# Bookstore

## #你需要实现一个书店管理系统

### 一、你需要实现该书店的库存。

#### 1.你需要用一个文件进行数据存储。

（1）运用\*\*块状链表\*\*对文件中存储的数据进行维护。块与块之间按照ISBN的字典序升序进行排列，块内不一定有序。其中块与块的头结点之间形成链表结构，块内采用数组进行存储。

$$

（2）你需要实时掌握库存量N（据说不超过10^5量级），并分别以\frac{\sqrt{N}}{2}和2\sqrt{N}为上下界，对块链进行必要的合并与分裂。

$$

（2）你需要实时掌握库存量N（据说不超过10^5量级），并分别以和为上下界，对块链进行必要的合并与分裂。

#### 2.查找、插入、与删除

（1）查找：顺序查找到关键值所在的块，再数组查找关键值

（2）插入：查找到后将该块中所有值后移一个并插入，N++，如果这时该块的大小超过上界，则在中间进行裂解。

（3）删除：查找到后将后面的值（如果有的话）前移一单位，若此时该块的大小小于下界，则该块和后面一块进行合并。

### 二、你需要实现四种不同权限的账户。

#### 1.概述

（1）每种权限的账户都是一个类，权限从高到低依次为：account\_owner,account\_assistant,account\_common,account\_visitor。由于低权限账户能够访问的功能严格是高权限账户能够访问的功能的\*\*子集\*\*，你\*\*可以\*\*将四个类声明为派生关系。

（2）四种账户类的接口文件（account\_owner.h)与实现文件(account\_owner.cpp)均已给出，你可以在类中添加你所需要的成员，并且需要在cpp中完成实现代码。

（3）账户的存储依然是基于文件的，可以参考库存的存储方式。

### 三、交互逻辑说明（针对每个测试点）

#### 1.概述

（1）main.cpp是程序执行的起始文件，他会直接或间接包括提供的所有的库。

（2）对于测试点的每个测试文件，申明一个流对象进行读取。对于每一条指令，你需要先检验其合法性，若未抛出异常，则进行对应的操作。

（3）显然测试文件只能顺序\*\*读取\*\*，而对于数据存储，你需要多个文件进行存储。

#### 2.操作处理与成员说明

（1）对于账户系统指令：你需要维护一个按先后顺序的登录账户序列，尾端视为当前账户。具体指令操作参考https://github.com/Listener4031/Lsner\_bookstore。

（2）对于图书系统指令：思路与账户系统指令大致相同，需要注意选择图书指令和图书进货指令，具体指令操作参考https://github.com/Listener4031/Lsner\_bookstore。

（3）对于日志系统指令：由于这部分功能需要向助教展示，对于每个员工，你需用一个文件存储所有员工的操作记录，存储方式可以是块状链表。