**DSA - Assignment 1**

**Library Management System**

**using linked list data structure**

INTRODUCTION  
Your **first assignment in this block** will be using linked list data struture for implementing a small Library Management System  (LMS) in Java language.   
LMS manages information about books, readers and  book lending items. These information are:  
About a book:

1. bcode (string): the code of the book (this should be unique for the book).
2. title (string): the title of the book.
3. quantity (integer): the number of  books with the same code the library has.
4. lended (integer): the number of  books with the same code, which are still lended. Condition: lended ≤ quantity.
5. price (double): The price of the book.

About a reader:

1. rcode (string): the code of the reader (this should be unique for the reader).
2. name (string): the name of the reader.
3. byear (integer): The birth year of the reader (must between 1900 and 2010).

About lending:

1. bcode (string): the code of the book to be lended.
2. rcode (string): the code of the borrower.
3. state (integer): takes values 0, 1, or 2 only.

0: the book is not given to the reader  
1: the book is still at the reader, not given back.  
2: the book is given back to the library.

YOUR TASKS  
You should use 3 linked lists, each one is used to store data for books, readers or book lending items. You should create linked lists from scratch, do not use list structures available in java like ArrayList, Vector or LinkedList classes.  
On running, your program displays the menu as below:  
Book list (8 marks):  
1.1.      Load data from file  
1.2.      Input & add to the end  
1.3.      Display data  
1.4.      Save book list to file  
1.5.      Search by bcode  
1.6.      Delete by bcode  
1.7.      Sort by bcode  
1.8.      Input & add to beginning  
1.9.      Add after position  k  
1.10.    Delete position k

Reader list (1 mark):  
2.1.      Load data from file  
2.2.      Input & add to the end  
2.3.      Display data  
2.4.      Save reader list to file  
2.5.      Search by rcode  
2.6.      Delete by rcode

Lending list (1 mark):  
3.1.      Input data  
3.2.      Display data  
3.3.      Sort  by bcode + rcode

tasks explanation  
Book list (8 marks):  
1.1.      Load data from file  
Allow a user to input  the file name that contains information of books. The content of the file may be

B03  |  Morning   |  12  |  25.1  
B01  |  The noon  |  10  |    5.2  
B02  |  River        |    5  |    4.3  
B05  |  Physics     |    7  |  15.4  
B07  |  Biology      |  11  |  12.2  
B04  |  Southern   |    9  |    5.2  
The first line means that: bcode = B03, title = Morning, quantity = 12, and price = 25.1 (lended is absent and set to 0)

1.2.      Input & add to the end  
Allow a user to add new information about a book. After checking validation of data (including that the bcode could not be duplicated), the book is added to the end of the list.  
1.3.      Display data  
Display data in format:  
bcode  |  title  |  quantity  |  lended  |  price  |   value  
where value = price\*quantity  
For example after loading the above file, this option give the output below:  
code |   Title        |  Quantity  |  Lended |  Price   |   Value  
-------------------------------------------------------------------  
B03  |  Morning   |      12        |      0       |  25.1    |   301.2  
B01  |  The noon  |      10        |      0       |    5.2    |     52  
B02  |  River        |        5        |      0       |    4.3     |    21.5  
B05  |  Physics     |        7        |      0       |   15.4   |   107.8  
B07  |  Biology      |      11        |      0       |   12.2   |   134.2  
B04  |  Southern   |       9         |     0       |   5.2      |     46.8

1.4.      Save book list to file  
Allow a user to input the file name and save the book list to the file. The information and format like the option 12.

1.5.      Search by bcode  
Write the function:  
Node search(String xCode) {}  
which return reference to the node whose info contains the book with bcode = xCode.  
Allow a user to input the pcode to be searched and display the result: found or not found.

1.6.      Delete by bcode  
Write the function:  
void dele(String xCode) {}  
which deletes the node whose info contains the book with bcode = xCode.  
Allow a user to input the bcode to be deleted and then delete the book having that bcode.

1.7.      Sort by bcode

1.8.      Input & add to beginning

1.9.      Add after position  k  
The position of the first element is 0, the second's is 1

1.10.    Delete position k

Reader list (1 mark):  
2.1.      Load data from file  
Allow a user to input  the file name that contains information of  readers. The content of the file may be

R03  |  Hoa   | 1902   
R01  |  La      | 1901  
R02  |  Canh  | 1903  
R05  |  Cay    | 1910  
The first line means that: rcode = R03, name = Hoa, byear = 1902

2.2.      Input & add to the end  
2.3.      Display data  
2.4.      Save reader list to file  
2.5.      Search by rcode  
2.6.      Delete by rcode

Lending list (1 mark):  
3.1.      Input data  
Allow a user to input lending item.  
When running, the screen looks like:  
Enter book code:  
Enter reader code:  
Enter state:  
After the user enter bcode and rcode, the program check and acts as follows:  
- If bcode not found in the books list or rcode not found in the readers list then data is not accepted.  
- If  both bcode  and  rcode found in the lending list and  state=1  then  data is not accepted.  
- If bcode and rcode found but lended = quantity then new lending item with state = 0 is added to the end of the Lending list.  
- If bcode and rcode found and lended < quantity then lended is increased by 1 and new lending item with state = 1 is added to the end of the Lending list.  
3.2.      Display lending data  
3.3.      Sort  by bcode + rcode  
  
Submission Requirements  
Create the directory with a name like **<class>-<name><roll number>-ASS1**, e.g.  
**SE0508-QuangTV00456-AS1                        (1)**

Assignment assessment  
You will be asked to modify immediately and to explain your assignment in lab room to be sure that you are really the author  of the assignment you submitted.