

The Problem Description

You are invited by Leo's Supermarket to build an inventory system to maintain their stock information.

Each stock item (also known as item) is described by the following data:

`id`, an integer, a unique number to identify the item.

`itemName`, a string, the name of the item.

`price`, a positive integer, the price of the item.

`quantity`, an integer, the quantity of the item in stock.

Items are divided into two categories: perishable and non perishable.

Each perishable item has all of the item information described above, and also an expiry date. For example, the figure below

```
item (1243, ritz_crackers, 3, 125)
      29/07/2012
id
itemName
price
quantity
expiry date
item (1247, omo_washing_powder, 8, 120)
id
itemName
price
quantity
```

Figure 1 data of the stock item `ritz_crackers` and `omo_washing_powder`

describes two stock items. One is perishable and has an id number 1243, item name `ritz_crackers`, price \$2 per box, and totally 125 boxes in stock. The 125 boxes of the product have an expiry date which is 29/07/2012. The other item is non perishable and has an id number 1247, item name `omo_washing_powder`, price \$8 per box, and totally 120 boxes in stock.

Your task is to write a program using PROLOG to manage the stock list. The program should have the following functionality.

(1) START

The program starts by reading stock details from the text file `records.txt` (a sample of the file is shown in Figure 2, and generating a list containing data of all stock items. It checks each item on the list, and outputs the id number and item name if a perishable item is expired.

```
1243. ◀id
ritz_crackers. ◀itemName
3. ◀price
125. ◀quantity
29. 7. 2012. ◀expiry date 29/07/2012 3
```

```

end. ◀ end of a record
1247. ◀ id
omo_washing_powder. ◀ itemName
8. ◀ price
120. ◀ quantity
end. ◀ end of a record

```

Figure 2 Sample of the text file `records.txt`

`records.txt` may contain multiple perishable and non perishable items. The list is maintained in ascending order based on id.

The program then displays a menu which consists of 4 possible function selections i) to insert a stock item; ii) to delete a stock item; iii) to display all stock items; iv) to exit from the program. A sample menu is shown in Figure 3

```

*****
*Leo's Supermarket Stock Management System*
*****
* i) Insert a stock item *
* d) Delete a stock item *
* l) List all stock items *
* *
* x) Exit from the Program *
*****
Select a function >>

```

Figure 3 Sample menu

(3) INSERT A STOCK ITEM

The program should be able to insert an item to the list. When the function ‘Insert a stock item’ is selected, the program will prompt the user to enter all details – id, item name, price and quantity. It will also ask the expiry date if the item is perishable. It inserts the item to the list and maintains the order of the list.

(4) DELETE A STOCK ITEM

The program should be able to delete an item from the list. When the function ‘delete a stock item’ is selected, the program prompts the user to enter the id or item name of the item is to be deleted. It searches on the list to find the item, and removes the item from the list. Your program should be able to search based on id or item name to identify the stock and delete it.

If search is failed (ie. the stock doesn’t exists), then your program should inform the user and then return to the main menu.

(5) LIST ALL STOCK ITEMS

The program should also be able to display data of each item on the monitor.

(6) EXIT FROM THE PROGRAM

Once the function is selected, the program should write all the items from the list back to the text file `record.txt`, then exit from its execution.

Requirements

(1) Data related to each stock item should be included within a structure, and hence handling data in this assignment should be mainly structure based operations.

2) Clauses defined for operations should be as generic as possible. This means that you should try to separate data from operations.