CS303 Programming Assignment #4: "Implementing and Measuring Performance of Sort"

Out: Mar 20, 2013. **Due**: Apr 3, 2013 by 7:00 pm on the CS303 Moodle site.
Total points: 100. approximately 20% of the total homework grade.
Useful textbook pages: Program 8.1 on page 337 has an example implementation of merge(). Program 8.7 on page 356 has an example implementation of mergesort() on linked-lists. Program 6.3 on page 264 has an example implementation of insertionSort().

Students are given the opportunity to write a C program and earn extra credit by measuring its performance. Students may choose to work alone or with a partner.

Policy on working in partners:

1. Choose your partner.
2. ***Each*** member of the partnership should e-mail Dr. Crenshaw before March 23rd. The subject line should read "CS303 Partners". In the e-mail, you should state the name of your partner and that you've agreed to work together.
3. Once you send the e-mail, you are committed to working with your chosen partner for the entire assignment. If you find yourself in the undesirable situation that your partner is a doofie slacker, you are stuck. By sending the e-mail, you have agreed that you and your partner will receive the same grade on the assignment.
4. You and your partner may view, copy, and e-mail code between each other, but you may not share code with any other pair of students. If one partner illegally obtains code, both partners may experience a consequence.

You must write a C program which has the following features (70 points):

The program must execute in three modes:

1. Verbose Mode. Read a file, create a linked-list of company entries from the file, print the linked list, and free memory. ***This has been completed for you***.

```
$ ./prog4 -v short.txt
./prog4: Successfully parsed file, short.txt.
* Smarsh
  Email archiving and encryption
  http://www.smarsh.com/
  921 SW Washington Street, Suite 540
  Portland, OR  97205
  (45.521201, -122.680772)
* Iovation
  Real-time device reputation and fraud detection
  http://www.iovation.com/
  111 SW 5th Avenue, Suite 3200
  Portland, OR  97204
  (45.522105, -122.676129)
There are 2 nodes in the list.
```

Note that Verbose Mode does not sort the linked list.

2. Mergesort Mode. Read a file, create a linked-list of company entries from the file, print the linked list and use the merge sort algorithm to sort the list according to three different comparison functions: alphabetically, by zipcode, and by distance from the UP Bell Tower. This is the expected output for the test file short.txt:

```
$ ./prog4 -m short.txt
./prog4: Successfully parsed file, short.txt.
* Smarsh: 97205.  (45.521201, -122.680772)
* Iovation: 97204.  (45.522105, -122.676129)
There are 2 nodes in the list.

****Sorted alphabetically****
* Iovation: 97204.  (45.522105, -122.676129)
* Smarsh: 97205.  (45.521201, -122.680772)

****Sorted by zip****
* Iovation: 97204.  (45.522105, -122.676129)
* Smarsh: 97205.  (45.521201, -122.680772)

****Sorted by distance from UP clock tower****
* Smarsh: 97205.  (45.521201, -122.680772)
* Iovation: 97204.  (45.522105, -122.676129)
```

3. Insertion Sort Mode. Read a file, create a linked-list of company entries from the file, print the linked list and use the insertion sort algorithm to sort the list according to three different comparison functions: alphabetically, by zipcode, and by distance from the UP Bell Tower. This is the expected output for the test file short.txt:

```
$ ./prog4 -i short.txt
./prog4: Successfully parsed file, short.txt.
* Smarsh: 97205.  (45.521201, -122.680772)
* Iovation: 97204.  (45.522105, -122.676129)
There are 2 nodes in the list.

****Sorted alphabetically****
* Iovation: 97204.  (45.522105, -122.676129)
* Smarsh: 97205.  (45.521201, -122.680772)

****Sorted by zip****
* Iovation: 97204.  (45.522105, -122.676129)
* Smarsh: 97205.  (45.521201, -122.680772)

****Sorted by distance from UP clock tower****
* Smarsh: 97205.  (45.521201, -122.680772)
* Iovation: 97204.  (45.522105, -122.676129)
```

To receive full points on this assignment, and to achieve the functionality described in items 2 and 3 above, students must implement all the function stubs in sort.c. All other source required by the program has been provided.

To receive 100/100 points on this assignment, you must utilize good programming practice (30 points):

• Variables must have meaningful names and global variables must not be used.	2
• Preprocessor directives must be used for constant values.	2
• Code must be documented with useful comments and should use standard tabbing rules for good readability.	9
• Code should not be redundant. If two snippets of code have similar functionality, make a function or write a loop.	6
• A makefile must be used to compile the program and should be submitted with your homework submission.	2
• All files opened by the program and all memory allocated to the program should be closed and freed before program exit.	5
• Only the main function, usage(), printX() functions should use printf(); other functions should not. Instead, the main() function should print a message depending on the return value of a function.	4
Total	30

Sample Output for long.txt

```
$ ./prog4 -m long.txt
./prog4: Successfully parsed file, long.txt.
* Bonneville Power Administration: 97232.  (45.529037, -122.656020)
* Cascade Microtech, Inc.: 97008.  (45.454297, -122.791545)
* Cypress Semiconductor: 97008.  (45.454297, -122.791545)
* Electro Scientific Industries: 97229.  (45.524983, -122.819959)
* Planar Systems: 97006.  (45.527995, -122.883460)
* Synopsys: 97124.  (45.533454, -122.903025)
* Daimler AG Headquarters: 0.  (48.775418, 9.181759)
* InFocus: 97223.  (45.425029, -122.745250)
* CH2MHill: 97201.  (45.508087, -122.681832)
* Webmd Health Service : 97210.  (45.537377, -122.708080)
* Sony Building: 0.  (35.686451, 139.691162)
* Kabam: 94065.  (37.518682, -122.254077)
* Hewlett-Packard: 98683.  (45.600222, -122.485283)
* Extensis: 97201.  (45.510132, -122.677186)
* CPqD: 13086.  (-23.548943, -46.638818)
* Digimarc: 97008.  (45.452323, -122.792794)
* Action Without Borders: 97204.  (45.520781, -122.673204)
* Harland Financial: 97204.  (45.520815, -122.676485)
* Thumbtack: 94117.  (37.772178, -122.437308)
* Elemental: 97204.  (45.519246, -122.677322)
* Radisys: 97124.  (45.545148, -122.929750)
There are 21 nodes in the list.

****Sorted alphabetically****
* Action Without Borders: 97204.  (45.520781, -122.673204)
* Bonneville Power Administration: 97232.  (45.529037, -122.656020)
* CH2MHill: 97201.  (45.508087, -122.681832)
* CPqD: 13086.  (-23.548943, -46.638818)
* Cascade Microtech, Inc.: 97008.  (45.454297, -122.791545)
* Cypress Semiconductor: 97008.  (45.454297, -122.791545)
* Daimler AG Headquarters: 0.  (48.775418, 9.181759)
* Digimarc: 97008.  (45.452323, -122.792794)
* Electro Scientific Industries: 97229.  (45.524983, -122.819959)
* Elemental: 97204.  (45.519246, -122.677322)
* Extensis: 97201.  (45.510132, -122.677186)
* Harland Financial: 97204.  (45.520815, -122.676485)
* Hewlett-Packard: 98683.  (45.600222, -122.485283)
* InFocus: 97223.  (45.425029, -122.745250)
* Kabam: 94065.  (37.518682, -122.254077)
* Planar Systems: 97006.  (45.527995, -122.883460)
* Radisys: 97124.  (45.545148, -122.929750)
* Sony Building: 0.  (35.686451, 139.691162)
* Synopsys: 97124.  (45.533454, -122.903025)
* Thumbtack: 94117.  (37.772178, -122.437308)
* Webmd Health Service : 97210.  (45.537377, -122.708080)
```

****Sorted by zip****
* Sony Building: 0. (35.686451, 139.691162)
* Daimler AG Headquarters: 0. (48.775418, 9.181759)
* CPqD: 13086. (-23.548943, -46.638818)
* Kabam: 94065. (37.518682, -122.254077)
* Thumbtack: 94117. (37.772178, -122.437308)
* Planar Systems: 97006. (45.527995, -122.883460)
* Digimarc: 97008. (45.452323, -122.792794)
* Cypress Semiconductor: 97008. (45.454297, -122.791545)
* Cascade Microtech, Inc.: 97008. (45.454297, -122.791545)
* Synopsys: 97124. (45.533454, -122.903025)
* Radisys: 97124. (45.545148, -122.929750)
* Extensis: 97201. (45.510132, -122.677186)
* CH2MHill: 97201. (45.508087, -122.681832)
* Harland Financial: 97204. (45.520815, -122.676485)
* Elemental: 97204. (45.519246, -122.677322)
* Action Without Borders: 97204. (45.520781, -122.673204)
* Webmd Health Service : 97210. (45.537377, -122.708080)
* InFocus: 97223. (45.425029, -122.745250)
* Electro Scientific Industries: 97229. (45.524983, -122.819959)
* Bonneville Power Administration: 97232. (45.529037, -122.656020)
* Hewlett-Packard: 98683. (45.600222, -122.485283)

****Sorted by distance from UP Bell Tower****
* Webmd Health Service : 97210. (45.537377, -122.708080)
* Harland Financial: 97204. (45.520815, -122.676485)
* Elemental: 97204. (45.519246, -122.677322)
* Action Without Borders: 97204. (45.520781, -122.673204)
* Bonneville Power Administration: 97232. (45.529037, -122.656020)
* Extensis: 97201. (45.510132, -122.677186)
* CH2MHill: 97201. (45.508087, -122.681832)
* Electro Scientific Industries: 97229. (45.524983, -122.819959)
* Planar Systems: 97006. (45.527995, -122.883460)
* Cascade Microtech, Inc.: 97008. (45.454297, -122.791545)
* Cypress Semiconductor: 97008. (45.454297, -122.791545)
* Digimarc: 97008. (45.452323, -122.792794)
* Synopsys: 97124. (45.533454, -122.903025)
* Radisys: 97124. (45.545148, -122.929750)
* InFocus: 97223. (45.425029, -122.745250)
* Hewlett-Packard: 98683. (45.600222, -122.485283)
* Thumbtack: 94117. (37.772178, -122.437308)
* Kabam: 94065. (37.518682, -122.254077)
* Sony Building: 0. (35.686451, 139.691162)
* Daimler AG Headquarters: 0. (48.775418, 9.181759)
* CPqD: 13086. (-23.548943, -46.638818)

Extra Credit

This programming assignment offers up to 25 points extra credit for measuring the performance of your program executing in verbose mode, merge sort mode and insertion sort mode. First, submit your source code to Moodle. Then, alter your program or your input files so that it may sort a large number of nodes. Use the time tool to measure how long it takes for your altered program to sort 10 up to 1000000 nodes.

Depending on performance, your machine may not be able to sort larger numbers of nodes. Don't worry. Just measure what you can.

<numberOfItems>	insertion	merge	verbose
10			
100			
1000			
10000			
100000			
1000000			

To earn the extra credit, you must submit a formal, typewritten, one-page write-up that should include:

1. A brief explanation of how you altered your program.
2. A code snippet that demonstrates your alteration.
3. A completed table, similar to the one above.
4. A labeled plot of the data in the table.
5. An explanation of the performance measurement results.

Do not submit your altered program.

The time tool

An important tool for understanding code performance is `time`†. This command-line tool can help you evaluate the performance of a program. This tool calculates the "total time elapsed, the time consumed by system overhead, and the time used to execute utility to the standard error stream. Times are reported in seconds." (taken from manual pages for time()).

```
$ time ./prog4 -m long.txt
...

real    0m0.004s
user    0m0.001s
sys     0m0.002s
```

†Your installation of cygwin may not have this tool. If not, I recommend doing this portion of the assignment in the lab, or getting with a partner who has these tools already installed.