Assignment #2: Pharmacy stock management

Goals

- Apply binary search trees to a problem.
- Write an effective program in time and memory.

Problem Statement

You must write a program in Java named Tp2 for the pharmacy drug stock management. Your program receives as input a transaction flow. The transactions can be of the following types:

- Drug delivery.
- Prescription: a request for a list of medications for a client.

- Pharmacy stock listing.
- Setting a current date.
- Order of missing drugs for the current date.

Depending on the quantities of drugs in the stock, the program must determine whether it accepts the request or puts it into the order. When a prescription is served or a delivery is made, the stock of drugs must be adjusted. Each prescription is identified by a unique number and contains the medication list, a one cycle dose, and the number of dose repetitions. An expiration date is associated with each drug. Expired drugs should be removed from the stock.

When a client requests a drug, it will be necessary to provide the medication with the closest expiry date for a total quantity of medication needed (a one cycle quantity * the number of renewals/cycles). For example, if a client requests a drug X for a cycle dose of 30 with 6 repetitions, to satisfy a request, it would be necessary that the stock contains enough unexpired drugs to cover a period of 6 * 30 = 180 days. If there are not enough drugs, the pharmacy must order all the necessary medicines from a supplier by generating a corresponding record in the list of orders for a current date.

For each current date (transaction DATE), a list of orders should be generated for all drugs with insufficient stock quantities and for those that are not in the stock. Expired drugs should be removed from the stock.

Program structure

Program call

The call of your program should be as follows:

```
java Tp2 nomfichier1.txt nomfichier2.txt
```

The results produced by your program have to be saved in the file **`nomfichier2.txt (args[1])`** with the specified format.

Input and Output formats

The input flow represents a sequence of actions/transactions specified in a file(**`nomfichier1.txt)`**. Each action begins with an action name (uppercase word).

Action	Syntax and description
`DATE`	**`"DATE" date ";"`** This action specifies a new current date. A current date could be between 2000-01-01 and 2025-01-01. The current date is important for determining which drugs are expired or which ones are approaching the expiry date. All dates in TP2 are specified in the format YYYY-MM-DD. It is necessary to validate the read dates. So, you must manage the number of days per month and leap years. For a given date, it will be necessary to list all ordered drugs indicating the ordered quantity and after then, delete all items from the order list. If the order list is empty, the current date is displayed following by the message “OK”. Input example: **`DATE 2018-06-5 ;`** Generated output assuming that the order list is empty: **`2018-06-5 OK`** Generated output assuming that the order list is not empty: **`2017-10-27 COMMANDES :`** **`Medicament2 40`**

```
Medicament3 7
Medicament6 3
Medicament8 4
Medicament9 4
```

PRESCRIPTION **"PRESCRIPTION" ":" (drug_name one_cycle_dose number_of renewals)* ";"**

This transaction specifies a request for a list of drugs for a client. Each drug is specified by its name, the amount of medication for one treatment cycle (5 units, one unit per day for drug 1) and the number of cycles (repetitions, 6 times for drug 1). If there is a sufficient quantity of the drug in the stock, you have to produce "OK" message, if not – you display the information of ordered medication followed by the message "**COMMANDE**".

Input example:

```
PRESCRIPTION  :
Medicament1    5       6
Medicament6    3       1
Medicament3    5       1
```

Output example:

```
PRESCRIPTION 1
Medicament1 5 6  OK
Medicament6 3 1  COMMANDE
Medicament3 5 1  COMMANDE
```

APPROV **"APPROV" ":" (drug_name quantity expiration)* ";"**

The transaction specifies medicines supply.

Input example:

```
APPROV :
Medicament1 120 2018-05-29
Medicament5 10 2018-05-27
```

```
;
```

Output example:

```
APPROV OK
```

The same medication could be listed several times but with different expiration dates. Example:

```
APPROV :
Médicament14 12 2018-10-29
Médicament14  2 2018-06-03
Médicament10  8 2018-06-29
Médicament2   7 2018-06-29
;
```

STOCK `"STOCK" ";"`

This transaction displays a current stock. Output format: word "**STOCK**" followed by current date (defined previously by the command "**DATE**") in one line, followed by stock drug list. Each medication is displayed on a new line. Expired drugs or drugs with quantities 0 don't should be present in the list as they must be removed from the stock.

```
STOCK ;
```

Output example:

```
STOCK 2017-10-27
Medicament1 80 2018-05-29
Medicament4 8 2017-10-30
Medicament5 10 2018-05-27

STOCK 2005-04-27
…
Medicament33 82 2009-09-01
Medicament33 96 2010-04-01
...
```

Transactions must be processed in their order of arrival (order of appearance in the input file). Once a transaction is read, it must be processed immediately to display its result before being able to read the next one.

To facilitate parsing, there is at least one white space (space, tab, or line feed) after each string or number. The drug name never contains white spaces.

Examples

Example 1

```
java Tp2 exemple1.txt exemple1+.txt
```

exemple1.txt

```
APPROV :
Medicament1 120 2018-05-29
Medicament5 10 2018-05-27
;
DATE 2017-10-26 ;
STOCK ;
PRESCRIPTION  :
Medicament1    5       6
Medicament6    3       1
Medicament3    5       1
;
PRESCRIPTION :
Medicament1    3       2
Medicament9    4       1
Medicament2    4       1
;
PRESCRIPTION :
Medicament3    2       1
;
PRESCRIPTION :
Medicament8    4       1
Medicament1    4       1
Medicament2    3       12
;
DATE 2017-10-27 ;
APPROV :
Medicament4 8 2017-10-30
;
```

exemple1+.txt

```
APPROV OK
2017-10-26 OK

STOCK 2017-10-26
Medicament1 120 2018-05-29
Medicament5 10 2018-05-27

PRESCRIPTION 1
Medicament1 5 6  OK
Medicament6 3 1  COMMANDE
Medicament3 5 1  COMMANDE

PRESCRIPTION 2
Medicament1 3 2  OK
Medicament9 4 1  COMMANDE
Medicament2 4 1  COMMANDE

PRESCRIPTION 3
Medicament3 2 1  COMMANDE

PRESCRIPTION 4
Medicament8 4 1  COMMANDE
Medicament1 4 1  OK
Medicament2 3 12  COMMANDE

2017-10-27 COMMANDES :
Medicament2 40
Medicament3 7
```

```
STOCK
;
```

```
Medicament6 3
Medicament8 4
Medicament9 4

APPROV OK
STOCK 2017-10-27
Medicament1 80 2018-05-29
Medicament4 8 2017-10-30
Medicament5 10 2018-05-27
```

Submission

10 July 23:59 at the latest.

Rapport

1. Auto evaluation
2. Time complexity analysis in Big O notation of the following transactions:

PRESCRIPTION;
APPROV;
DATE;
Algorithm running time should be expressed in terms of:

- *n* – number of different medications.
- *m* – number of items on a prescription.
- *k – number of items on the order list (COMMANDES)*
- and any other variable that you consider would be necessary.

Grading

- 10% of the final grade.

Grading scheme

Criteria	Description	Score
A.	**Directives**	/ 1
B.	• **General Appreciation** Program Structure + Code quality	/ 1
C.	**Correctness**	/ 5
D.	Efficiency Binary search trees should be used. Any others data structures could be used if necessary in conjunction with ABR.	/ 1.5
E.	**Algorithm analysis** Temporal complexity with justification	/ 1.5
	Total:	/ 10