | 1            |
|--------------------------------|
|using          | 1            |
|--------------------------------|

|--------------------------------|
|       word    | count        |
|--------------------------------|
|a              | 14           |
|--------------------------------|
|file           | 2            |
|--------------------------------|
|from           | 2            |
|--------------------------------|
|each           | 1            |
|--------------------------------|
|basically      | 1            |
|--------------------------------|
|for            | 1            |
|--------------------------------|
|calling        | 1            |
|--------------------------------|
|function       | 1            |
|--------------------------------|
|i              | 1            |
|--------------------------------|
|in             | 1            |
|--------------------------------|
|my             | 1            |
|--------------------------------|
|scanning       | 1            |
|--------------------------------|
|string         | 1            |
|--------------------------------|
|strings        | 1            |
|--------------------------------|
|the            | 1            |
|--------------------------------|
|then           | 1            |
|--------------------------------|
|using          | 1            |
|--------------------------------|

|--------------------------------|
|       word    | count          |
|--------------------------------|
|file           | 4            |
|--------------------------------|
|a              | 2            |
|--------------------------------|
|from           | 2            |
|--------------------------------|
|each           | 1            |
|--------------------------------|
|basically      | 1            |
|--------------------------------|
|for            | 1            |
|--------------------------------|
|calling        | 1            |
|--------------------------------|
|function       | 1            |
|--------------------------------|
|i              | 1            |
|--------------------------------|
|in             | 1            |
|--------------------------------|
|my             | 1            |
|--------------------------------|
|scanning       | 1            |
|--------------------------------|
|string         | 1            |
|--------------------------------|
|strings        | 1            |
|--------------------------------|
|the            | 1            |
|--------------------------------|
|then           | 1            |
|--------------------------------|
|using          | 1            |
|--------------------------------|


Total execution time cost for program:  60.334515 seconds
```