# UNIVERSITY OF WOLLONGONG

## SCHOOL OF COMPUTER SCIENCE AND SOFTWARE ENGINEERNG

CSCI124    APPLIED PROGRAMMING    SPRING 2013

ASSIGNMENT 5 (5 MARKS)

DUE:  11:59PM MONDAY 28 OCTOBER 2013 (WEEK 13)

### AIM

Practice the use of the tree data structure

On completion of the assignment, you should know how to:

- Design a program by decomposing the problem into smaller functional units
- Implement a program in stages
- Produce a C++ database program for statistical analysis

### REQUIREMENTS

**This assignment is a refinement of assignment 3. Please read the specification carefully.**

In this task, you are going to write a program that conducts analysis on a large input data set. The problem is defined as follows.

In an experiment a beam of molecules is observed along a straight line.



This point is called the origin and has a position of 0.0.
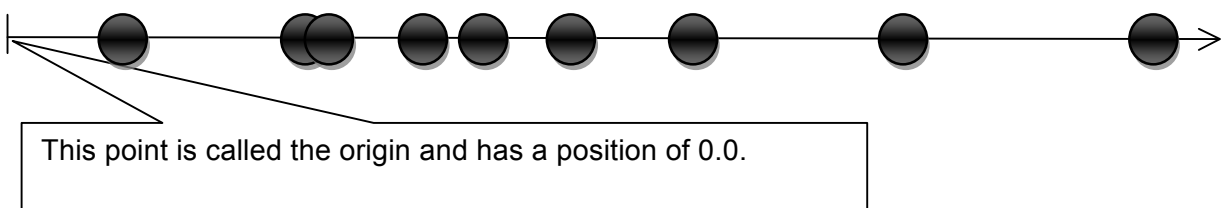
The information recorded for each molecule in the experiment include:

| | |
|---|---|
| position | a double which represent its distance from the original in nm (nanometer) |
| speed | a double in m/s (meter per second) |
| Energy | energy that it carries in mj (micro joule) |
| **fingerprint** | **a sequence of 8 characters from the set {a-z, A-Z, 0-9} which identifies a molecules** |

Note that due to accuracy issue of the experiment setup, two molecules may be observed at the same position.

**The data to be processed could be huge (more than 10 millions) and thus you have to use a binary search tree as the underlying data structure.**

**Experimental Error: It is discovered that each molecule can be identified by a sequence of 8 characters (e.g. ACACDDEF) and that no two molecules should have the same fingerprint (just like human beings). However, in the experiment, the machine in the laboratory may produce data of molecules of the same fingerprint. The researchers thus conclude that the machine is not 100% accurate. To prevent the experiment result from being affected, the data of molecules of the same fingerprint has to be discarded before a report is generated.**

Definition 1: A molecule is **lonely** if there is no adjacent molecules that is within 100 nm.

Your task is to report information related to lonely molecules in the experiment. **Remember, the report generated should depend on the molecules that have unique fingerprint. That is, your program should ignore all the input data the contains the same fingerprint.**

A Sample data file is given below

```
#
23.6
3.3
28
ABCDABCD
34.5
2.4
22
ABCD1234
1230.6
3.3
28
ABCD1234
450.5
2.4
22
ABCD1234
#
```

All data file starts with a symbol # and ends with another symbol #. In the above file, information of 4 molecules are recorded. For example, the first molecule is at the position 23.6 nm with a speed of 3.3 ms and energy of 28 mj. **The fingerprint of the first molecule is ABCDABCD. Since the second, third and fourth molecules share the same fingerprint; they should not be used in the report generation. Thus, the first molecule will be regarded as a lonely molecule. (Since the second molecule will not be used in the report generation.)**

Your program contains the following user interface.

| I | Import data from a data file. Prompt the user for the name of a file for import. |
|---|---|
| N | Display the number of molecules imported **(including those that are to be discarded)** |
| R | Generate a report to the display. A report contains the following information: <br> 1. total number of molecules **(excluding those that have been discarded)** <br> 2. number of lonely molecules <br> 3. average speed of the lonely molecules <br> 4. average energy of the lonely molecules |

| q | quit the program |
|---|---|

There is no upper limit on the total number of molecules. The user may repeatedly import the same or different data files before issuing the command R. The user could also issue command R before and after importing another file. You can safely assume all the files are error-free.

All commands are case sensitive. If the user enters an invalid input, your system should prompt the user to re-enter. You can assume the user will never enter anything longer than 100 characters.

## GUIDELINES

Do not alter the menu options or input data requirements as they will be used to test your program. Comment your code appropriately.

Implement your program in stages. Design is important. Please use a linked list as the underlying data structure in this assignment.

**Do not hesitate to ask your lecturer or lab tutor if you are unsure of anything or encounter any difficulties.** (Remember it is typically the software developer's responsibility to clarify the requirements of the software with the client.)

**Important: YOU MUST SUBMIT A MAKE FILE that compiles your code on Banshee.**

**To show all the warnings, you should compile with the -Wall option. Your code should compile without any warnings.**

As usual, please start early. I am always happy to discuss with you about the idea.

## SUBMISSION

Submit your program (and Makefile, if any) via the submit command.

<div align="center">submit –u <u>userid</u> –c CSCI124 –a ass5 <u>filenames</u></div>

Please make sure you receive the submission receipt after submitting your files.

Remember that you have to put the following information on the header of each source file you will be submitting in this assignment:

- Student name
- Student number
- Lab

An extension of time for the completion of the assignment may be granted in certain circumstances. A request for an extension must be made to the Subject Coordinator before the due date. Supporting documentation must accompany the request for extension. Late assignments without granted extension will be marked but the mark awarded will be reduced by 1 mark for each day late. Assignments more than 3 days late will not be accepted.

For late submission, please use the command:

<div align="center">submit –u <u>userid</u> –c CSCI124 –a ass5-late <u>filenames</u></div>