

復旦大學

计算机科学技术学院

设计报告

课程名称：数据结构

学 院：计算机科学技术学院

专 业：计算机科学与技术

学生姓名：祁佳薇

学 号：16307130293

指导老师：张玥杰

## 一、设计意义

图书管理系统的发展历史可以追溯到上世纪六十年代末期，凭着高速度、高精确性的特点应运而生。数十年来，随着信息爆炸、知识经济时代的到来，人们对图书管理系统有了更高的需求，同时这项技术也在不断发展进步。

本次设计模拟图书馆的日常运营，目的是进一步熟悉和掌握数据结构课程所学基本知识：主要是结构体的定义、链表操作、建立索引表等内容，充分体会数据在程序设计中的重要作用，学会运用数据结构的相关知识解决问题。本次设计主要通过应用结构体、链表、循环等 C++ 语言结构的设计实现对书籍信息的录入、删除、全显、修改等功能。使用结构体可使不同类型的数据存储在相邻存储单元中，便于对相关信息处理。链表的使用可有效地将数据按一定顺序有规律的存储，解决了查找、删除时的无序性。Hash 表的使用使得数据查找更加高效。本次课程实践为下学期的数据库系统学习打下了良好的基础，也让我们迈出了工程化开发实践的第一步。

## 二、需求分析

分析题目可知，程序中需要传递的信息包括：图书信息（书号、书名、著作者、现存量和库存量等）、读者信息（借阅号、姓名、剩余借阅数量、借阅图书信息）和借阅记录（书号、借阅号、借阅日期）。重点在于信息的存储和查找数据结构的选择。

由于图书馆藏书数量巨大，所以需要为书号建立索引表来提高查找效率。索引（index）时把一个关键码与它对应的记录相关联的过程。索引并不重新排列记录的顺序，每个索引往往支持一个关键码，并且通过该索引实现对记录的快速访问。分为线性索引和树形索引，前者索引项组织为线性结构，后者为树结构。看到题目我最先想到的是用哈希表来实现 key 值即书号与指向书籍的指针之间的映射。

选择哈希表的原因一是其查找效率高，它是一个在时间和空间上做出权衡的经典例子：如果没有内存限制，那么可以直接将键作为数组的索引，查找的时间复杂度为  $O(1)$ ；如果没有时间限制，我们可以使用无序数组进行顺序查找，这样只需要很少的内存；哈希表使用了适度的时间和空间来在这两个极端之间找到平衡，只需要调整哈希函数算法即可在时间和空间上做出取舍。常用的 map 数据类型就是这种 key-value 存储的，通过哈希表来实现快速查找。在数据较为平均、哈希函数选择合理的情况下哈希表的查找时间复杂度为  $O(1)$ ，远小于顺序查找  $O(n)$ 、树形查找  $O(\log_2(n))$ ；原因二是哈希表实现简单、操作方便，我假设书籍编号的上限是 200，采用最常见的除留余数法，因为装填因子（表中填入结点数/表长）取 2 左右，所以取 100 左右质数，索引表长定为 97，除数也定为 97，总共有 97 个 bucket， $\text{hash}(\text{key}) = \text{key} \% 97$ ，添加书籍时只需要用书号模 97 取余，然后将结点链接到对应的桶中，而使用树形索引创建树时较为复杂，尤其是动态 AVL 树涉及到添加删除时的平衡因子变化需要调整旋转十分繁琐，所以最终选择了线性索引的方法。

任何事物都有两面性，Hash 索引效率虽高，但不能保证一一对应，不可避免地会出现冲突，解决方法有两种：一是闭散列法（开放定址法），Hash 表的大小固定不变，发生冲突时，用线性探测或二次探测等方法找到；第二种是开散列法（拉链法），将大小为 97 的数组每个元素指向一个单链表，链表中每个节点存储一个书号和指向该书籍的指针。该方法的基本思想就是选择足够大的数组，是的所有链表都尽可能地短小，每次查找时，先根据散列值找到对应链表，再巡链找到相应地键值。我选用了第二种方法解决冲突，实际在平时地应用中开散列法应用更广泛，它有如下几个优点：处理冲突时，非同义词绝不会发生冲突，平均查找长度较短；各链表上的结点空间是动态申请地，更适合无法确定元素个数的情况；闭散列法为减少冲突，要求装填因子  $\alpha$  较小，当结点规模较大时会浪费很多空间，而开散列法中可取  $\alpha \geq 1$ ，且结点较大时，增加的指针域可忽略不计，因此节省了空间；最后是其删除操作容易实现不过在本系统中为涉及书籍出库，所以未能体现。

选择好数据结构，我们可以将系统分为如下四大功能模块：

### 1、图书、读者信息管理

#### ① 采编入库

新购进一种图书，确定书号后，入库，如果库中已有，只增加库存量。

#### ② 新生注册

新生注册需输入姓名，系统为其编号。

### 2、借阅管理

#### ① 借阅

如果一种书现存量大于 0，且该借阅者总借书数目未达上限，则借出一本，改变现存量、改变借阅者信息、将借阅行为记录下来。

#### ② 归还

改变该书现存量，注销该借阅者结构中借阅这本书的记录，注销对借阅行为的记录。

### 3、查找

#### ① 书籍查找

读者信息首先从文件读入，存入一个单链表中。在查找时，分按照书号、书名、作者三种方式。按书号查找时由于图书馆藏书数量巨大，所以需要为书号建立索引关系，根据索引表来查找，本次设计采用哈希表存储的方式建立索引，用开散列的方法解决冲突，采用最常见的除留余数法存储，查找时先将书号模 97 取余，找到该书所在桶，然后在桶内继续遍历查找即可。后两种查找直接遍历单链表即可，将书名或者作者做匹配。如果查找成功，返回该书籍的指针，否则，返回空。

#### ② 读者查找

读者信息也是从文件读入的，存入一个单链表中。在按照借阅号查找时，也需建立借阅号和指针之间的索引关系。本设计中采用与书号相同的哈希函数构造。

### 4、其它信息查看

由用户输入当前的时间（年月日），可以查看所有逾期（三个月）未还的书籍，并输出超时应交罚金。借阅记录链表在从文件中读入读者信息时创建，查找所有逾期未还书籍需要对其进行遍历。

## 三、概要设计

为方便查找和修改，定义结构体四个

功能	结构体名
书籍信息	Struct Book
借阅者信息	Struct Reader
借阅行为记录	Struct Borrow
索引结构	Struct index<Type>

将整个程序分成如下八个文件，对应不同的功能

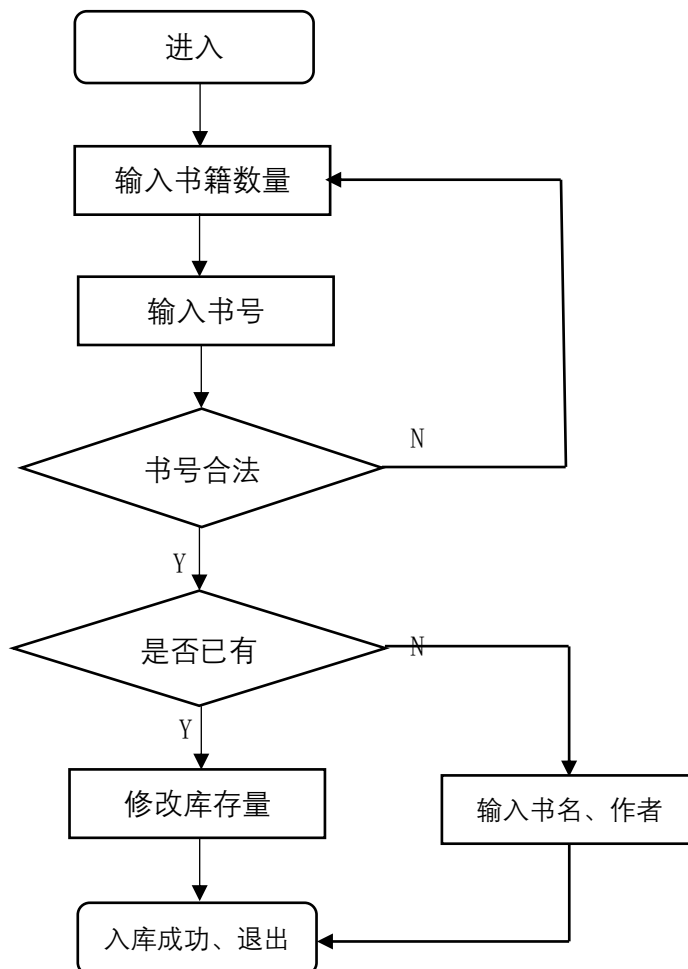
文件名	功能	函数/结构体名称
Library.h	定义结构体、函数声明	Struct Book、 Struct Reader、 Struct Borrow、 Struct index
main.cpp	主函数	case 语句选择功能

file_in.cpp	文件读入	Book* Book_in()、 Reader* Reader_in(Borrow **B2head);
search.cpp	书籍、 借阅者查找	index<T>** init_index(T *Bhead)、 Book* searchBOOK_by_number(long num, Book *Bhead)、 void searchBOOK_by_name(Book *Bhead)、 void searchBOOK_by_auther(Book *Bhead)、 Reader*searchREADER_by_number(longnum, Reader*Rhead);
insert_register.cpp	采编入库、 新生注册	void insert(Book *Bhead)、 void regist(Reader *Rhead);
borrow_return.cpp	借阅、 归还	void borrow_book(Book *Bhead, Reader *Rhead, Borrow *B2head)、 void return_book(Book *Bhead, Reader *Rhead, Borrow *B2head)
fine.cpp	罚款计算 和显示	int isLegal(char s[8])、 void countFine(Reader *reader, Book *book)、 void showFine(Borrow *B2head)
show.cpp	格式打印	void showB(Book *b)、 void showR(Reader *r, Book *Bhead)、 void Menu()

下面画出三个重要算法的框图，便于理解说明

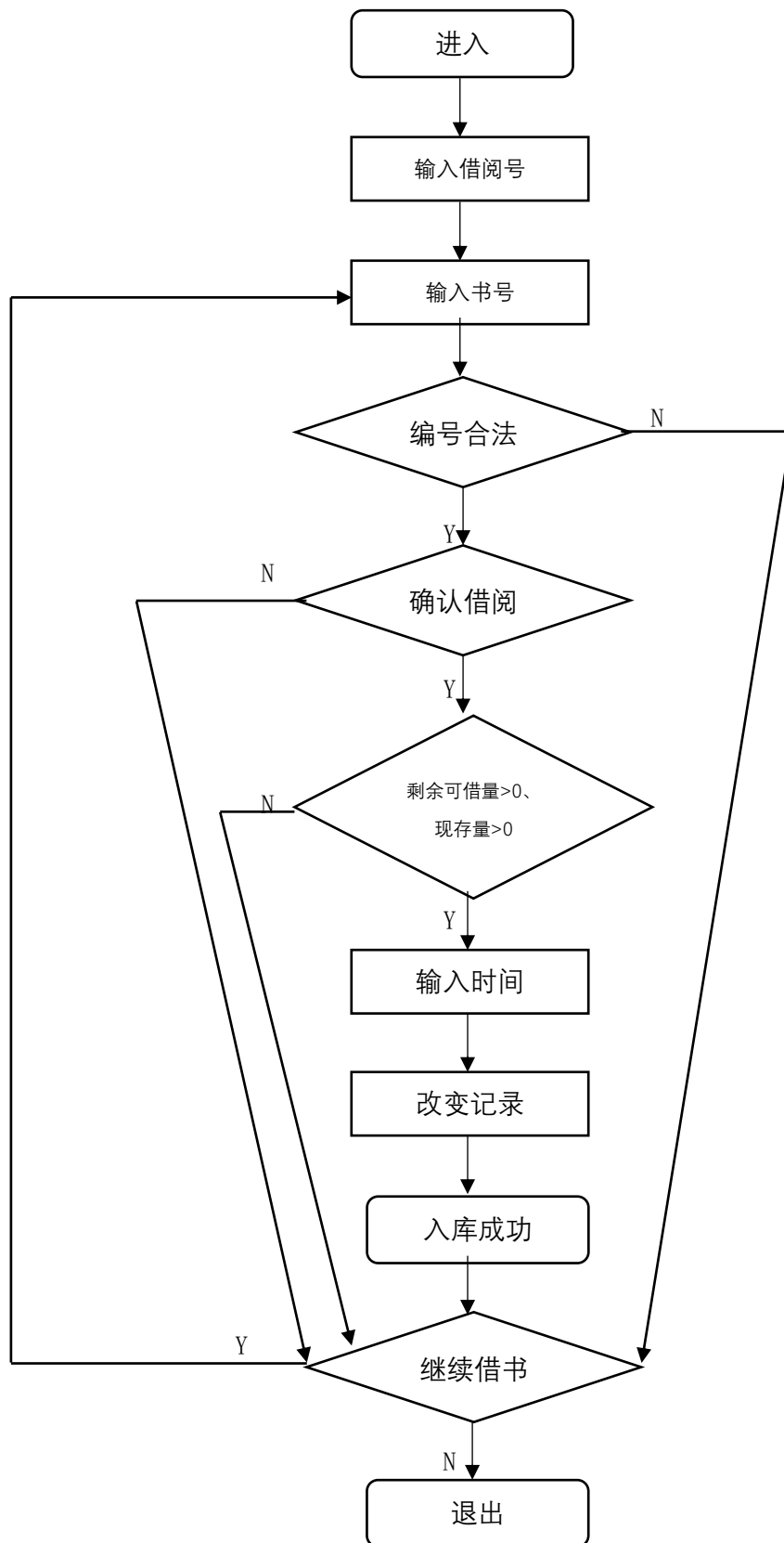
#### 1、采编入库

```
void insert(Book *Bhead);
```



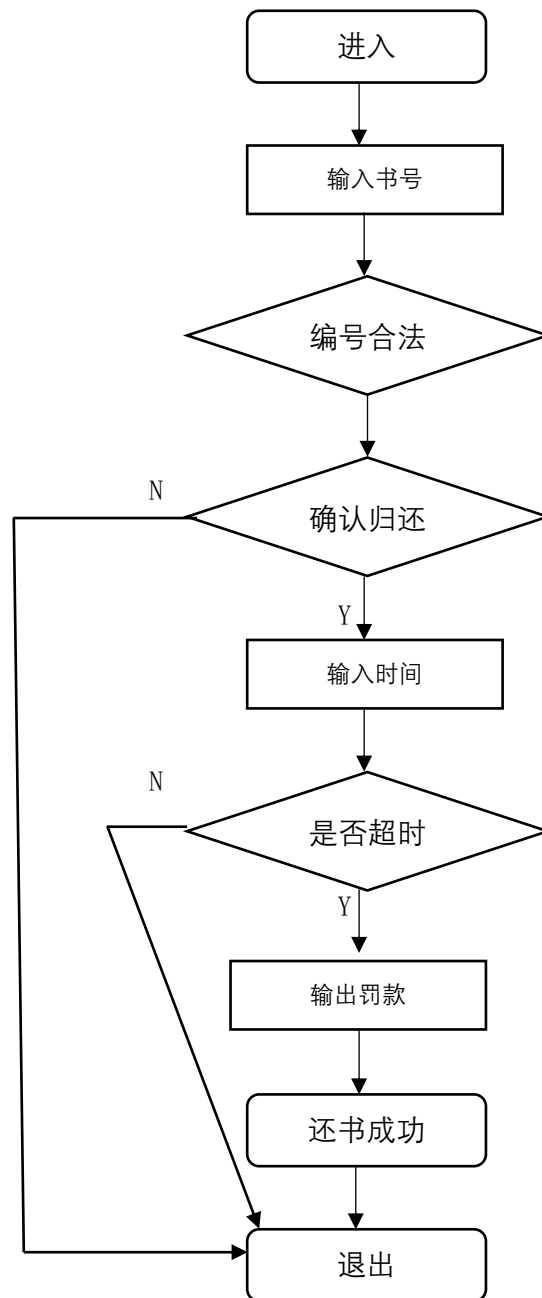
## 2、借阅

```
void borrow_book(Book *Bhead, Reader *Rhead, Borrow *B2head)
```



### 3、归还

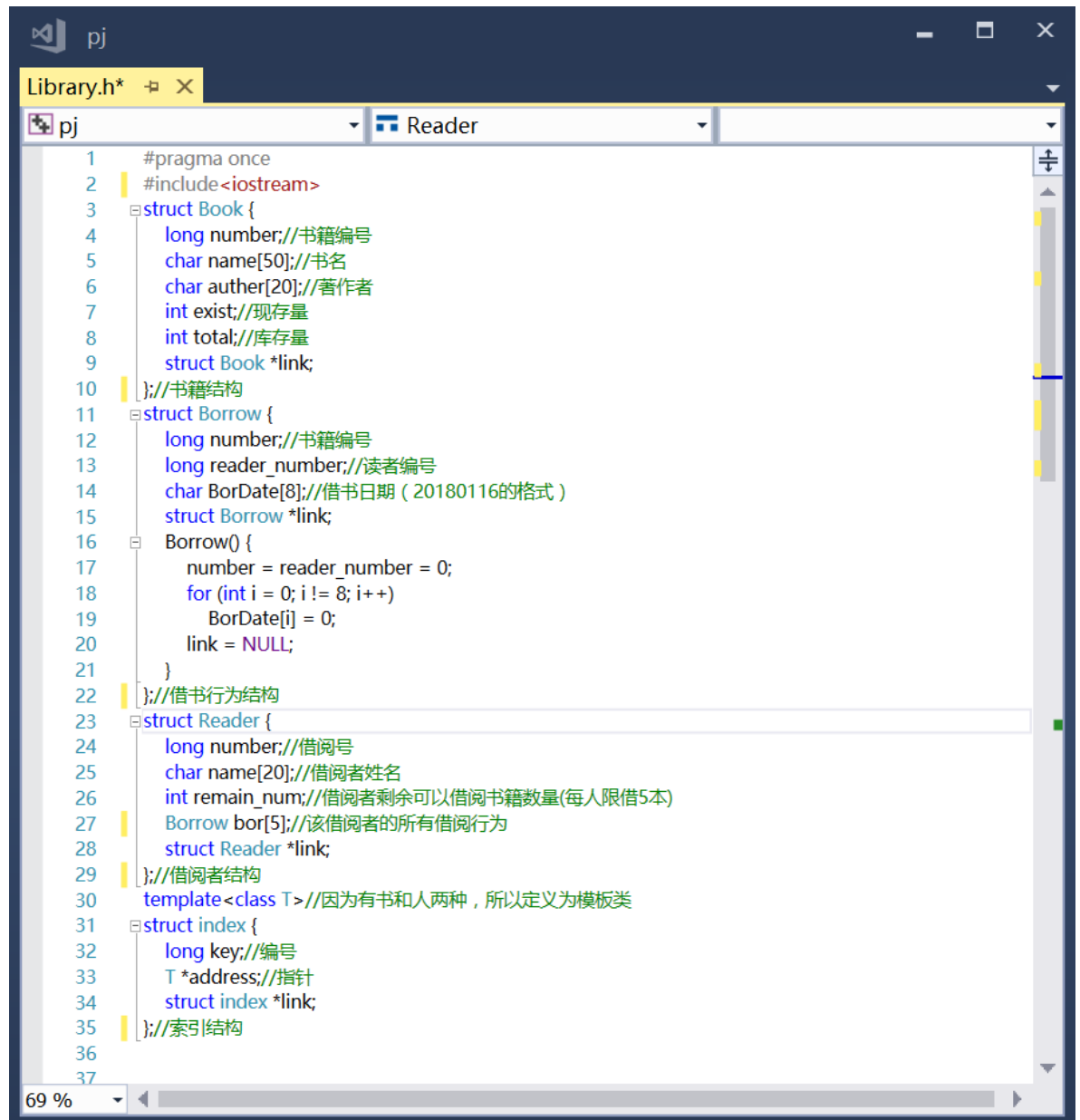
```
void return_book(Book *Bhead, Reader *Rhead, Borrow *B2head)
```



#### 四、详细设计

##### 1、Library.h

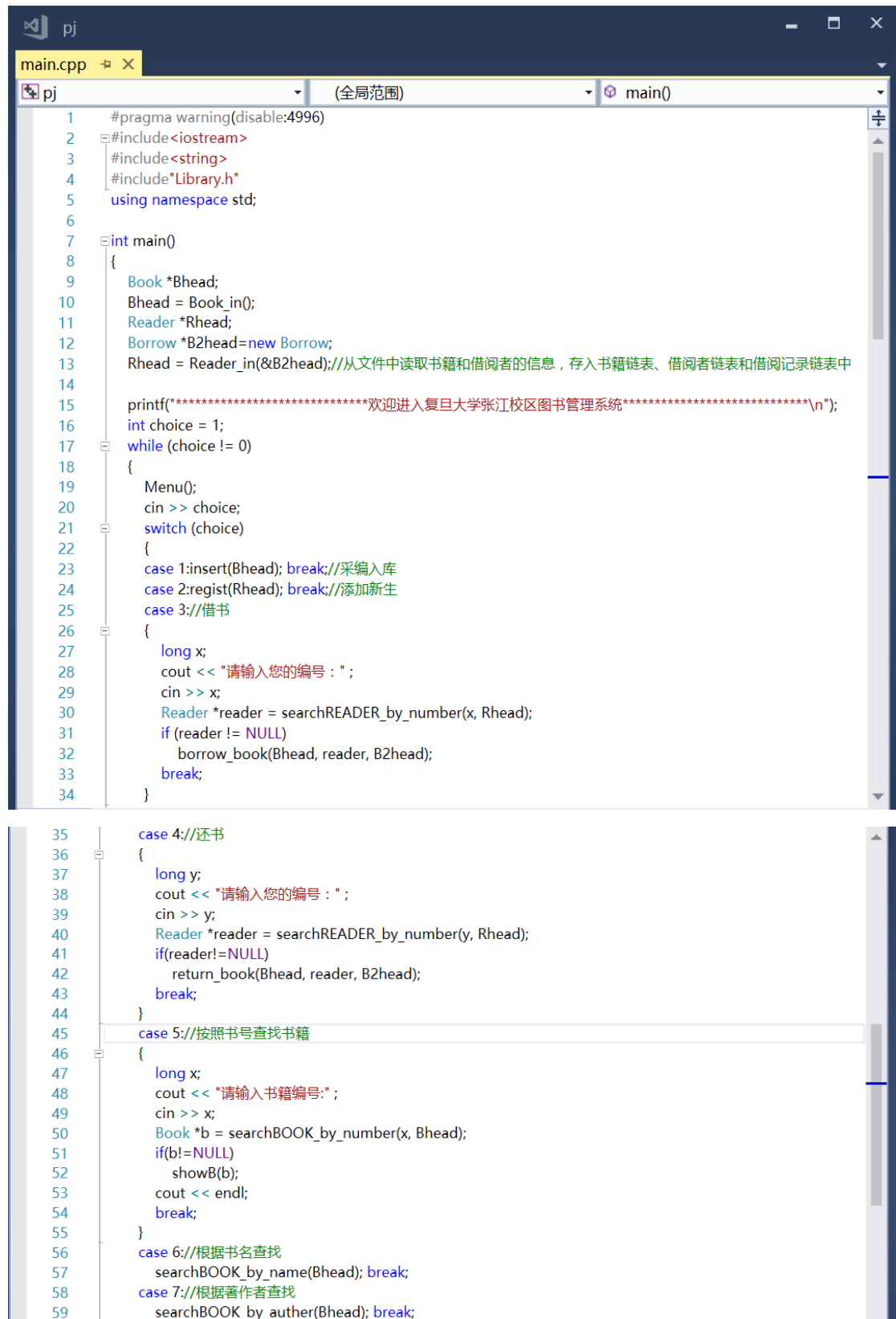
定义了四个结构



```
1  #pragma once
2  #include<iostream>
3  struct Book {
4      long number;//书籍编号
5      char name[50];//书名
6      char auther[20];//著作者
7      int exist;//现存量
8      int total;//库存量
9      struct Book *link;
10 };//书籍结构
11 struct Borrow {
12     long number;//书籍编号
13     long reader_number;//读者编号
14     char BorDate[8];//借书日期 ( 20180116的格式 )
15     struct Borrow *link;
16     Borrow() {
17         number = reader_number = 0;
18         for (int i = 0; i != 8; i++)
19             BorDate[i] = 0;
20         link = NULL;
21     }
22 };//借书行为结构
23 struct Reader {
24     long number;//借阅号
25     char name[20];//借阅者姓名
26     int remain_num;//借阅者剩余可以借阅书籍数量(每人限借5本)
27     Borrow bor[5];//该借阅者的所有借阅行为
28     struct Reader *link;
29 };//借阅者结构
30 template<class T>//因为有书和人两种，所以定义为模板类
31 struct index {
32     long key;//编号
33     T *address;//指针
34     struct index *link;
35 };//索引结构
36
37
```

## 2、main 函数

可以选择九个功能



```
1  #pragma warning(disable:4996)
2  #include<iostream>
3  #include<string>
4  #include"Library.h"
5  using namespace std;
6
7  int main()
8  {
9      Book *Bhead;
10     Bhead = Book_in();
11     Reader *Rhead;
12     Borrow *B2head=new Borrow;
13     Rhead = Reader_in(&B2head);//从文件中读取书籍和借阅者的信息，存入书籍链表、借阅者链表和借阅记录链表中
14
15     printf("*****欢迎进入复旦大学张江校区图书管理系统*****\n");
16     int choice = 1;
17     while (choice != 0)
18     {
19         Menu();
20         cin >> choice;
21         switch (choice)
22         {
23             case 1:insert(Bhead); break;//采编入库
24             case 2:regist(Rhead); break;//添加新生
25             case 3://借书
26             {
27                 long x;
28                 cout << "请输入您的编号：" ;
29                 cin >> x;
30                 Reader *reader = searchREADER_by_number(x, Rhead);
31                 if (reader != NULL)
32                     borrow_book(Bhead, reader, B2head);
33                 break;
34             }
35
36             case 4://还书
37             {
38                 long y;
39                 cout << "请输入您的编号：" ;
40                 cin >> y;
41                 Reader *reader = searchREADER_by_number(y, Rhead);
42                 if(reader!=NULL)
43                     return_book(Bhead, reader, B2head);
44                 break;
45             }
46
47             case 5://按照书号查找书籍
48             {
49                 long x;
50                 cout << "请输入书籍编号:" ;
51                 cin >> x;
52                 Book *b = searchBOOK_by_number(x, Bhead);
53                 if(b!=NULL)
54                     showB(b);
55                 cout << endl;
56                 break;
57             }
58
59             case 6://根据书名查找
60             searchBOOK_by_name(Bhead); break;
61
62             case 7://根据作者查找
63             searchBOOK_by_auther(Bhead); break;
```



```

60     case 8://按照借阅号查找借阅者
61     {
62         long y;
63         cout << "请输入借阅者编号 : " ;
64         cin >> y;
65         Reader *r = searchREADER_by_number(y, Rhead);
66         if(r!=NULL)
67             showR(r,Bhead);
68         cout << endl;
69         break;
70     }
71     case 9://显示所有逾期未还的借阅信息
72     showFine(B2head,Rhead);
73     case 0://退出系统
74     break;
75     default:
76     cout << "请输入正确的序号 ! " << endl; break;
77     }
78     }
79
80     system("pause");
81     return 0;
82 }

```

3、file\_in.cpp 从文件中读取信息，存入链表中  
先读 book.txt, 存入书籍链表中

```

10 Book* Book_in()
11 {
12     FILE *p;
13     Book *Bhead = new Book;
14     Bhead->link = NULL;
15     Book *b1, *b2;
16     if ((p = fopen("book.txt", "r")) == NULL)
17     {
18         cout << "打开文件失败 ! " << endl;
19         return NULL;
20     }
21     else
22     {
23         char s[200];
24         fgets(s, 100, p);//跳过第一行
25         b1 = new Book;
26         fscanf(p, "%d%s%s%d\n", &b1->number, b1->name, b1->author, &b1->exist, &b1->total);
27         b1->link = NULL;
28         Bhead = b1;
29         while (!feof(p))
30         {
31             b2 = new Book;
32             fscanf(p, "%d%s%s%d\n", &b2->number, b2->name, b2->author, &b2->exist, &b2->total);
33             b2->link = NULL;
34             b1->link = b2;
35             b1 = b1->link;
36         }
37         cout << "...书籍信息读取成功 ! " << endl;
38         return Bhead;
39     }
40 }

```

再读 reader.txt, 存入借阅者和借阅行为链表中

```
60 Reader* Reader_in(Borrow** B2head)
61 {
62     FILE *p;
63     Reader *Rhead = new Reader;
64     Rhead->link = NULL;
65     (*B2head)->link = NULL;
66     Reader *r1, *r2;
67     Borrow *b1, *b2, *b3;
68     b3 = NULL;
69     if ((p = fopen("reader.txt", "r")) == NULL)
70     {
71         cout << "打开文件失败！" << endl;
72         return NULL;
73     }
74     else
75     {
76         char s[200];
77         fgets(s, 100, p); //跳过第一行
78         r1 = new Reader;
79         fscanf(p, "%d%s%d", &r1->number, &r1->name, &r1->remain_num); //先读编号、姓名、剩余可借数
80         for (int i = 0; i != 5; i++) //再读已借书籍 (上限5本)
81         {
82             b1 = new Borrow;
83             fscanf(p, "%d", &(r1->bor[i].number));
84             fscanf(p, "%s", &r1->bor[i].BorDate);
85             if (r1->bor[i].number != 0)
86             {
87                 b1->number = r1->bor[i].number; //同时读到借阅表中
88                 b1->reader_number = r1->number;
89                 for (int j = 0; j != 8; j++)
90                     b1->BorDate[j] = r1->bor[i].BorDate[j];
91                 b1->link = NULL;
92                 if (i == 0)
93                     *B2head = b1;
94                 else
95                 {
96                     b3 = *B2head;
97                     while (b3->link != NULL)
98                         b3 = b3->link;
99                     b3->link = b1;
100                 }
101             }
102         }
103         r1->link = NULL;
104         Rhead = r1;
105         while (!feof(p))
106         {
107             r2 = new Reader;
108             b2 = new Borrow;
109             fscanf(p, "%d%s%d", &r2->number, &r2->name, &r2->remain_num);
110             for (int i = 0; i != 5; i++)
111             {
112                 fscanf(p, "%d%s", &r2->bor[i].number, &r2->bor[i].BorDate);
113                 if (r2->bor[i].number != 0)
114                 {
115                     b2->number = r2->bor[i].number;
116                     b2->reader_number = r2->number;
117                     for (int j = 0; j != 8; j++)
118                         b2->BorDate[j] = r2->bor[i].BorDate[j];
119                     b2->link = NULL;
120                     b3 = *B2head;
121                     if (b3 == NULL)
122                         *B2head = b2;
123                     else
124                     {
125                         while (b3->link != NULL)
```

```

126         b3 = b3->link;
127         b3->link = b2;
128     }
129 }
130 }
131 r2->link = NULL;
132 r1->link = r2;
133 r1 = r1->link;
134 }
135 cout << "...借阅者信息读取成功 !" << endl;
136 return Rhead;
137 }
138 }

```

#### 4、search.cpp 书籍和借阅者查找

首先，按照前面构思部分建立索引表

```

5 //用哈希函数建立索引表，用开散列法解决冲突
6 template<class T>
7 index<T> ** init_index(T *head)
8 {
9     index<T> ** Hash_Table;
10    Hash_Table = new index<T> * [97];
11    for (int i = 0; i != 97; i++)
12    {
13        Hash_Table[i] = NULL;
14    }
15    T *temp = head;
16    index<T> *p, *q;
17    while (temp != NULL)
18    {
19        p = new index<T>;
20        p->key = temp->number;
21        p->address = temp;
22        p->link = NULL;
23        long re = p->key % 97;
24        q = Hash_Table[re];
25        if (q == NULL)
26        {
27            q = p;
28            Hash_Table[re] = q;
29        }
30        else
31        {
32            while (q != NULL && q->link != NULL)
33                q = q->link;
34            q->link = p; //插入对应链表中
35        }
36        temp = temp->link;
37    }
38    return Hash_Table;
39 }

```

按照书号从哈希表中查找书籍，返回指向书籍的指针

```
41 //按照编号查找书籍
42 Book* searchBOOK_by_number(long num, Book *Bhead)
43 {
44     index<Book> **table = init_index(Bhead);
45     index<Book> *q;
46     int flag = 0;
47     for (int i = 0; i != 97; i++)
48     {
49         q = table[i];
50         if (q != NULL)
51         {
52             flag = 1; break;
53         }
54     }
55     if (flag == 0)
56     {
57         cout << "书库无记录，请先输入！" << endl;
58         return NULL;
59     }
60     else
61     {
62         long re = num % 97;
63         q = table[re];
64         if (q == NULL)
65         {
66             cout << "没有找到编号为" << num << "的书籍！" << endl;
67             return NULL;
68         }
69         else
70         {
71             while (q != NULL)
72             {
73                 if (q->key == num)
74                     return q->address;
75                 else q = q->link;
76             }
77             if (q == NULL)
78             {
79                 cout << "没有找到编号为" << num << "的书籍！" << endl;
80                 return NULL;
81             }
82         }
83     }
84     return NULL;
85 }
```

按照书名和作者从链表中查找书籍

```
86 void searchBOOK_by_name(Book *Bhead)//根据书名查找书籍
87 {
88     cout << "输入想要查找的书名：" ;
89     char name[50];
90     cin >> name;
91     Book *b = Bhead;
92     int flag = 0;
```

```

94     {
95         if (strcmp(name, b->name) == 0)
96         {
97             showB(b);
98             b = b->link;
99             flag = 1;
100             cout << endl;
101         }
102         else b = b->link;
103     }
104     if (flag == 0)
105         cout << "没有找到书名为"<<name<<"的书籍 !" << endl;
106 }
107 void searchBOOK_by_auther(Book *Bhead)//根据作者查找书籍
108 {
109     cout << "请输入想要查找的作者 : ";
110     char auther[20];
111     cin >> auther;
112     Book *b = Bhead;
113     int flag = 0;
114     while (b != NULL)
115     {
116         if (strcmp(auther, b->auther) == 0)
117         {
118             showB(b);
119             b = b->link;
120             flag = 1;
121             cout << endl;
122         }
123         else b = b->link;
124     }
125     if (flag == 0)
126         cout << "没有找到作者为" << auther << "的书籍 !" << endl;
127 }

```

按照借阅号在哈希表中查找借阅者

```

128 //按照借阅号查找借阅者
129 Reader* searchREADER_by_number(long num, Reader *Rhead)
130 {
131     index<Reader> **table = init_index(Rhead);
132     index<Reader> *q;
133     int flag = 0;
134     for (int i = 0; i != 97; i++)
135     {
136         q = table[i];
137         if (q != NULL)
138         {
139             flag = 1; break;
140         }
141     }

```

```

142     if (flag == 0)
143     {
144         cout << "借阅者者库无记录，请先输入！" << endl;
145         return NULL;
146     }
147     else
148     {
149         long re = num % 97;
150         q = table[re];
151         if (q == NULL)
152         {
153             cout << "没有找到编号为" << num << "的借阅者！" << endl;
154             return NULL;
155         }
156         else
157         {
158             while (q != NULL)
159             {
160                 if (q->key == num)
161                     return q->address;
162                 else q = q->link;
163             }
164             if (q == NULL)
165             {
166                 cout << "没有找到编号为" << num << "的借阅者！" << endl;
167                 return NULL;
168             }
169         }
170     }
171     return NULL;
172 }

```

##### 5、insert\_register.cpp 采编入库和新生注册

采编入库如上面流程图所示

```

6     //采编入库
7     void insert(Book *Bhead)
8     {
9         printf("\n***采编入库***\n");
10        Book *b1,*b2,*b3;
11        long x,y;
12        b3 = Bhead;
13        cout << "请输入书籍数量:" ;
14        cin >> y;
15        while (1)
16        {
17            cout << "请指定将入库书的编号（四位以内整数）:" ;
18            cin >> x;
19            if (x <= 0 || x >= 9999)
20            {

```

```

21         cout << "boom!标号超出处理范围！" << endl;
22         continue;
23     }
24     else
25     {
26         b2 = searchBOOK_by_number(x, Bhead); //先判断该编号是否已存在
27         if (b2 == NULL) //如果不存在，跳出循环，添加新项
28             break;
29         else
30         {
31             b2->total += y;
32             b2->exist += y;
33             printf("***编号为%d的书籍已存在，入库成功！***\n", x);
34             return;
35         }
36     }
37 }
38 cout << "添加新书！" << endl;
39 b1 = new Book; //添加新书籍
40 b1->number = x;
41 cout << "请输入书名：";
42 scanf("%s", b1->name);
43 cout << "请输入作者：";
44 scanf("%s", b1->author);
45 b1->link = NULL;
46 b1->total = y;
47 b1->exist = y;
48 while (b3->link != NULL)
49     b3 = b3->link;
50 b3->link = b1;
51 printf("***新书入库成功***\n");
52 }

```

添加新生时，由用户输入姓名，遍历读者链表得到目前总读者数，然后加一返回新生的借阅号。

```

53 //注册新生
54 void regist(Reader *Rhead)
55 {
56     printf("\n***注册***\n");
57     Reader *r, *r1;
58     r = new Reader;
59     r1 = Rhead;
60     long count = 1;
61     cout << "请输入姓名：";
62     scanf("%s", r->name);
63     r->remain_num = 5;
64     for (int i = 0; i != 5; i++)
65     {
66         r->bor[i].number = 0;
67         r->bor[i].reader_number = 0;
68         for (int j = 0; j != 8; j++)
69             r->bor[i].BorDate[j] = ' ';
70     }

```

```

71     while (r1->link != NULL)
72     {
73         r1 = r1->link;
74         count++;
75     }
76     count++;
77     r->number = count;
78     r->link = NULL;
79     r1->link = r;
80     printf("您的借阅号为%d\n", count);
81     printf("***注册成功***\n");
82 }

```

## 6、borrow\_return.cpp 借阅归还

借书还书流程如上图所示

```

7     //借书
8     void borrow_book(Book *Bhead, Reader *reader, Borrow *B2head)
9     {
10        while (1)
11        {
12            printf("\n***借书***\n");
13            cout << "请输入您要借阅的书籍编号：";
14            long x;
15            cin >> x;
16            Book *b = searchBOOK_by_number(x, Bhead);
17            if (b != NULL)//该书在库中存在
18            {
19                showB(b);//显示该书信息
20                cout << "确认借阅此书籍吗？（请回答Y/N）";
21                cin.clear();//清空输入流
22                cin.ignore();
23                char c1 = getchar();
24                char c2 = getchar();
25                if (c1 == 'N')//如果不借，返回选择界面
26                    return;
27            }
28            else
29            {
30                if (reader->remain_num == 0)
31                {
32                    cout << "勤奋的您已借满五本书，先研究完手里的吧！" << endl;
33                    break;
34                }
35                else if (b->exist == 0)
36                {
37                    cout << "对不起您来晚了~该书已无库存，请看看其它书吧！" << endl;
38                }
39                else//当库存充足且借阅者未借满五本时
40                {
41                    //改变借阅者链表
42                    reader->remain_num--;

```



```

40         int i = 0, j = 0;
41         for (i; i != 5 && reader->bor[i].number != 0; i++);
42         reader->bor[i].number = x;
43         reader->bor[i].reader_number = reader->number;
44         cout << "请输入当前时间 : ";
45         cin.clear();
46         cin.ignore();
47         string s;
48         cin >> s;
49         for (; j != 8; j++)
50             reader->bor[i].BorDate[j] = s[j];
51         //改变书籍链表
52         b->exist--;
53         //改变借阅记录链表
54         Borrow *b2, *b3;
55         b2 = B2head;
56         b3 = new Borrow;
57         b3->number = x;
58         j = 0;
59         b3->BorDate[j++] = reader->bor[i].BorDate[j++];
60         b3->link = NULL;
61         b3->reader_number = reader->number;
62         while (b2->link != NULL)
63             b2 = b2->link;
64         b2->link = b3;
65     }
66 }
67 printf("***借阅成功***\n");
68 }
69 cout << "您要继续借阅其它书籍吗? (请回答Y/N) ";
70 cin.clear();
71 cin.ignore();
72 char t = getchar();

```

```

73     char t2 = getchar();
74     if (t == 'N')
75         break;
76 }
77 }
78
79 //还书
80 void return_book(Book *Bhead, Reader *reader, Borrow *B2head)
81 {
82     printf("\n***还书***\n");
83     cout << "请输入您要还的书籍编号 : ";
84     long x;
85     cin >> x;
86     int i, j;
87     for (i = 0; i != 5; i++)
88     {
89         if (reader->bor[i].number == x)
90         {
91             j = 1;
92             break;
93         }
94     }

```

```

95  Book *b = searchBOOK_by_number(x, Bhead);
96  if (b != NULL && j == 1)
97  {
98      printf(" \n");
99      printf(" | 书籍编号 | 书名 | 著作者 | 借书时间 | \n");
100     printf(" |-----|-----|-----|-----| \n");
101     printf(" | %16d | %16s | %16s | %16s | \n", b->number, b->name, b->author, reader->bor[i].BorDate);
102     printf(" |-----|-----|-----|-----| \n");
103     cout << "确认归还此书吗? (请回答Y/N) ";
104     cin.clear();
105     cin.ignore();
106     char ch = getchar();
107     char ch2 = getchar();
108     if (ch == 'Y')
109     {
110         //改变书籍链表
111         reader->remain_num++;
112         for (int i = 0; i != 5; i++)
113         {
114             if (reader->bor[i].number == x)
115             {
116                 reader->bor[i].number = 0;
117                 for (int j = 0; j != 8; j++)
118                     reader->bor[i].BorDate[j] = 0;
119                 break;
120             }
121         }
122         //改变读者链表
123         b->exist++;
124         //改变借阅记录链表
125         Borrow *b1 = B2head;
126         if (b1->number == x)
127         {
128             B2head = b1->link;
129             delete b1;
130         }
131         else
132         {
133             while (b1 != NULL)
134             {
135                 if (b1->link->number == x)
136                     break;
137                 else b1 = b1->link;
138             }
139             Borrow *b2 = b1->link;
140             b1->link = b2->link;
141             delete b2;
142         }
143     }
144     printf("***还书成功***\n");
145 }
146 else cout << "编号有误! 请仔细检查!" << endl;
147 }

```

- 7、fine.cpp 计算借阅时间、判断日期是否合法并打印所有逾期未还的借阅记录  
难点在于当前日期和借阅日期之间相差天数的计算

```
1  #include<iostream>
2  #include"Library.h"
3  using namespace std;
4
5  struct Date {
6      int year;
7      int month;
8      int day;
9  };
10 Date tran(char s[8])
11 {
12     Date date;
13     char s2[8];
14     for (int i = 0; i != 8; i++)
15         s2[i] = s[i] - 48;
16     date.year = s2[0]* 1000 + s2[1] * 100 + s2[2] * 10 + s2[3];
17     date.month = s2[4] * 10 + s2[5];
18     date.day = s2[6] * 10 + s2[7];
19     return date;
20 }
21
22 int isLeapYear(int year)//判断是否是闰年
23 {
24     return year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
25 }
26 int getDayOfMonth(int year,int month)//求某年某月的天数
27 {
28     int dayArr[] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
29     return month == 2 && isLeapYear(year) ? dayArr[month] + 1 : dayArr[month];
30 }
31 int isLegal(char s[8])//判断日期是否合法
32 {
33     Date d;
34     d=tran(s);
35     if (d.year < 0)
36         return 0;
37     else if (d.month < 1 || d.month>12)
38         return 0;
39     else if (d.day<1 || d.day>getDayOfMonth(d.year, d.month))
40         return 0;
41     else return 1;
42 }
43 int countFine(Reader *reader, Borrow *b,char s[8])//计算此人借阅此书的罚款
44 {
45     Date d1, d2;
46     d1 = tran(b->BorDate);//借阅日期
47     d2 = tran(s);//当前日期
48     long daylen=0;
49     for (int y = d1.year; y != d2.year; y++)
50     {
51         daylen += 365;
52         if (isLeapYear(y))
53             daylen++;
54     }
55     for (int m = 1; m != d2.month; m++)
56         daylen += getDayOfMonth(d2.year, m);
```

```

57     daylen += d2.day;
58     for (int m = 1; m != d1.month; m++)
59         daylen -= getDayOfMonth(d1.year, m);
60     daylen -= d1.day;
61     return daylen;
62 }
63
64 void showFine(Borrow *B2head, Reader *Rhead) //显示所有逾期未还的借阅记录
65 {
66     cout << "请输入当前日期" << endl;
67     char data[8];
68     cin >> data;
69     if (isLegal(data)) //如果当前日期合法
70     {
71         Borrow *b = B2head;
72         int count = 0;
73         while (b != NULL)
74         {
75             //计算日期差
76             int length = countFine(searchREADER_by_number(b->reader_number, Rhead), b, data);
77             if (length > 90) //如果超时 ( 三个月 )
78             {
79                 count++;
80                 int money = length - 90; //罚款
81
82                 if (count == 1)
83                 {
84                     printf(" |-----|\n");
85                     printf(" | 书籍编号 | 借阅者编号 | 借阅日期 | 借阅天数 | 罚金 | \n");
86                     printf(" |-----|\n");
87                     printf(" |%16d| %16d| %16s| %16d| %16d| \n", b->number, b->reader_number, b->BorDate, length, money);
88                 }
89                 else
90                 {
91                     printf(" |-----|\n");
92                     printf(" |%16d| %16d| %16s| %16d| %16d| \n", b->number, b->reader_number, b->BorDate, length, money);
93                 }
94                 b = b->link;
95             }
96             printf(" |-----|\n");
97         }
98     }
99     else
100         cout << "输入日期错误 !" << endl;
}

```

## 8、show.cpp

此模块负责打印内容，包括查询到的书籍信息、读者信息以及选择菜单

显示书籍

```

6 void showB(Book *p)
7 {
8     printf(" |-----|\n");
9     printf(" | 书籍编号 | 书名 | 著作者 | 现存量 | 库存量 | \n");
10    printf(" |-----|\n");
11    printf(" |%16d| %16s| %16s| %16d| %16d| \n", p->number, p->name, p->author, p->exist, p->total);
12    printf(" |-----|\n");
13 }

```

## 显示借阅者

```

15 void showR(Reader *p, Book *Bhead)
16 {
17     printf("      |      |      |      |\n");
18     printf("      |      | 读者信息  |      |\n");
19     printf("      |      |      |      |\n");
20     printf("      | 借阅证号 | 姓名 | 剩余可借 | \n");
21     printf("      |      |      |      |\n");
22     printf(" | %16d | %16s | %16d | \n", p->number, p->name, p->remain_num);
23     printf("      |      |      |      |\n");
24     if (p->remain_num == 5)
25     {
26         cout << "这个人账下空空，竟然一本都没有借！" << endl;
27         return;
28     }
29     printf("所借书籍信息\n");
30     printf("      |      |      |      |      |\n");
31     printf("      | 书籍编号 | 书名 | 著作者 | 借书时间 | \n");
32     printf("      |      |      |      |      |\n");
33     for (int i = 0; i != 5 - p->remain_num; i++)
34     {
35         if (p->bor[i].number == 0 || p->bor[i].BorDate == 0)
36             break;
37         else
38         {
39             Book *q = searchBOOK_by_number(p->bor[i].number, Bhead);
40             printf(" | %16d | %16s | %16s | %16s | \n", q->number, q->name, q->auther, p->bor[i].BorDate);
41             if (i != 5 - p->remain_num - 1)
42                 printf("      |      |      |      |      |\n");
43         }
44     }
45     printf("      |      |      |      |      |\n");
46 }

```

## 显示选择菜单

```

48 void Menu()//选择菜单
49 {
50     cout << endl;
51     printf("      |      |      |      |      | \n");
52     printf("      |      |      |      |      | \n");
53     printf(" | 1、采编入库（新购入一本书，如果已在库，则库存量加一，否则新增一种书，需输入数量和编号） | \n");
54     printf(" | 2、添加新生（注册需输入姓名，产生一个借阅号） | \n");
55     printf(" | 3、借书 | \n");
56     printf(" | 4、还书 | \n");
57     printf(" | 5、按书号查找书籍 | \n");
58     printf(" | 6、按书名查找书籍 | \n");
59     printf(" | 7、按作者查找书籍 | \n");
60     printf(" | 8、按借阅号查找借阅者，并显示其借阅情况 | \n");
61     printf(" | 9、查看所有逾期未还的情况 | \n");
62     printf(" | 10、退出图书馆里系统 | \n");
63     printf("      |      |      |      |      | \n");
64     printf("      |      |      |      |      | \n");
65     cout << endl;
66 }

```

五、调试分析

1、文件

book - 记事本				
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)				
书籍编号	书名	著作者	现存量	库存量
1	数据结构	施伯乐	99	100
115	蟋蟀也吃兴奋剂	张之路	6	7
46	蓝鲸的眼睛	赵冰波	17	17
103	小巴掌童话	张秋生	13	13
79	怒气收集袋	管家琪	1	1
28	复制瞌睡羊	口水龙	48	48
127	捉拿古奇台风	糊涂头鬼	44	44
81	皮皮鲁传	郑渊洁	5	5
89	鲁西西传	郑渊洁	36	36
51	幽默三国	周锐	40	40
153	魔法学校(系列)	葛竞	36	36
177	李大米和他的影子	张之路	2	2
74	五(三)班的坏小子	杨红樱	4	4
108	爷爷铁床下的密室	车培晶	22	22
53	肚皮上的塞子	周锐	30	30
64	三个小宠物	肖定丽	36	36
147	小老虎历险记	汤素兰	2	2
59	塌鼻子警察	庄大伟	5	5
41	绿刺猬	彭懿	10	10
172	阿笨猫全传	赵冰波	5	5
193	嘭嘭嘭	童喜喜	12	12
4	出窍	周锐	10	10
198	西游记	吴承恩	32	32
121	水浒传	施耐庵	45	45
178	苦儿流浪记	埃多克马洛(法)	31	31
49	蓝色海豚岛	斯.奥台尔(美)	6	6
148	时代广场的蟋蟀	乔治塞尔登(美)	11	11
17	随风而来玛丽阿姨	特拉弗斯(英)	49	49
199	鲁滨逊漂流记	笛福(英)	26	26
52	当世界年纪还小时	于尔克.舒比格(法)	14	14
186	柳树间的风	格雷厄姆(英)	28	28
94	水孩子	查.金斯莱(英)	28	28
2	杜利特医生故事	休.洛夫廷(美)	22	22
110	哈利伯特与魔法石	J.K.罗琳(英)	12	12
50	夏洛的网	怀特(美)	11	11
19	长袜子皮皮	林格伦(瑞典)	22	22
61	格列佛游记	斯威夫特(英)	40	40
92	爱的教育	阿米琪斯(意)	5	5
42	草原上的小木屋	罗兰.英格斯.怀德(美)	48	48
159	西顿动物故事集	西顿(加)	43	43
116	女巫	罗尔德.达尔(美)	33	33
194	鸟雀街上的孤岛	尤里.奥莱夫(以)	40	40
3	豪夫童话	豪夫(德)	18	18
63	丛林之书	吉朴林(英)	22	22
30	窗边的小豆豆	黑柳彻子(日)	33	33
16	彼得潘	詹姆斯巴里(苏)	33	33

reader - 记事本				
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)				
编号	姓名	剩余可借数	1编号	借阅日期
77	祁佳薇	4	1	20170909
115		20180101	0	00000000
0		00000000	0	00000000
0		00000000	0	00000000

## 2、初始界面

```
...书籍信息读取成功!
...借阅者信息读取成功!
*****欢迎进入复旦大学张江校区图书管理系统*****

-----MENU-----

1、采编入库（新购入一本书，如果已在库，则库存量加一，否则新增一种书，需输入数量和编号）
2、添加新生（注册需输入姓名，产生一个借阅号）
3、借书
4、还书
5、按书号查找书籍
6、按书名查找书籍
7、按作者查找书籍
8、按借阅号查找借阅者，并显示其借阅情况
9、查看所有逾期未还的情况
0、退出图书馆里系统

-----请输入你需要的操作-----
```

## 3、采编入库

### 新书

```
1
***采编入库***
请输入书籍数量:5
请指定将入库书的编号（200以内整数）:23
没有找到编号为23的书籍!
添加新书!
请输入书名: 离散数学
请输入作者: 赵一鸣
***新书入库成功***
```

### 旧书

```
1
***采编入库***
请输入书籍数量:1
请指定将入库书的编号（200以内整数）:1
***编号为1的书籍已存在，入库成功!***
```

## 3、注册新生

```
2
***注册***
请输入姓名: 杨希希
您的借阅号为2
***注册成功***
```

## 4、借书

```
3
请输入您的编号: 2

***借书***
请输入您要借阅的书籍编号: 1
```

书籍编号	书名	著作者	现存量	库存量
1	数据结构	施伯乐	100	101

```
确认借阅此书籍吗?（请回答Y/N） Y
请输入当前时间: 20180119
***借阅成功***
您要继续借阅其它书籍吗?（请回答Y/N） N
```

## 5、还书

4

请输入您的编号：2

\*\*\*还书\*\*\*

请输入您要还的书籍编号：1

书籍编号	书名	著作者	借书时间
1	数据结构	施伯乐	0180119

确认归还此书吗？（请回答Y/N） Y

\*\*\*还书成功\*\*\*

## 6、查找书籍

5

请输入书籍编号:1

书籍编号	书名	著作者	现存量	库存量
1	数据结构	施伯乐	100	101

6

输入想要查找的书名：数据结构

书籍编号	书名	著作者	现存量	库存量
1	数据结构	施伯乐	100	101

7

请输入想要查找的作者：施伯乐

书籍编号	书名	著作者	现存量	库存量
1	数据结构	施伯乐	100	101

## 7、查找借阅者

8

请输入借阅者编号：77

	读者信息	
借阅证号	姓名	剩余可借
77	祁佳薇	4

所借书籍信息

书籍编号	书名	著作者	借书时间
1	数据结构	施伯乐	20170909

8

请输入借阅者编号：2

	读者信息	
借阅证号	姓名	剩余可借
2	杨希希	5

这个人账下空空，竟然一本都没有借！



## 8、查看所有逾期未还的书籍

9  
请输入当前日期  
20190101

书籍编号	借阅者编号	借阅日期	借阅天数	罚金
1	77	20170909H<□	480	390
115	77	20180101	365	275

## 9、分析

### (1) 时间复杂度

本次设计时间复杂度主要体现在查找上，不同索引技术对查询效率的影响见本报告开始部分，由于时间较紧，未能按计划再用树形索引来实现。

### (2) 做题思路

看到图书馆问题，首先想到考察的是存储和查找，因为藏书数量巨大，所以必须建立索引，经过如前面所述的比较选择，最后选定哈希表建立索引，并用开散列法解决冲突。确立思路后，先写 search 函数，文件读写，然后是采编入库、借阅归还三大功能，最后是打印部分。

在完成的过程中因为各个函数关联性较大，所以一直到全部写完才开始测试，结果出现不少错误，前面代码也有所忘记，此时我也真正认识到了注释的重要作用。因为思路较为清晰，所以并未遇到大的问题，主要是一些小问题，比如：char 型数组赋值、链表中元素插入删除时指针的改写等。

### (3) 有待改进

- ① 如上图所示，罚款部分输出总是出现问题，日期后面字符不明
- ② 在写完之后意识到哈希表的位置放置不妥，应该是和链表并列，在采集借阅归还时同时改变哈希表的内容，而不是只有在查找的时候才根据链表的改变刷新内容，这样每次查找都要重新哈希，重复多次，这一点需要改进。

## 六、课程设计感想收获

这是第一次独立完成整个 project，在做之前有些担心，但在审完题、确定好思路、开始编写后又感慨编程的乐趣和计算机技术对生活的影响之大。在熟悉了数据结构课程的内容之余，为下学期数据库的学习打下基础。

## 七、参考资料

- 1、《数据结构教程》施伯乐，复旦大学出版社
- 2、《数据结构——C++版》殷人昆著，清华大学出版社