Michael Owusu
Com Sci 161
Professor Walker
5/5/2014

Supplemental Problem 5 Write-up

Possible Circumstances and Test Cases

This supplemental problem dealt with the implementation of three functions to a linked lists C program.

1. An error that can arise is in the implementation of addNamesFromFile concerns the use of the fgets procedure. This procedure when reading a line from a file returns the value with a new line statement at the end. This could be an issue in that it may cause the names on the list to be read as a different name from the same name entered in manually (without a newline character). It also would make the list have weird gaps in it because of the extra lines.

To test and fix this, I ran the addNamesFromFile procedure I had written and it indeed inserted names into my list followed by an extra new line character. This made it so my removeDuplicates procedure read these names as different ones from the same name without an extra new line. It is also evident when the print procedure is run because all the names that have been read in from a file have an extra empty line afterwards. To fix this, in the procedure I had for addNamesFromFile, I included an if statement that in the event that there is a new line at the end of a name, it is removed.

2. Another problem that can reasonably arise in the implementation of addNamesFromFile is the names that aren't in the file being read.

To test this this I ran my procedure and noticed that it was reading in several additional characters that seemed nonsensical. I checked every bit of my code and could not find out why the file was being read incorrectly. After trial and error with the files I was reading in, I found that it was an error in the type of file I was reading. Through experimentation, I found that these functions only work for files of type .txt. When I used .rtf type text, all the extra formatting was being read by my fgets procedure therefore giving additional characters when creating new names for the list.

3. A problem that can arise with the removeDuplicates procedure is whether or not the procedure identifies the right nodes and removes all of the duplicates.

To test this, I ran several lists through the removeDuplicates procedure. When a list with no duplicates is tested, I expected the same list to be printed before and after the procedure. I then tested the procedure with one duplicate which I expected to be removed. I then tried with multiple duplicates of the same name and multiple names with duplicates. When my code removed all duplicates while leaving the rest of the list in tact, the code was correct.

4. Lastly, a potential problem with the duplicate procedure is if it created or did not create new nodes the way it is supposed to. That is to say, if it didn't duplicate every node or did more than duplicate a node.

I tested this by making lists both on hand and on my program and then running the duplicate procedure. I compared the results of my program output with what I expected. When I checked my code output for accurately duplicating everything on the list, the code was correct.

Statement of Correctness:
I know my code is working correctly for several reasons. For one, I identified possible circumstances that could arise and my code has none of those problems. My code's functionality is also observable in my sample output section. Additionally, all my procedures agree with the pre and postconditions that came already defined with this problem. As far as I understand this problem, my code is meeting the requirements as proposed and I therefore believe my work is correct.

Sample Output:
Below is a sample output from running this procedure which shows that the three functions I implemented work correctly:

```
Program to Maintain a List of Names
Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: F
Function to insert names from a file to the end of the list
  Enter file name:  names.txt
Renn was added to the list
Anita was added to the list
Noah was added to the list
Richard was added to the list
Krista was added to the list
The names from file have successfully been added to list
Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
```

```
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: P
The names on the list are:

Krista
Richard
Noah
Anita
Renn

End of List

Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: D
Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: P
The names on the list are:

Krista
Krista
Richard
Richard
Noah
Noah
Anita
Anita
Renn
Renn
```

```
End of List

Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: R
Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: P
The names on the list are:

Krista
Richard
Noah
Anita
Renn

End of List

Options available
I - Insert a name (from the keyboard) into the list
P - Print the names on the list
F - Read a sequence of names from a file onto the list
R - Remove duplicate names (leaving only the first occurence)
D - Duplicate each node
Q - Quit
Enter desired option: Q
Program terminated
```