CS210-001: Assignment #2
Fall 2016

Due Date and Time: Tuesday, October 18, 2016 at 11:55 PM

1. (60 marks) Please use an sorted list to simulate a student database. The student database stores information about each student. This information consists of student ID, student name, and grade. The database provides a menu for users to operate the database, shown as follows:

[I] initialize the database

[N] insert a new student

[S] search the database with a key

[D] delete a student information

[P] print all student information

[X] exit

Here, initialize the database means reading all student information from the file *students.txt* into the database; insert operation should prompt users to input student ID, student name and grade; search and delete operations should prompt users to input the key value.

a. (30 marks out of 60 marks) Array-based sorted list. The student database should be declared as a variable of an array-based sorted list type defined as a C++ class, where each item in the list corresponds to one student information. You can use the array-based sorted list type defined in the lectures, or you can define your own array-based sorted list type. Please note that in this implementation, the key member of the student information is student ID.

Please show that that the program is working correctly by completing the following operations in sequence:

- initialize the database
- Print all student information
- insert a new student information (*2367 Fred 87*)
- Print all student information
- Search the student information with the student ID *8230*
- Delete the student information with the student ID *2367*
- Print all student information

For programming problems, the appearances of the program are worth 30%, and the results are worth 70%. Please read the file *General Marking Criteria for Programming Problems* in URCourses CS210 about the detail distribution information. So for this problem, the results are worth 21 marks out of the 30 marks available.

b.(30 marks out of 60 marks) Linked structure-based sorted list. The student database should be declared as a variable of a linked structure-based sorted list type defined as a C++ class, where each item in the list corresponds to one student information. You can use the linked structure-based sorted list type defined in the lectures, or you can define your own linked structure-based sorted list type. Please note that in this implementation, the key member of the student information is student name.

Please show that that the program is working correctly by completing the following operations in sequence:

- Initialize the database
- Print all student information
- insert a new student information (*2367 Fred 87*)
- Print all student information
- Search the student information with the student name *Mary*
- Delete the student information with the student name *Fred*
- Print all student information

For programming problems, the appearances of the program are worth 30%, and the results are worth 70%. Please read the file *General Marking Criteria for Programming Problems* in URCourses CS210 about the detail distribution information. So for this problem, the Results are worth 21 marks out of the 30 marks available.

2. (40 marks) Stack. Write a program that uses the stack class (you can use the stack type defined in the lectures, or you can define your own stack type) to solve a problem. The problem contains two tasks, the first one is to convert an infix arithmetic expression into a postfix arithmetic expression; the second task is to evaluate arithmetic expressions written in postfix form.

In infix form, the operator of an arithmetic statement is in-between every pair of operands. For example:

```
1.5 + 2.3
2.0 * (3.3 + 4.5)
(2.1 + 3.0) * 4.8
```

In postfix form, the operands of an arithmetic statement appear followed by the operator. One of the virtues of postfix form is that expressions can be written without the need for parentheses. Here are some examples of arithmetic expressions written in postfix form:

```
1.5 2.3 + // Equivalent to 1.5 + 2.3
2.0 3.3 4.5 + * // Equivalent to 2.0 * (3.3 + 4.5)
2.1 3.3 + 4.8 * // Equivalent to (2.1 + 3.0) * 4.8
```

Please write a C++ program that uses an operator stack to convert an infix arithmetic expression expressions that the user enters (assuming the infix arithmetic expressions users input are all valid) into a postfix arithmetic expression, and then use a float stack to evaluate postfix arithmetic. Your program should support all five of the arithmetic operators, +, - , *, /, and %.

Here is an algorithm for evaluating an arithmetic expression written in postfix form with the aid of a stack of floats.

- Scan the expression from left to right.
- When encounter an operand or a float, push it on the stack.
- When encounter an arithmetic operator, pop the top two numbers off the stack and use them as operands for the indicated operation. Push the resulting number back on the stack.
- When you reach the end of the expression, the result of the calculation should be the sole item on the stack.

Please show that that the program is working correctly by converting the following infix arithmetic expressions and then computing the obtained postfix arithmetic expressions:

```
6.5 * 6.5 - 4 *1.5* 1.44

3.1 * (2.5 + 4.7 *1.6) + 8.6

70.0 - (10.5 + 6) / 0.5 + 18.2
```

For programming problems, the appearances of the program are worth 30%, and the results are worth 70%. Please read the file *General Marking Criteria for Programming Problems* in URCourses CS210 about the detail distribution information. So for this problem, the Results are worth 28 marks out of the 40 marks available.