

# Bài Tập Lớn 2: HỆ THỐNG QUẢN LÝ SẢN XUẤT KINH DOANH

Phiên bản 1.0.0

## 1. Giới thiệu

Một công ty chuyên sản xuất và cung ứng các mặt hàng ra thị trường. Công ty cần phải quản lý lượng sản xuất được, đơn đặt hàng từ khách hàng cũng như các tính toán thống kê cần thiết. Giả định rằng, chỉ có mã các sản phẩm là được lưu vào trong hệ thống. Ở mỗi ca trực, công ty cần ghi nhận lại toàn bộ quá trình sản xuất, kinh doanh.

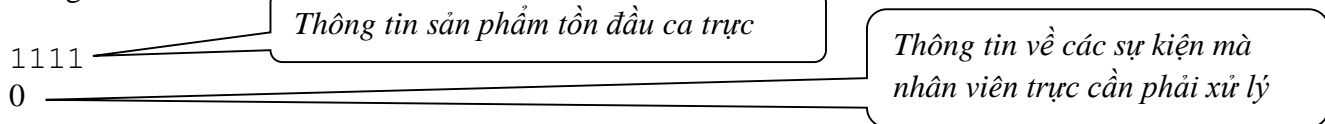
## 2. Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập, bao gồm thông tin về một sản phẩm tồn kho đầu ca trực và các sự kiện đến hệ thống. Một sự kiện có thể là một lệnh nhập kho về một sản phẩm vừa mới được sản xuất ra, hoặc xuất kho cho một đơn đặt hàng về một sản phẩm của công ty, hoặc là một yêu cầu tính toán thống kê,... Sau khi kết thúc công việc, chương trình sẽ xuất ra màn hình hiện trạng dữ liệu của hệ thống.

Các dữ liệu nhập xuất mà sinh viên phải xử lý đều được biểu diễn dưới dạng *danh sách liên kết* (linked list). Chi tiết cụ thể công việc sinh viên phải làm sẽ mô tả trong phần 4.

## 3. Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong file mang tên `input.txt`. File này sẽ chứa các thông tin như sau:



Như vậy file nhập sẽ bao gồm ít nhất hai dòng. Dòng đầu tiên sẽ mô tả thông tin về sản phẩm tồn đầu ca trực, có định dạng `XXXY`. Trong đó `XXX` là mã sản phẩm và `Y` là số lượng tồn.

Kể từ dòng thứ hai của file nhập sẽ mô tả các sự kiện trong ca trực. Mỗi sự kiện sẽ được mô tả bằng một giá trị số, gọi là *mã sự kiện*. Ý nghĩa tương ứng của từng sự kiện được mô tả trong **Bảng 1**. Số sự kiện là không cố định, có thể thay đổi tùy theo test case, và tối đa là  $2^{100}$  sự kiện. Một sự kiện có thể xảy ra nhiều lần. Các sự kiện có thể trình bày thành nhiều dòng.

Bảng 1 – Các sự kiện xảy ra trong ca trực

Mã sự kiện	Ý nghĩa
0	Ca trực chấm dứt đột ngột
1XXX <sup>Y</sup>	Nhập kho <code>Y</code> sản phẩm <code>XXX</code>
2XXX <sup>Y</sup>	Một số lượng <code>Y</code> của sản phẩm <code>XXX</code> đã được khách hàng đặt hàng
3XXX	Yêu cầu thống kê lượng nhập kho của sản phẩm <code>XXX</code>
4XXX	Yêu cầu thống kê lượng đặt hàng của sản phẩm <code>XXX</code>
5XXX	Yêu cầu tính lượng tồn của sản phẩm <code>XXX</code>
6	Yêu cầu cho biết sản phẩm nào có lượng tồn nhiều nhất
8	Yêu cầu loại bỏ các phiếu đặt hàng bất hợp lệ

**Ví dụ 1:** Với dữ liệu nhập là

2223

11113 21112

thì ban đầu có 3 sản phẩm 222 tồn. Trong ca trực, có các sự kiện sau:

Sự kiện 1: nhập kho 3 sản phẩm 111.

Sự kiện 2: có 2 sản phẩm 111 được đặt hàng.

#### 4. Hiện thực chương trình

Sinh viên sẽ hiện thực một phương thức *storage* trong lớp *MyCompanyMgrList* có khai báo như sau:

```
public void storage (notesList theFirst, eventList pEvent)
```

Trong đó *notesList* sẽ là một danh sách liên kết chứa các ghi chú (về nhập/đặt hàng/thống kê kho) trong ca trực, được khai báo như sau:

```
public class notesList {  
    private int nProdID;           //Product ID  
    private int nQuan;             //số lượng, một số có tối đa 2 chữ số  
    private int nType;             //kiểu note: 1: nhập kho, 2: đặt hàng, 0: thống kê  
    private notesList* next;  
}
```

Danh sách này gọi là **Sổ kho**. Cách xây dựng sổ kho sẽ được mô tả trong phần 5. *theFirst* là thông tin sản phẩm tồn đầu ca trực.

##### Định nghĩa 1:

- *NI (NoteInfo)* của một phần tử trong sổ kho là một chuỗi số nguyên có 6 chữ số được tạo thành bằng cách ghép các chữ số mã sản phẩm (3 chữ số), số lượng (2 chữ số) và kiểu (1 chữ số).
- *NsI (NotesInfo)* của một sổ kho là một chuỗi ghép các NI của từng phần tử của danh sách từ phần tử đầu tiên đến phần tử cuối cùng. *NsI* dùng để biểu diễn sổ kho và in ra kết quả khi kết thúc chương trình.

**Ví dụ 2:** Nếu một sổ kho có 3 ghi chú lần lượt chứa các bộ sản phẩm, số lượng và kiểu như sau <11,3,1>, <222,5,1> và <3,12,2>, thì các NI lần lượt là '011031', '222051' và '003122'. *NsI* là '011031222051003122'.

Thông số *pEvent* là một tham khảo trỏ đến danh sách liên kết của các sự kiện được đọc từ file input, được

định nghĩa như sau:

```
public class eventList {  
    private int nEvenCode;  
    private eventList next;  
}
```

#### 5. Xây dựng sổ kho

Khởi tạo, sổ kho sẽ có một phần tử duy nhất là số liệu tồn kho ban đầu.

**Ví dụ 3:** Với dữ liệu nhập là

2223

0

(Ca trực kết thúc ngay sự kiện đầu tiên.) Kết quả của sổ kho là '222031' (Danh sách có 1 phần tử chứa 3 sản phẩm 222).

Trong quá trình thực, sổ kho có thể thay đổi dựa theo các sự kiện. Nếu trong bất kỳ trường hợp nào mà sổ kho rỗng, phương thức *storage* sẽ kết thúc và sổ kho là *null*.

### CÁC SỰ KIỆN CỤ THỂ NHƯ SAU:

S1) Nếu gặp sự kiện có mã là 0, ca trực kết thúc, phương thức *storage* sẽ chấm dứt ngay lập tức. (xem ví dụ 3).

S2) Nếu gặp sự kiện có mã 1XXXXY hoặc 2XXXXY, các phần tử tương ứng sẽ được thêm vào sổ kho.

**Ví dụ 4:** Với dữ liệu nhập là

2223

12224 23331

Ban đầu sổ kho khởi tạo là ‘222031’. Sau đó thêm vào lần lượt các phần tử ‘222041’ và ‘333012’. Kết quả của sổ kho là ‘222031222041333012’.

S3) Nếu gặp sự kiện có mã 3XXX, tổng YY các sản phẩm XXX đã nhập kho, kể cả lượng tồn đầu ca trực nếu có, sẽ được tính toán. Một ghi chú mới được tạo ra với các thông tin là ‘XXXXYY1’. Nếu  $YY > 0$ , tất cả các phần tử ghi chú việc nhập kho của sản phẩm XXX, kể cả ghi chú tồn đầu nếu có, sẽ bị xóa khỏi sổ kho. Ghi chú mới sẽ được thêm vào đầu danh sách. Nếu  $YY = 0$ , ghi chú mới sẽ được thêm vào cuối sổ kho.

*Ghi chú: khi tính toán, YY có thể lớn hơn 99 (số có nhiều hơn 2 chữ số). Khi đó YY sẽ được tính lại bằng cách modulo cho 100.*

**Ví dụ 5:** Với dữ liệu nhập là

2223

12224 22223 3222

Ban đầu sổ kho khởi tạo là ‘222031’. Sau đó thêm vào lần lượt các phần tử ‘222041’ và ‘222032’. Khi gặp yêu cầu thống kê lượng nhập, chương trình tính toán được lượng nhập của sản phẩm 222 là  $7 > 0$ . Toàn bộ các phần tử nhập kho (kể cả tồn đầu) sẽ bị xóa đi và phần tử mới sẽ được thêm vào đầu sổ kho. Kết quả của sổ kho là ‘222071222032’.

**Ví dụ 6:** Với dữ liệu nhập là

2223

3444

Ban đầu sổ kho khởi tạo là ‘222031’. Sau đó gặp yêu cầu thống kê lượng nhập, chương trình tính toán được lượng nhập của sản phẩm 444 là 0. Một phần tử mới sẽ được thêm vào cuối sổ kho. Kết quả của sổ kho là ‘222031444001’.

S4) Nếu gặp sự kiện có mã 4XXX, tổng YY các sản phẩm XXX đã đặt hàng sẽ được tính toán. Một ghi chú mới được tạo ra với các thông tin là ‘XXXXYY2’. Nếu  $YY > 0$ , tất cả các phần tử ghi chú việc đặt hàng của sản phẩm XXX sẽ bị xóa khỏi sổ kho. Ghi chú mới sẽ được thêm vào cuối danh sách bất kể  $YY = 0$  hay  $YY > 0$ .

*Ghi chú: Tương tự như trong S3, YY có thể sẽ được tính lại bằng cách modulo cho 100.*

**Ví dụ 7:** Với dữ liệu nhập là

2223

22221 22222 4222

Ban đầu sổ kho khởi tạo là ‘222031’. Sau đó thêm vào lần lượt các phần tử ‘222012’ và ‘222022’. Khi gặp yêu cầu thống kê lượng đặt hàng, chương trình tính toán được lượng đặt hàng của sản phẩm 222 là  $3 > 0$ . Toàn bộ các phần tử đặt hàng sẽ bị xóa đi và phần tử mới sẽ được thêm vào cuối sổ kho. Kết quả của sổ kho là ‘222031222032’.

S5) Nếu gặp sự kiện có mã 5XXX, lượng tồn YY các sản phẩm XXX sẽ được tính toán (= tổng nhập + lượng tồn đầu (nếu có) – tổng đặt hàng). Tất cả các phần tử ghi chú việc nhập kho (kể cả tồn đầu nếu có) và đặt hàng của sản phẩm XXX sẽ bị xóa khỏi sổ kho. Một ghi chú mới với các thông tin là ‘XXXYY1’ sẽ được thêm vào cuối danh sách.

*Ghi chú: Nếu  $YY > 99$ , YY sẽ được modulo 100. Nếu  $YY < 0$ , YY sẽ được gán bằng 0.*

**Ví dụ 8:** Với dữ liệu nhập là

2223

22221 22222 5222

Kết quả của sổ kho là ‘222001’.

S6) Nếu gặp sự kiện có mã 6, lượng tồn (tính như trong S5) YY của tất cả các sản phẩm có trong sổ ghi chú sẽ được tính toán. Sau đó, ứng với mỗi sản phẩm XXX nào có lượng tồn nhiều nhất, một ghi chú mới với các thông tin là ‘XXXYY0’ sẽ được thêm vào ngay sau phần tử ghi chú nhập kho/tồn kho đầu sản phẩm XXX cuối cùng. Nếu sản phẩm XXX chưa có lần nhập hàng/tồn đầu nào, ghi chú sẽ được thêm ngay vào trước lệnh đặt hàng đầu tiên của XXX.

*Ghi chú: Nếu  $YY > 99$ , YY sẽ được modulo cho 100. Nếu  $YY < 0$ , YY sẽ được gán bằng 0. Việc tính lại YY này sẽ được thực hiện trước khi tìm giá trị lớn nhất.*

**Ví dụ 9:** Với dữ liệu nhập là

2223

12221 22222 13331 6

Lượng tồn nhiều nhất là 2 của sản phẩm 222 (sản phẩm 333 có lượng tồn là 1). Kết quả của sổ kho trả về là ‘22203122201122202022202233011’.

**Ví dụ 10:** Với dữ liệu nhập là

2223

12221 22225 23334 6

Lượng tồn nhiều nhất tính toán được là sản phẩm 222 tồn 0 (do  $3+1-5=-1 < 0$ , làm tròn thành 0) và sản phẩm 333 tồn 0 (do  $0-4=-4 < 0$ , làm tròn thành 0). Một ghi chú ‘222000’ sẽ được thêm vào ngay sau ghi chú ‘222011’ và một ghi chú ‘333000’ sẽ được thêm vào ngay trước ghi chú ‘333042’. Kết quả của sổ kho là ‘222031222011222000222052333000333042’.

S7) Nếu gặp sự kiện có mã 8, chương trình sẽ kiểm tra các ghi chú đặt hàng từ đầu sổ trở đi. Nếu có đặt hàng nào mà lượng tồn của sản phẩm đó là không đủ để xuất hàng thì ghi chú đó được xem là không hợp lệ và cần phải bị loại trừ.

*Ghi chú: Trong trường hợp này, lượng tồn của sản phẩm không bị modulo cho 100.*

**Ví dụ 11:** Với dữ liệu nhập là

2223

12221 13334 22221 23331 22224 8

Khi gặp sự kiện 8, sổ kho có dạng như sau ‘**222031222011333041222012333012222042**’. Chương trình sẽ rà soát và thấy rằng khi nhận đặt hàng 22224 (đặt 4 sản phẩm 222), lượng tồn của sản phẩm 222 chỉ là 3, do đó ghi chú này không hợp lệ và cần phải được loại bỏ. Kết quả sổ kho là ‘**222031222011333041222012333012**’

S8) (Bonus – Câu này chỉ được tính điểm nếu bài làm vượt qua được ít nhất 80% các testcase)

Nếu sổ kho sau cùng là đối xứng, nhân viên của công ty sẽ được thưởng. Trong trường hợp này chương trình sẽ xuất một giá trị chuỗi đặc biệt ra màn hình (Xem phần 6 để biết thêm chi tiết).

**Ví dụ 12:** Nếu sổ kho là ‘**222031130222**’ thì nhân viên công ty sẽ được thưởng.

## 6. Cách dịch và thực thi chương trình

Sinh viên download file *Assignment2\_DataStructures&Algorithms1.zip* từ hệ thống Sakai của môn học. File *input.txt* là một file nhập mẫu như được mô tả ở phần 3. File *RunMgr.java* là chương trình khởi tạo.

**Lưu ý rằng sinh viên không được phép thay đổi các file sau: *RunMgr.java*, *CompanyMgrList.java*, *notesList.java*, và *eventList.java* khi hiện thực chương trình. Ngoài ra, các phương thức do sinh viên viết không được xuất bất kỳ dữ liệu nào ra màn hình khi thực thi.**

Để dịch và thực thi chương trình, sinh viên làm như sau:

- **Bước 1:** mở file *config.bat* bằng notepad hoặc một chương trình chỉnh sửa file text, sau đó sửa đoạn: *C:\Java\jdk1.8.0\_71\bin* thành đường dẫn tới thư mục bin của JDK trên máy mình.
- **Bước 2:** chạy file *compile.bat* để biên dịch, các file sau khi biên dịch được chứa trong thư mục *compiled*.
- **Bước 3:** để kiểm tra chương trình, sinh viên chạy file *run.bat*. File nhập *input.txt* được chứa trong thư mục *compiled*, sinh viên thay đổi file này để kiểm tra các trường hợp khác nhau.

### Lưu ý:

L1) Sinh viên phải **sử dụng danh sách liên kết để lập trình bài tập, KHÔNG DÙNG MẢNG HOẶC CHUỖI, KHÔNG SỬ DỤNG CÁC CẤU TRÚC DỮ LIỆU CÓ THỂ CHỨA NHIỀU PHẦN TỬ SẴN CÓ TRONG JAVA**. Nếu vi phạm, bài làm sẽ **NHẬN ĐIỂM 0**.

L2) Danh sách các sự kiện là một tham khảo *pEvent* chỉ đến phần tử đầu tiên của danh sách. Sinh viên phải tự viết đoạn mã để **truy xuất trực tiếp** lần lượt các phần tử của danh sách này khi xử lý các sự kiện.

L3) Sổ kho kết quả của phương thức *storage* là một tham khảo chỉ đến phần tử đầu tiên của danh sách ghi chú (kiểu tham khảo của *notesList*). Sinh viên phải viết **các thao tác trực tiếp** trên danh sách này trong quá trình thao tác. Sinh viên có thể viết các đoạn mã dùng chung thành các phương thức để gọi lại khi cần.

## 7. Nộp bài

Khi nộp bài, sinh viên sử dụng tài khoản đã được cấp phát trên hệ thống Sakai để nộp bài qua mạng. Sinh viên chỉ nộp đúng một file *MyCompanyMgrList.java* (không được thay đổi tên). **Tất cả các file nộp khác file *MyCompanyMgrList.java* sẽ bị tự động xóa khi chấm bài.** File được nộp phải là file chương trình gốc, sinh viên không được nén file khi nộp bài

Thời hạn chót để nộp bài là **23 giờ 55 phút 00 giây, Thứ Năm ngày 29 tháng 04 năm 2016**. Sinh viên phải dùng tài khoản trên hệ thống Sakai để nộp bài. **KHÔNG** nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác. Bài nộp trễ sẽ **KHÔNG** được nhận.

## 8. Xử lý gian lận

Bài tập lớn phải được sinh viên TỰ LÀM. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, **TẤT CẢ** các bài nộp đều bị coi là gian lận. *Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình. Các bài làm của các sinh viên ở các học kỳ trước cũng sẽ được dùng để kiểm tra gian lận.*
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn trong chương trình khởi tạo. *Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là **KHÔNG ĐƯỢC** sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.*

***Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị xử lý cấm thi cuối kỳ.***

**KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO - KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!**

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.