

西安工业大学

XI'AN TECHNOLOGICAL UNIVERSITY

课程设计报告

课程名称: C 语言项目实践

专 业: 软件工程

班 级: 23060208

姓 名: 高明杰

学 号: 2023032369

指导教师: 黄姝娟

成 绩:

2024 年 7 月 5 日

西安工业大学课程设计（论文）用纸

课程名称	C 语言项目实践				指导教师	黄姝娟	
课设题目 4	药品销售管理系统						
学号	2023032369	姓名	高明杰	地点	工 1-202	完成时 间	2024/7/1 2024/7/5
设计任务	<p>建立一个药品销售管理系统，利用堆区创建数组(指针+动态内存)实现，具有如下的五大功能模块，具体要求如下：</p> <div><div>药品销售管理系统</div><div><div>输入记录模 块</div><div>查询记录模 块</div><div>更新记录模 块</div><div>统计记录模 块</div><div>输出记录模 块</div></div></div>						
设计要求	<p>1. 输入记录模块 逐个从文件中读取数据初始化数组结构。 药品包含属性：药品名称，品牌，单价，库存量，生产日期（年月日格式），药品简介等。</p> <p>2. 查询记录模块 分别按照药品名称、年份(例如:2023 年以前生产的药品)进行查询，把满足条件的卡信息显示出来。</p> <p>3. 更新记录模块 3.1 修改某一药品的属性信息 3.2 插入一种新药品信息； 3.3 购买药品，根据药品名称购买药品，若药品库存量为 1，则根据药品名称删除信息；若库存量大于 1，则将该药品库存量减 1。</p> <p>1. 统计记录模块 分别按照库存量统计单价>=30 元的所有药品信息，并输出统计结果。</p> <p>2. 显示模块 显示所有药品的信息。</p> <p>3. 输出记录模块 能够将所有药品信息保存到文件中，并能从文件中读取药品信息； 以上为最基本要求，可根据自己的理解在此基础上对各个模块增加和完善相应的功能，使得整个系统能更加接近实际的应用。</p>						
成果要求	<p>1. 设计出完整流程图。</p> <p>2. 编写出全部模块对应的函数。</p> <p>3. 在集成开发工具中调试通过。</p> <p>4. 参加答辩。</p> <p>5. 写出完整、规范的课程设计报告，在指定的时间内提交指导教师。</p>						

设计报告 要求	1. 封面：（格式附后） 2. 课程设计任务书 3. 课程设计报告： (1) 系统总体方案 (2) 设计思路和主要步骤 (3) 各功能模块和流程图 (4) 设计代码 (5) 心得体会
------------	--

药品销售管理系统

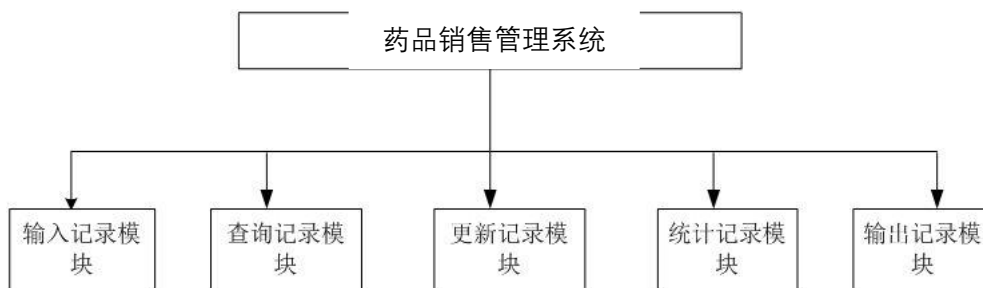
一、课程设计任务书

建立一个药品销售管理系统

二、课程设计报告

2.1 系统总体方案

利用堆区创建数组(指针+动态内存)实现，具有如下的五大功能模块，具体要求如下：



2.2 设计思路和主要步骤

2.2.1 输入记录模块

逐个从文件中读取数据初始化数组结构。药品包含属性：药品名称，品牌，单价，库存量，生产日期（年月日格式），药品简介等。

2.2.2 查询记录模块

分别按照药品名称、年份(例如:2023 年以前生产的药品)进行查询，把满足条件的卡信息显示出来。

2.2.3 更新记录模块

2.3.1 修改某一药品的属性信息

2.3.2 插入一种新药品信息；

2.3.3 购买药品，根据药品名称购买药品，若药品库存量为 1，则根据药

品名称删除信息；若库存量大于 1，则将该药品库存量减 1。

2.2.4 统计记录模块

分别按照库存量统计单价 ≥ 30 元的所有药品信息，并输出统计结果。

2.2.5 显示模块

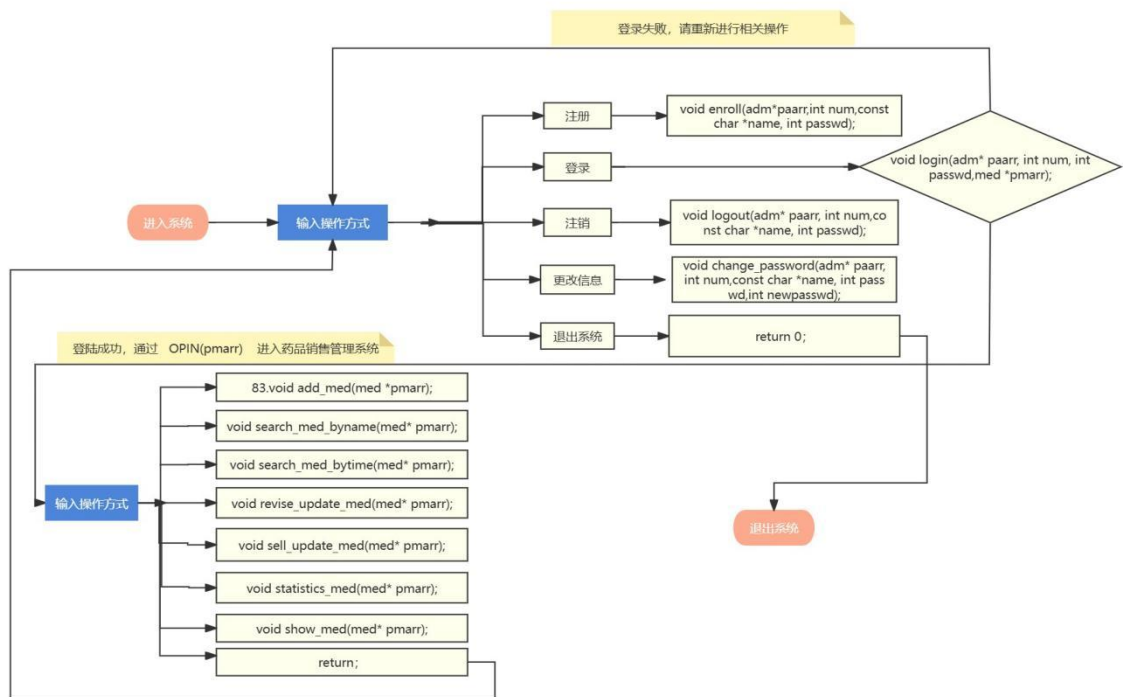
显示所有药品的信息。

2.2.6 输出记录模块

能够将所有药品信息保存到文件中，并能从文件中读取药品信息。

2.3 各功能模块和流程图

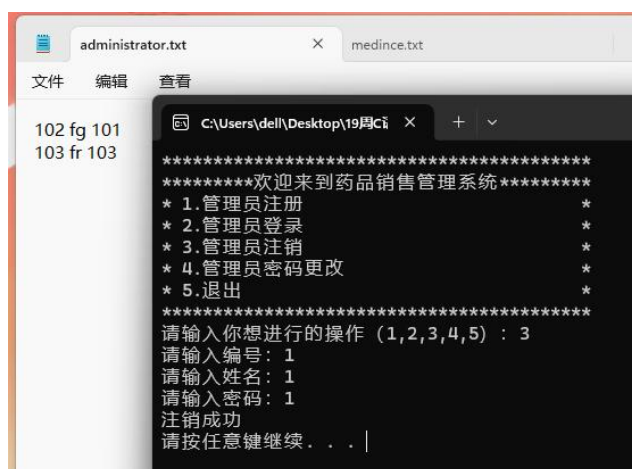
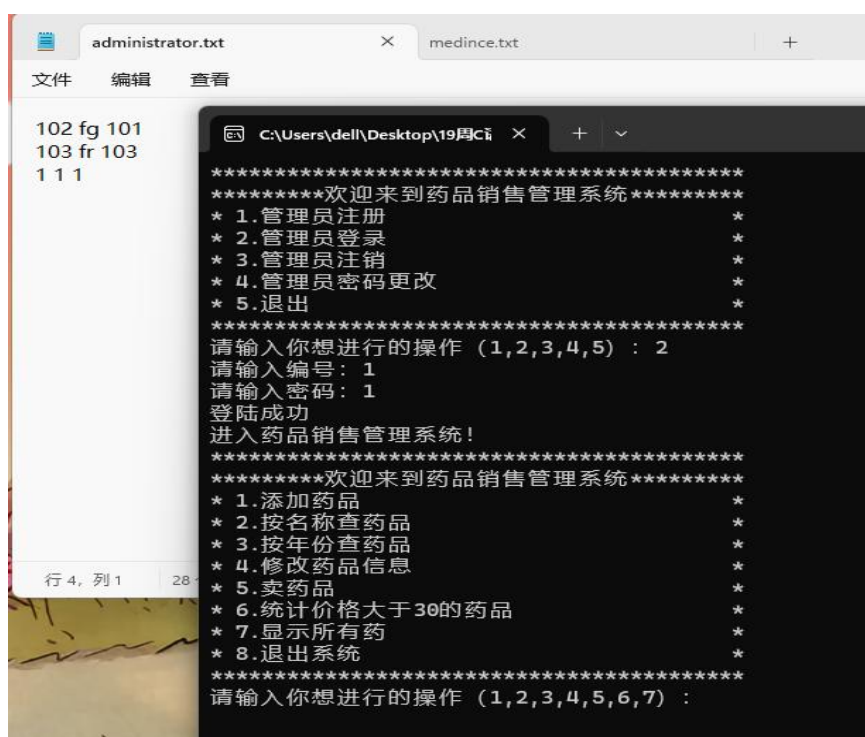
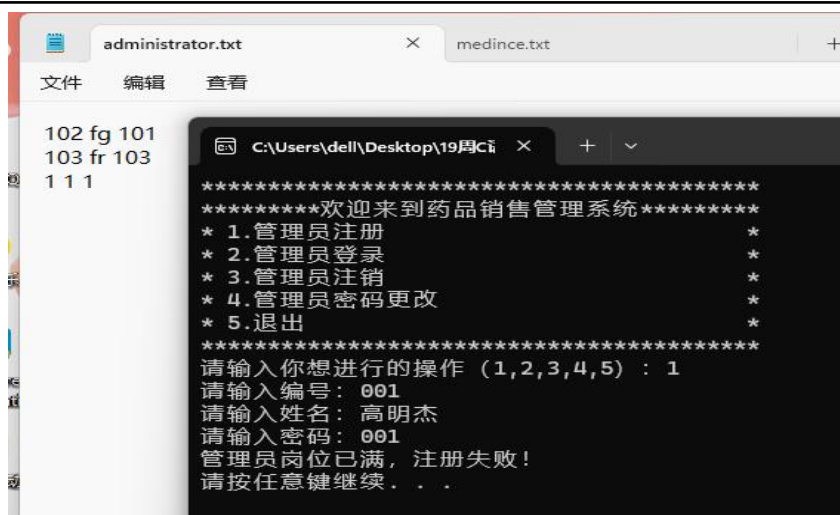
为了进一步提升药品销售管理系统的安全性与合规性，我精心设计并引入了一项关键的进入条件——管理员验证系统。这一创新举措旨在确保系统访问的严格控制，最大限度地保障系统内部数据的安全性与完整性，从而让药品销售管理流程更加可靠、高效。通过实施管理员验证系统，我们不仅能够有效防范未经授权的访问，还能为系统操作增设一道坚实的防线，全方位提升系统的安全防护能力。

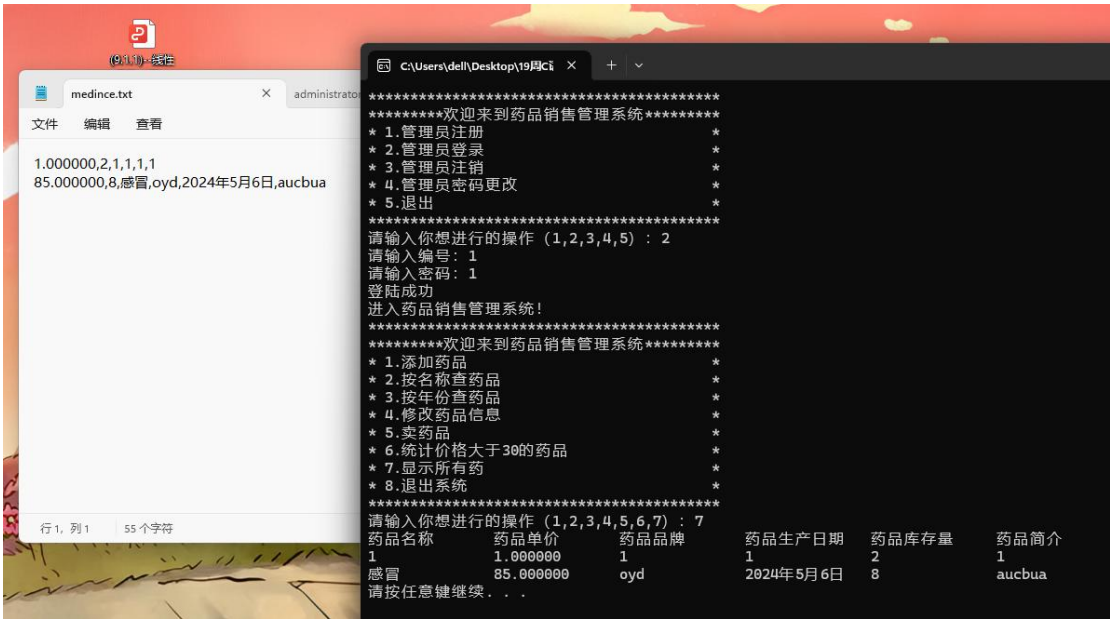
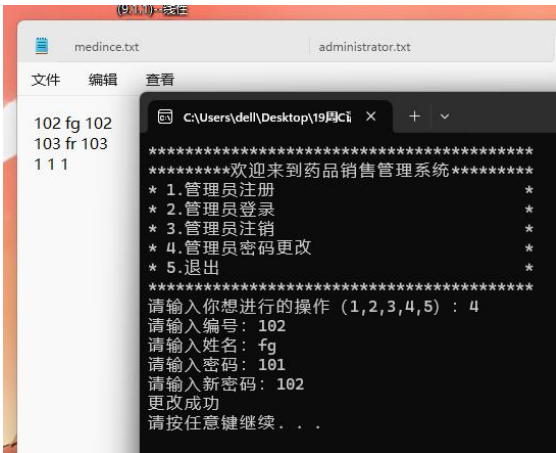
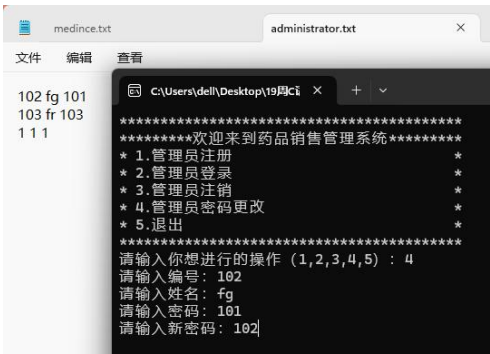


如上是通过 ProcessOn 制作的本药品管理系统的流程图。

2.4 代码编译放在本文末尾

2.5 程序界面





通过全面而详尽的测试与调试流程，我们可以确药品销售管理系统在实际运用中能够稳定流畅地运行，准确高效地处理用户输入的数据。同时，系统在面对各种异常情况时，也具备相应的应对策略，以保障运营的连贯性和安全性。在软件开发过程中，测试与调试是不可或缺的关键环节，它们协助开发者及时发现并解决潜在问题，进而显著提升系统的整体质量和可靠性。一个优秀的程序，必然能够经受住细致入微的推敲与考验。

2.6 总结

药品销售管理系统是一款基于数据结构设计的信息管理系统，其目标是高效管理和操作药品的库存与销售信息。该系统巧妙地运用了 C 语言的数据结构以及动态内存管理功能，为用户提供了丰富的功能函数，支持药品信息的录入、储存、查询及删除等多项操作，从而大大提升了药品销售管理的便捷性和效率。

药品销售管理系统不仅充分实现了基础的信息管理功能，更进一步彰显了在实际软件开发过程中，如何精妙地运用数据结构和算法来构建一个稳定且高效的系统。借助这一系统，药品销售者能够更为便捷地管理药品库存与销售信息，同时，系统管理员也能轻松进行数据统计与分析，从而有力推动业务进展，显著提升运营效率。

2.7 参考文献

1. 谭浩强. *C 程序设计*[M]. 北京：清华大学出版社，2005.
2. 严蔚敏，吴伟民. *数据结构（C 语言版）*[M]. 北京：清华大学出版社，2011.
3. 张海藩，吕凤翥. *C 语言程序设计教程（第 4 版）*[M]. 北京：人民邮电出版社，2010.
4. 王晓东. "医药信息系统设计与实现"[J]. *计算机知识与技术*, 2012, 8(1): 42-44.
5. 李红. "药品销售管理系统的设计与实现"[D]. 河北大学，2015.
6. 李春葆. *数据库原理及应用*[M]. 北京：清华大学出版社，2010.
7. 赵仲民. "基于 C/S 模式的药品库存管理系统的设计与实现"[J]. *计算机技术与发展*, 2011, 21(3): 164-167.
8. 张晓峰. "药品供应链管理信息系统的设计与开发"[J]. *计算机系统应用*, 2009, 18(7): 112-115.
9. 陈立波. "药品销售管理软件的设计与实现"[D]. 浙江大学，2008.
10. 李晓明. "药品流通领域的供应链管理研究"[J]. *中国医药导刊*, 2013, 5(2): 222-224.
11. 赵丽华. "药品销售管理系统需求分析与设计"[J]. *计算机知识与技术*, 2010, 6(3): 304-306.
12. 王海燕. "药品销售管理信息系统的设计与实现"[D]. 电子科技大学，2014.

附录 代码

项目实践本文将代码分入 5 个文件中

1.Operation.h

```
1.
2. #pragma once
3.
4. #ifndef OPERATION_H
5.
6.
7. #define ADMSIZE 3 //目前设置管理员仅有三人
8. #define MEDINCE 100 //初始药
9.
10.
11. //设计管理员
12. typedef struct administrator {
13.     int a_num; //编号
14.     char a_name[20]; //姓名
15.     int a_passwd; //密码
16. }administrator;
17.
18.
19. //设计药品
20. typedef struct medince {
21.     float m_price; //药品单价
22.     int m_inventory; //药品库存量
23.     char m_name[20]; //药品名称
24.     char m_brand[20]; //药品品牌
25.     char m_manufacture[20]; //药品生产日期
26.     char m_introduction[50]; //药品简介
27.
28.
29. }medince;
30.
31.
32. //管理员数组
33. typedef struct adm {
34.     administrator a_arr[ADMSIZE];
35.     int a_size;
36. }adm;
```



```
31. //药品数组
32. typedef struct med {
33.     medince* pmeds;    //动态内存
34.     int m_size;        //有效药品数量
35.     int medcapacity;   //药品数组容量
36. }med;
37.
38.
39.
40.
41. // File 文件操作
42.
43. //药品文件读入
44. void File_read_med(med* pmarr, const char* filepath);
45.
46. //药品文件写入
47. void File_write_med(med *pmarr, const char* filepath);
48.
49. //管理员文件读入
50. void File_read_adm(adm *paarr, const char* filepath);
51.
52. //管理员文件写入
53. void File_write_adm(adm* paarr, const char* filepath);
54.
55.
56. //Administrator 管理员操作
57.
58. //初始化Administrator 数组
59. void AInit(adm *paarr);
60.
61. //注册
62. void enroll(adm*paarr,int num,const char *name, int passwd);
```

```
63.
64. //登录
65. void login(adm* paarr, int num, int passwd, med *pmarr);
66.
67. //注销
68. void logout(adm* paarr, int num, const char *name, int passwd);
69.
70. //更改密码
71. void change_password(adm* paarr, int num, const char *name, int passwd, int newpasswd);
72.
73.
74. //Medince 药品操作
75.
76. //初始化药品数组
77. void MInit(med* pmarr);
78.
79. //动态内存扩容
80. void Grow(med* pmarr);
81.
82. //添加药品
83. void add_med(med *pmarr);
84.
85. //查询药品
86. void search_med_byname(med* pmarr); // 通过名称
87. void search_med_bytime(med* pmarr); //通过时间
88.
89. //更新药品
90. void revise_update_med(med* pmarr); // 修改药品信息
91. void sell_update_med(med* pmarr); //卖药品
92.
93. // 统计价格（例如统计大于30元药品）
94. void statistics_med(med* pmarr);
```

```

95.
96. //显示药品信息
97. void show_med(med* pmarr);
98.
99.
100.
101. void show1();    //登录前页面展示
102.
103. void OPIN(med *pmarr);    //药品管理系统进入
104.
105. void show2();    //药品操作目录显示
106.
107. #endif OPERATION_H

```

2. main.cpp

```

1. #define _CRT_SECURE_NO_WARNINGS
2. #include "Operation.h"
3. #include <stdio.h>
4. #include <stdlib.h>
5. #include <windows.h>
6.
7. void show1() {
8.     printf("*****\n");
9.     printf("*****欢迎来到药品销售管理系统*****\n");
10.    printf("* 1.管理员注册                *\n");
11.    printf("* 2.管理员登录                *\n");
12.    printf("* 3.管理员注销                *\n");
13.    printf("* 4.管理员密码更改            *\n");
14.    printf("* 5.退出                    *\n");
15.    printf("*****\n");
16.
17. }    //登录前页面展示
18.
19. static void Clear() {

```

```
20. Sleep(3000);
21. system("cls");
22. }
23.
24.
25. int main() {
26.
27.     int a = 0, b = 0;
28.
29.     adm paarr[1];
30.     med pmarr[1];
31.
32.
33.     //初始化 Administrator 数组
34.     AInit(paarr);
35.
36.     //初始化药品数组
37.     MInit(pmarr);
38.
39.     int num, passwd, newpasswd;
40.     char name;
41.
42.     while (1) {
43.         show1();    // 登录前页面展示
44.         printf("请输入你想进行的操作 (1,2,3,4,5): ");
45.         int n = 0;
46.         scanf("%d", &n);
47.
48.         switch (n) {
49.             case 1:
50.                 printf("请输入编号: ");
51.                 scanf("%d", &num);
52.                 printf("请输入姓名: ");
53.                 scanf("%s", &name);
54.                 printf("请输入密码: ");
```

```

55.     scanf("%d", &passwd);
56.     enroll(paarr,num,&name,passwd );
57.     Clear();
58.     break;
59. case 2:
60.     printf("请输入编号: ");
61.     scanf("%d", &num);
62.     printf("请输入密码: ");
63.     scanf("%d", &passwd);
64.     login(paarr,num, passwd,pmarr);
65.     Clear();
66.     break;
67. case 3:
68.     printf("请输入编号: ");
69.     scanf("%d", &num);
70.     printf("请输入姓名: ");
71.     scanf("%s", &name);
72.     printf("请输入密码: ");
73.     scanf("%d", &passwd);
74.     logout(paarr,num,&name, passwd);
75.     Clear();
76.     break;
77. case 4:
78.     printf("请输入编号: ");
79.     scanf("%d", &num);
80.     printf("请输入姓名: ");
81.     scanf("%s", &name);
82.     printf("请输入密码: ");
83.     scanf("%d", &passwd);
84.     printf("请输入新密码: ");
85.     scanf("%d", &newpasswd);
86.     change_password(paarr,num,&name, passwd,newpasswd);
87.     Clear();
88.     break;
89. case 5:
90.     return 0;
91. default:
92.     printf("请重新输入你想进行的操作! \n");
93.     Clear();
94.     //show1();    //登录前页面展示
95. }
96. }
97.

```

```

98.
99. return 0;
100. }
101.
102. void show2() {
103.     printf("*****\n");
104.     printf("*****欢迎来到药品销售管理系统*****\n");
105.     printf("* 1.添加药品                *\n");
106.     printf("* 2.按名称查药品                *\n");
107.     printf("* 3.按年份查药品                *\n");
108.     printf("* 4.修改药品信息                *\n");
109.     printf("* 5.卖药品                    *\n");
110.     printf("* 6.统计价格大于 30 的药品        *\n");
111.     printf("* 7.显示所有药                *\n");
112.     printf("* 8.退出系统                    *\n");
113.     printf("*****\n");
114. } // 药品操作目录显示
115.
116. void OPIN(med *pmarr) {
117.
118.     while (1) {
119.         show2(); // 药品操作目录显示
120.         printf("请输入你想进行的操作 (1,2,3,4,5,6,7) : ");
121.         int n = 0;
122.         scanf("%d", &n);
123.         switch (n) {
124.             case 1:
125.                 add_med(pmarr); // 添加药品
126.                 Clear();
127.                 break;
128.             case 2:
129.                 search_med_byname( pmarr); // 通过名称
130.                 Clear();
131.                 break;
132.             case 3:
133.                 search_med_bytime(pmarr); // 通过时间
134.                 Clear();
135.                 break;
136.             case 4:
137.                 revise_update_med(pmarr); // 修改药品信息

```

```

138.   Clear();
139.   break;
140.   case 5:
141.       sell_update_med(pmarr); //卖药品
142.       Clear();
143.       break;
144.   case 6:
145.       statistics_med(pmarr);
146.       Clear();
147.       break;
148.   case 7:
149.       show_med(pmarr);
150.       Clear();
151.       break;
152.   case 8:
153.       return ;
154.       Clear();
155.   default:
156.       printf("请重新输入你想进行的操作! \n");
157.       Clear();
158.       //show2();    //登录前页面展示
159.   }
160. }
161. }

```

3. Administrator operation.cpp

```

1.  #define _CRT_SECURE_NO_WARNINGS
2.  #include "Operation.h"
3.  #include <cassert>
4.  #include <stdio.h>
5.  #include <string.h>
6.
7.  //初始化 Administrator
8.  void AInit(adm* paarr) {
9.      assert(paarr != NULL);
10.     paarr->a_size = 0;
11.     File_read_adm( paarr, "administrator.txt");
12. }
13.
14. //注册

```

```

15. void enroll(adm* paarr, int num, const char *name, int passwd) {
16.     if (paarr->a_size == ADMSIZE) {
17.         printf("管理员岗位已满, 注册失败! \n");
18.         return;
19.     }
20.     else {
21.         paarr->a_arr[paarr->a_size ].a_num = num;
22.         strcpy(paarr->a_arr[paarr->a_size ].a_name , name);
23.         paarr->a_arr[paarr->a_size ].a_passwd = passwd;
24.         paarr->a_size++;
25.         printf("注册成功\n");
26.     }
27.     File_write_adm(paarr, "administrator.txt");
28.
29. }
30.
31. // 登录
32. void login(adm* paarr, int num, int passwd, med *pmarr) {
33.     int k = 0;
34.     for (int i = 0; i < paarr->a_size; i++) {
35.         if ((paarr->a_arr[i].a_num == num)
36.             && (paarr->a_arr[i].a_passwd == passwd)) {
37.             printf("登陆成功\n");
38.             printf("进入药品销售管理系统! \n");
39.             OPIN(pmarr);
40.             k = 1;
41.
42.         }
43.     } if (k == 0) printf("登录失败\n");
44.
45. }
46.
47. // 注销
48. void logout(adm* paarr, int num, const char* name, int passwd) {
49.     int k = 0;
50.     for (int i = 0; i < paarr->a_size; i++) {
51.         administrator* pca =& paarr->a_arr[i];
52.         if ((paarr->a_arr[i].a_num == num) && !strcmp(paarr->a_arr[i].a
            _name, name)
53.             && (paarr->a_arr[i].a_passwd == passwd)) {
54.             int n = paarr->a_size - i - 1;

```



```

55.
56.     memmove(pca, pca + 1, n * sizeof(administrator));
57.     //memset(&(paarr->a_arr[paarr->a_size]), NULL, sizeof(administ
    rator));
58.
59.     printf("注销成功\n");
60.     paarr->a_size--;
61.     k = 1;
62.     //return;
63.
64. }
65. }
66. if (k == 0) {
67.     printf("注销失败，当前用户不存在或密码不正确\n");
68. }
69. File_write_adm(paarr, "administrator.txt");
70. }
71.
72. //更改密码
73. void change_password(adm* paarr, int num, const char* name, int p
    asswd, int newpasswd) {
74.     int k = 0;
75.     for (int i = 0; i < paarr->a_size; i++) {
76.         //administrator* pcm = &paarr->a_arr[i];
77.         if ((paarr->a_arr[i].a_num == num) && !strcmp(paarr->a_arr[i].a
            _name, name)
78.             && (paarr->a_arr->a_passwd == passwd)) {
79.             paarr->a_arr[i].a_passwd = newpasswd;
80.             printf("更改成功\n");
81.             k = 1;
82.             //return ;
83.
84.         }
85.     }
86.     if (k == 0) {
87.         printf("更改失败，当前用户不存在或密码不正确\n");
88.     }
89.     File_write_adm(paarr, "administrator.txt");
90. }
91.
92.

```

4. Medince operation.cpp

```
1. #define _CRT_SECURE_NO_WARNINGS
2. #include "Operation.h"
3. #include <cassert>
4. #include <malloc.h>
5. #include <stdio.h>
6. #include <string.h>
7. //Medince 药品操作
8.
9. //初始化药品数组
10. void MInit(med* pmarr) {
11.     pmarr->pmeds = (medince*)malloc(MEDINCE * sizeof(medince));
12.     assert(pmarr->pmeds != NULL);
13.     if (pmarr->pmeds == NULL) {
14.         return;
15.     }
16.     pmarr->medcapacity = MEDINCE;
17.     pmarr->m_size = 0; //有效卡数量0
18.     //printf("初始化数组成功\n");
19.     File_read_med(pmarr, "medince.txt");
20. }
21.
22. //动态内存扩容
23. void Grow(med* pmarr) {
24.     medince* p = (medince*)realloc(pmarr->pmeds, 2 * pmarr->medcapacity * sizeof(medince));
25.     assert(p != NULL);
26.     if (p == NULL) {
27.         free(pmarr->pmeds);
28.         pmarr->pmeds = NULL;
29.         return;
30.     }
31.     pmarr->pmeds = p;
32.     pmarr->medcapacity *= 2;
33. }
34.
35. //通过名称查询药品
36. static medince* searchmedbyname(med* pmarr, const char name[]) {
```

```

37. for (int i = 0; i < pmarr->m_size; i++) {
38.     if (strcmp(pmarr->pmeds[i].m_name, name)==0)
39.         return &pmarr->pmeds[i];
40. }
41. return NULL;
42. }
43.
44. //添加药品
45. void add_med(med* pmarr) {
46.     if (pmarr->m_size == pmarr->medcapacity) Grow(pmarr);
47.     char name[20];
48.     char manufacture[20];
49.     //如果将要输入的药品，名称和生产日期之前存在并相同，直接在库存量增加
50.     printf("请输入药品名称: ");
51.     scanf("%s", &name);
52.     printf("请输入药品生产日期: ");
53.     scanf("%s", &manufacture);
54.     if ((searchmedbyname(pmarr, name) != NULL) &&
55.         strcmp(searchmedbyname(pmarr, name)->m_manufacture , manufacture) ==0){
56.         searchmedbyname(pmarr, name)->m_inventory++;
57.         printf("当前药品已存在，且生产日期相同，已在库存量增加。\\n");
58.     }
59.
60.     else {
61.         strcpy(pmarr->pmeds[pmarr->m_size].m_name, name);
62.         strcpy(pmarr->pmeds[pmarr->m_size].m_manufacture, manufacture);
63.
64.         float price;
65.         printf("请输入药品单价: ");
66.         scanf("%f", &price);
67.         pmarr->pmeds[pmarr->m_size].m_price = price;
68.
69.         int inventory;
70.         printf("请输入药品库存量: ");
71.         scanf("%d", &inventory);
72.         pmarr->pmeds[pmarr->m_size].m_inventory = inventory;
73.
74.         char brand[20];

```

```

75. printf("请输入药品品牌: ");
76. /*fgets(brand, sizeof(brand), stdin);
77. size_t len = strlen(brand);
78. if (len > 0 && brand[len - 1] == '\n') brand[len - 1] = '\0';*/
79. scanf("%s", &brand);
80. strcpy(pmarr->pmeds[pmarr->m_size].m_brand, brand);
81.
82. char introduction[50];
83. printf("请输入药品简介: ");
84. scanf("%s", &introduction);
85. //fgets(introduction, sizeof(introduction), stdin);
86. //size_t len_2 = strlen(introduction);
87. //if (len_2 > 0 && introduction[len_2 - 1] == '\n') introduction
    n[len_2 - 1] = '\0';
88. strcpy(pmarr->pmeds[pmarr->m_size].m_introduction, introduction)
    ;
89.
90. pmarr->m_size++;
91. printf("药品添加成功, 请进下下一步操作\n");
92. }
93. File_write_med(pmarr, "medince.txt");
94.
95. }
96.
97. // 查询药品
98. // 通过名称
99. void search_med_byname (med* pmarr) {
100. char name[20];
101. printf("请输入药品名称: ");
102. scanf("%s", &name);
103. if (searchmedbyname(pmarr, name) != NULL) {
104. //打印查询到的药品信息
105. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价",
        "药品品牌", "药品生产日期", "药品库存量", "药品简介");
106. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", searchmedbyname(pmarr, name)->m_name,
        searchmedbyname(pmarr, name)->m_price, searchmedbyname(pmarr, name)->m_brand, searchmedbyname(pmarr, name)->m_manufacture,
        searchmedbyname(pmarr, name)->m_inventory, searchmedbyname(pmarr, name)->m_introduction);

```

```

109. }
110. else
111.     printf("当前药品不存在，请重新进行相关操作！");
112. }
113.
114. //通过时间
115. void search_med_bytime(med* pmarr) {
116.     char manufacture[20];
117.     printf("请输入药品生产日期：");
118.     scanf("%s", &manufacture);
119.     int time = (manufacture[0] - '0') * 1000 + (manufacture[1] - '0')
        * 100 + (manufacture[2] - '0') * 10 + (manufacture[3] - '0');
120.     printf("将会查找%d 年之前生产的药品\n", time);
121.     printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价",
        "药品品牌", "药品生产日期", "药品库存量", "药品简介");
122.     int k = 0;
123.     for (int i = 0; i < pmarr->m_size; i++) {
124.         int time2 = (pmarr->pmeds[i].m_manufacture[0] - '0') * 1000 +
            (pmarr->pmeds[i].m_manufacture[1] - '0') * 100
125.             + (pmarr->pmeds[i].m_manufacture[2] - '0') * 10 + (pmarr->pmeds[i].m_manufacture[3] - '0');
126.         if (time2 < time) {
127.             k = 1;
128.
129.             printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m_name, pmarr->pmeds[i].m_price,
130.                 pmarr->pmeds[i].m_brand, pmarr->pmeds[i].m_manufacture, pmarr->pmeds[i].m_inventory, pmarr->pmeds[i].m_introduction);
131.         }
132.     }
133.     if (k == 0) {
134.         printf("抱歉未查询到相关药品\n");
135.     }
136. }
137.
138.
139. //更新药品
140. // 修改药品信息
141. void revise_update_med(med* pmarr) {
142.     char name[20];
143.     printf("请输入药品名称：");

```

```

144. scanf("%s", &name);
145. if (searchmedbyname(pmarr, name) != NULL) {
146.     printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价",
        "药品品牌", "药品生产日期", "药品库存量", "药品简介");
147.     printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", searchmedbyname(pmarr, name)->m_name,
        searchmedbyname(pmarr, name)->m_price, searchmedbyname(pmarr, name)->m_brand, searchmedbyname(pmarr, name)->m_manufacture,
        searchmedbyname(pmarr, name)->m_inventory, searchmedbyname(pmarr, name)->m_introduction);
150.     /*printf("请输入药品名称: ");
151.     scanf("%s", &searchmedbyname(pmarr, name)->m_name);
152.     printf("请输入药品生产日期: ");
153.     scanf("%s", &searchmedbyname(pmarr, name)->m_manufacture);
154.     printf("请输入药品单价: ");
155.     scanf("%f", &searchmedbyname(pmarr, name)->m_price);
156.     printf("请输入药品库存量: ");
157.     scanf("%d", &searchmedbyname(pmarr, name)->m_inventory);
158.     printf("请输入药品品牌: ");
159.     scanf("%s", &searchmedbyname(pmarr, name)->m_brand);
160.     printf("请输入药品简介: ");
161.     scanf("%s", &searchmedbyname(pmarr, name)->m_introduction);*/
162.
163.     char name[20];
164.     char manufacture[20];
165.     printf("请输入药品名称: ");
166.     scanf("%s", &name);
167.     printf("请输入药品生产日期: ");
168.     scanf("%s", &manufacture);
169.
170.     strcpy(searchmedbyname(pmarr, name)->m_name, name);
171.     strcpy(searchmedbyname(pmarr, name)->m_manufacture, manufacture);
172.
173.     float price;
174.     printf("请输入药品单价: ");
175.     scanf("%f", &price);
176.     searchmedbyname(pmarr, name)->m_price = price;
177.
178.     int inventory;

```

```

179. printf("请输入药品库存量: ");
180. scanf("%d", &inventory);
181. searchmedbyname(pmarr, name)->m_inventory = inventory;
182.
183. char brand[20];
184. printf("请输入药品品牌: ");
185. //fgets(brand, sizeof(brand), stdin);
186. //size_t len = strlen(brand);
187. //if (len > 0 && brand[len - 1] == '\n') brand[len - 1] = '\0'
    ;
188. scanf("%s", &brand);
189. strcpy(searchmedbyname(pmarr, name)->m_brand, brand);
190.
191. char introduction[50];
192. printf("请输入药品简介: ");
193. scanf("%s", &introduction);
194. /*fgets(introduction, sizeof(introduction), stdin);
195. size_t len_2 = strlen(introduction);
196. if (len_2 > 0 && introduction[len_2 - 1] == '\n') introduction
    [len_2 - 1] = '\0';*/
197. strcpy(searchmedbyname(pmarr, name)->m_introduction, introduct
    ion);
198.
199.
200. }
201. else
202.     printf("当前药品不存在, 请重新进行相关操作!");
203.
204.
205. File_write_med(pmarr, "medince.txt");
206. }
207.
208. //卖药品
209. void sell_update_med(med* pmarr) {
210.     char name[20];
211.     printf("请输入药品名称: ");
212.     scanf("%s", &name);

```

```

213. int k = 0;
214. for (int i = 0; i < pmarr->m_size; i++) {
215.     medince* pcm = &pmarr->pmeds[i];
216.     if (strcmp(pmarr->pmeds[i].m_name, name) == 0) {
217.         if (pmarr->pmeds[i].m_inventory == 1) {
218.             int n = pmarr->m_size - i + 1;
219.             memmove(pcm, pcm + 1, n * sizeof(medince));
220.             pmarr->m_size--;
221.         }
222.         else pmarr->pmeds[i].m_inventory--;
223.         k = 1;
224.     }
225. }
226. if(k==0)
227.     printf("当前药品不存在，请重新进行相关操作！");
228. File_write_med(pmarr, "medince.txt");
229.
230. }
231.
232. // 统计价格（例如统计大于30元药品）
233. void statistics_med(med* pmarr) {
234.     float price = 0;
235.     printf("请输入价格：");
236.     scanf("%f", &price);
237.     printf("将会输出%f价格之上的药品\n");
238.     printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价",
239.         "药品品牌", "药品生产日期", "药品库存量", "药品简介");
239.     int k = 0;
240.     for (int i = 0; i < pmarr->m_size; i++) {
241.
242.         if (pmarr->pmeds[i].m_price > price) {
243.             printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m_
244.                 name, pmarr->pmeds[i].m_price,
245.                 pmarr->pmeds[i].m_brand, pmarr->pmeds[i].m_manufacture, pmarr->pmeds[i].m_inventory, pmarr->pmeds[i].m_introduction);
246.             k = 1;
247.         }
248.     }
249.     if (k == 0) {
250.         printf("未找到价格大于30的药品\n");
251.     }

```



```
252.
253. }
254.
255. //显示药品信息
256. void show_med(med* pmarr) {
257.     printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价",
        "药品品牌", "药品生产日期", "药品库存量", "药品简介");
258.     for (int i = 0; i < pmarr->m_size; i++) {
259.         printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m_n
            ame, pmarr->pmeds[i].m_price,
260.             pmarr->pmeds[i].m_brand, pmarr->pmeds[i].m_manufacture, pmarr
                ->pmeds[i].m_inventory, pmarr->pmeds[i].m_introduction);
261.     }
262. }
```

5. File operation.cpp

```
1. #define _CRT_SECURE_NO_WARNINGS
2. #include "Operation.h"
3. #include <cassert>
4. #include <stdio.h>
5.
6.
7. // File 文件操作
8.
9. //药品文件读入
10. void File_read_med(med* pmarr, const char* filepath) {
11.     assert(pmarr && filepath);
12.     FILE* fp = fopen(filepath, "r");
13.     if (fp == NULL) {
14.         fp = fopen(filepath, "w");
15.         if (fp == NULL) {
16.             printf("文件以写模式打开失败\n");
17.             fclose(fp);
18.             return;
19.         }
20.     }
21.     int i = 0;
```

```

22. medince* pc;
23. while (1) {
24.     pc = &pmarr->pmeds[i];
25.     if (fscanf(fp, "%f,%d,%[^,],%[^,],%[^,],%[^\\n]\\n", &pc->m_price,
        &pc->m_inventory,
26.         &pc->m_name, &pc->m_brand, &pc->m_manufacture, &pc->m_introduc
            tion) == 6) {
27.         i++;
28.         pmarr->m_size++;
29.     }
30.     else break;
31.     if (pmarr->m_size == pmarr->medcapacity) Grow(pmarr);
32. }
33.
34. fclose(fp);
35. }
36.
37. //药品文件写入
38. void File_write_med(med* pmarr, const char* filepath) {
39.     assert(pmarr && filepath);
40.     FILE* fp = fopen(filepath, "w");
41.     if (fp == NULL) {
42.         printf("文件以写方式打开失败\\n");
43.         return;
44.     }
45.     for (int i = 0; i < pmarr->m_size; i++) {
46.
47.         medince *pc= &pmarr->pmeds[i];
48.         fprintf(fp, "%f,%d,%s,%s,%s,%s\\n", pc->m_price, pc->m_inventory
            ,
49.             pc->m_name, pc->m_brand, pc->m_manufacture, pc->m_introduction)
            ;
50.
51.
52.     }
53.     fclose(fp);
54. }
55.
56. //管理员文件读入
57. void File_read_adm(adm* paarr, const char* filepath) {
58.     assert(paarr && filepath);

```

```
59. FILE* fp = fopen(filepath, "r");
60. if (fp == NULL) {
61.     fp = fopen(filepath, "w");
62.     if (fp == NULL) {
63.         printf("文件以写模式打开失败\n");
64.         fclose(fp);
65.         return;
66.     }
67. }
68. int i = 0;
69. administrator* pc;

70. while (1) {    //读取到文件末尾    feof(fp)!= EOF
71.     pc = &paarr->a_arr[i];
72.     if (fscanf(fp, "%d %s %d\n", &pc->a_num, &pc->a_name, &pc->a_passwd) == 3) {
73.         i++;
74.         paarr->a_size++;
75.     }
76.     else break;
77. }
78. fclose(fp);
79. }
80.

81. //管理员文件写入
82. void File_write_adm(adm* paarr, const char* filepath){
83.     assert(paarr && filepath);
84.     FILE* fp = fopen(filepath, "w");
85.     if (fp == NULL) {
86.         printf("文件以写方式打开失败\n");
87.         return;
88.     }
89.     for (int i = 0; i < paarr->a_size; i++) {
90.         //fwrite(&paarr->a_arr[i], sizeof(administrator), 1, fp);
91.         fprintf(fp, "%d %s %d\n", paarr->a_arr[i].a_num, paarr->a_arr[i].a_name, paarr->a_arr[i].a_passwd);
92.     }
93. }
94. fclose(fp);
95. }
```