COP4020 Programming Languages
Programming Assignment 4: Course roster management in Prolog

Objective

Experience logic programming with Prolog

Statement of Work

The task in this assignment is the same as that in Assignment 3 (except using slightly different IO due to language features). In this assignment, we will write a course roster management system that provides basic functions for entering, removing, and displaying student information in a course roster using the Prolog language. The course roster is stored as a table of student entries with each entry having three fields: the unique student ID (a string of digits), the student name (a string of characters), and the grade (an integer). The menu for the management system has the following choices:

```
Class roster management system
============================
   MENU
============================
0. Reset roster
1. Load roster from file
2. Store roster to file
3. Display roster sorted by ID
4. Display roster sorted by name
5. Display roster sorted by grade
6. Display student info
7. Add a student to roster
8. Remove a student from roster
9. Exit
```

The *Reset roster* function resets the current roster to be an empty roster. The *Load roster from file* function prompts the user for a file name and loads the roster from the file. The *Store roster to file* function prompts the user for a file name and stores the current roster to the file. The *Display roster* functions display the current roster, sorted by different fields. The *Display student info* function prompts the user to input either the student ID or name, and displays the student information (or reports that the student is not in the roster). The *Add a student* function prompts the user for the student ID, name, and grade (one in each line) and adds the student to the current roster (or reports an error if the student is already in the roster). The *Remove student* function prompts the user to input either the student ID or name, and removes the student from the roster (or reports that the student is not in the roster). More detail is given the trace of a sample run.

Due date and grading policy

The assignment is due on April 24. Put all of your program in one file grade.pl and submit the file on the blackboard. The program should start with ``menu([]).'' in the Prolog environment after the file is loaded.

The program must be able to display the menu and perform at least the following two tasks: enter one student information (add a student to roster) and display the student information (Display roster). Programs that cannot perform these two functions (this includes all programs with compiling errors) will receive at most 10 points. If a program can perform these two functions, it will be graded as follows.

- Reset roster (5 points)
- Exit (5 points)
- Add a student to roster (25 points)
- Display roster sorted by ID (25 points)
- Display roster sorted by name (5 points)
- Display roster sorted by grade (5 points)
- Display student info (10 points)
- Remove a student from roster (10 points)
- Store roster to file (5 points)
- Load roster from file (5 points)

Misc.

- A sample implementation of this assignment has 187 lines of Prolog code.
- All submitted code will be tested by anti-plagiarism software.
- After you implement the basic menu code, you should implement the "Add a student to roster" and one of the "Display roster" functions.
- Prolog has a very limited debugging support and compiling error reporting capability. You should allocate much more time than your normal C++ coding for this assignment. Once you get pass the first two functions, the rest will be **much** easier (prepare to spend a lot of time doing the first two functions).
- You can assume that the input is correct: the code can have undefined behavior when the input is wrong.

Sample output trace:

```
<linprog2:1015> pl
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 5.6.62)
Copyright (c) 1990-2008 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- [’grade.pl’].
% grade.pl compiled 0.01 sec, 33,416 bytes
true.

?- menu([]).
        Class roster management system
        ==============================
           MENU
        ==============================
        0. Reset roster
        1. Load roster from file
        2. Store roster to file
        3. Display roster sorted by ID
        4. Display roster sorted by name
        5. Display roster sorted by grade
        6. Display student info
        7. Add a student to roster
        8. Remove a student from roster
        9. Exit

        Enter your choice (followed by a ’.’): 7.

        Add a student to the class roster
        Student ID : "001".
        Student name : "Michael Jordan".
        Grade : 9.
        Student (ID:001) inserted.

        Class roster management system
        ============================
           MENU
        ============================
        0. Reset roster
        .......
        9. Exit

        Enter your choice (followed by a ’.’):  7.

        Add a student to the class roster
        Student ID : "900".
        Student name : "Justin Fincher".
        Grade : 80.
```

```
Student (ID:900) inserted.

...... (the Menu)
9. Exit

Enter your choice (followed by a ’.’): 7.

Add a student to the class roster
Student ID : "888".
Student name : "Amy S. Stevenson".
Grade : 80.
Student (ID:888) inserted.

......(the Menu)
Enter your choice (followed by a ’.’): 7.

Add a student to the class roster
Student ID : "666".
Student name : "Mike Harris".
Grade : 100.
Student (ID:666) inserted.

...... (the Menu)
Enter your choice (followed by a ’.’): 3.

Display Roster, sort by ID:

No.1: ID="001", Name="Michael Jordan", Grade=9
No.2: ID="666", Name="Mike Harris", Grade=100
No.3: ID="888", Name="Amy S. Stevenson", Grade=80
No.4: ID="900", Name="Justin Fincher", Grade=80

...... (the Menu)
Enter your choice (followed by a ’.’): 4.

Display Roster, sort by name:

No.1: ID="888", Name="Amy S. Stevenson", Grade=80
No.2: ID="900", Name="Justin Fincher", Grade=80
No.3: ID="001", Name="Michael Jordan", Grade=9
No.4: ID="666", Name="Mike Harris", Grade=100

...... (the Menu)
Enter your choice (followed by a ’.’): 5.

Display Roster, sort by grade:

No.1: ID="001", Name="Michael Jordan", Grade=9
No.2: ID="888", Name="Amy S. Stevenson", Grade=80
No.3: ID="900", Name="Justin Fincher", Grade=80
No.4: ID="666", Name="Mike Harris", Grade=100
```

```
...... (the Menu)
Enter your choice (followed by a ’.’): 6.

Display student information:

Enter student name or ID : "Michael Jordan".
ID="001", Name="Michael Jordan", Grade=9

...... (the Menu)
Enter your choice (followed by a ’.’): 6.

Display student information:

Enter student name or ID : "001".
ID="001", Name="Michael Jordan", Grade=9

...... (the Menu)
Enter your choice (followed by a ’.’): 6.

Display student information:

Enter student name or ID : "002".
Student 002 is not in the roster.

...... (the Menu)
Enter your choice (followed by a ’.’): 7.

Add a student to the class roster
Student ID : "001".
Student name : "New Kids".
Grade : 90.
Studnet (ID:001) is already on the roster.

...... (the Menu)
Enter your choice (followed by a ’.’):  7.

Add a student to the class roster
Student ID : "002".
Student name : "Michael Jordan".
Grade : 50.
Student (ID:002) inserted.

...... (the Menu)
Enter your choice (followed by a ’.’): 3.

Display Roster, sort by ID:

No.1: ID="001", Name="Michael Jordan", Grade=9
No.2: ID="002", Name="Michael Jordan", Grade=50
No.3: ID="666", Name="Mike Harris", Grade=100
No.4: ID="888", Name="Amy S. Stevenson", Grade=80
No.5: ID="900", Name="Justin Fincher", Grade=80
```

```
...... (the Menu)
Enter your choice (followed by a ’.’):  2.

Store roster to a file:
Enter file name: cop4020.
Roster stored.

...... (the Menu)
Enter your choice (followed by a ’.’): 8.

Remove a student from roster:
Enter student name or ID : "Justin Fincher".
Student Justin Fincher removed.

...... (the Menu)
Enter your choice (followed by a ’.’): 3.

Display Roster, sort by ID:

No.1: ID="001", Name="Michael Jordan", Grade=9
No.2: ID="002", Name="Michael Jordan", Grade=50
No.3: ID="666", Name="Mike Harris", Grade=100
No.4: ID="888", Name="Amy S. Stevenson", Grade=80

Enter your choice (followed by a ’.’):  8.

Remove a student from roster:
Enter student name or ID : "001".
Student 001 removed.

...... (the Menu)
Enter your choice (followed by a ’.’): 3.

Display Roster, sort by ID:

No.1: ID="002", Name="Michael Jordan", Grade=50
No.2: ID="666", Name="Mike Harris", Grade=100
No.3: ID="888", Name="Amy S. Stevenson", Grade=80

Enter your choice (followed by a ’.’): 8.

Remove a student from roster:
Enter student name or ID : "003".
Student 003 is not in the roster.

...... (the Menu)
Enter your choice (followed by a ’.’): 0.

Roster reset (now empty).

...... (the Menu)
```

```
        Enter your choice (followed by a ’.’): 3.

        Display Roster, sort by ID:
        The class roster is empty.

        ...... (the Menu)
        Enter your choice (followed by a ’.’): 1.

        Read roster from a file:
        Enter file name: cop4020.
        Roster read from file

        ...... (the Menu)
        Enter your choice (followed by a ’.’): 3.

        Display Roster, sort by ID:

        No.1: ID="001", Name="Michael Jordan", Grade=9
        No.2: ID="002", Name="Michael Jordan", Grade=50
        No.3: ID="666", Name="Mike Harris", Grade=100
        No.4: ID="888", Name="Amy S. Stevenson", Grade=80
        No.5: ID="900", Name="Justin Fincher", Grade=80

        ...... (the Menu)
        Enter your choice (followed by a ’.’): 9.
Good-bye
true .

?-
```