廣東工業大學

程序设计课程设计报告

题　　目　冰箱管理系统设计

学　　院　先进制造学院

专　　业　计算机科学与技术

年级班别　22 级计科 8 班

学　　号　3122008883

学生姓名　陈煜祺

指导教师　李泓澍

成　　绩

程序功能完成情况	
测试用例全面情况	
报告格式是否与要求相符	
报告内容是否准确全面	

2022 年 12 月

一、课程设计题目、内容与要求

选题题目：

冰箱管理系统设计

选题内容与要求：

设有一个容积为 50 的冰箱，可以存放多个食物。

冰箱温度必须设置为其中所有食物中最低的保存温度。

其中冰箱信息包括：容积、温度（-20℃～10℃）。

食物信息包括：食物名称、体积、保存温度、食物种类（蔬菜、肉类、水果）。

必选实现功能：

1. 使用链表保存冰箱中的食物信息。
2. 显示冰箱信息，包含冰箱剩余容积、温度、冰箱中的食物信息列表。
3. 能往冰箱中存入食物。如果一个食物保存温度低于冰箱温度下限，即拒绝放入冰箱中。如果一个食物体积大于冰箱剩余容积，即拒绝放入冰箱中（只考虑剩余容积和食物体积的关系，忽略物体形状）。
4. 能从冰箱中取出食物，即删除食物信息。
5. 能够查询冰箱中某一食物种类的食物信息。
6. 能修改某一食物的信息，注意如果体积或者保存温度不合适则应拒绝修改并输出报错信息。
7. 使用文件方式存储、读取数据。

可选实现功能：

1. 能根据食物体积进行排序。
2. 使用图形界面。
3. 程序鲁棒性，如输入错误判断。
4. 其他特色功能。

二、总体设计

程序包含七大功能模块组，分别为：

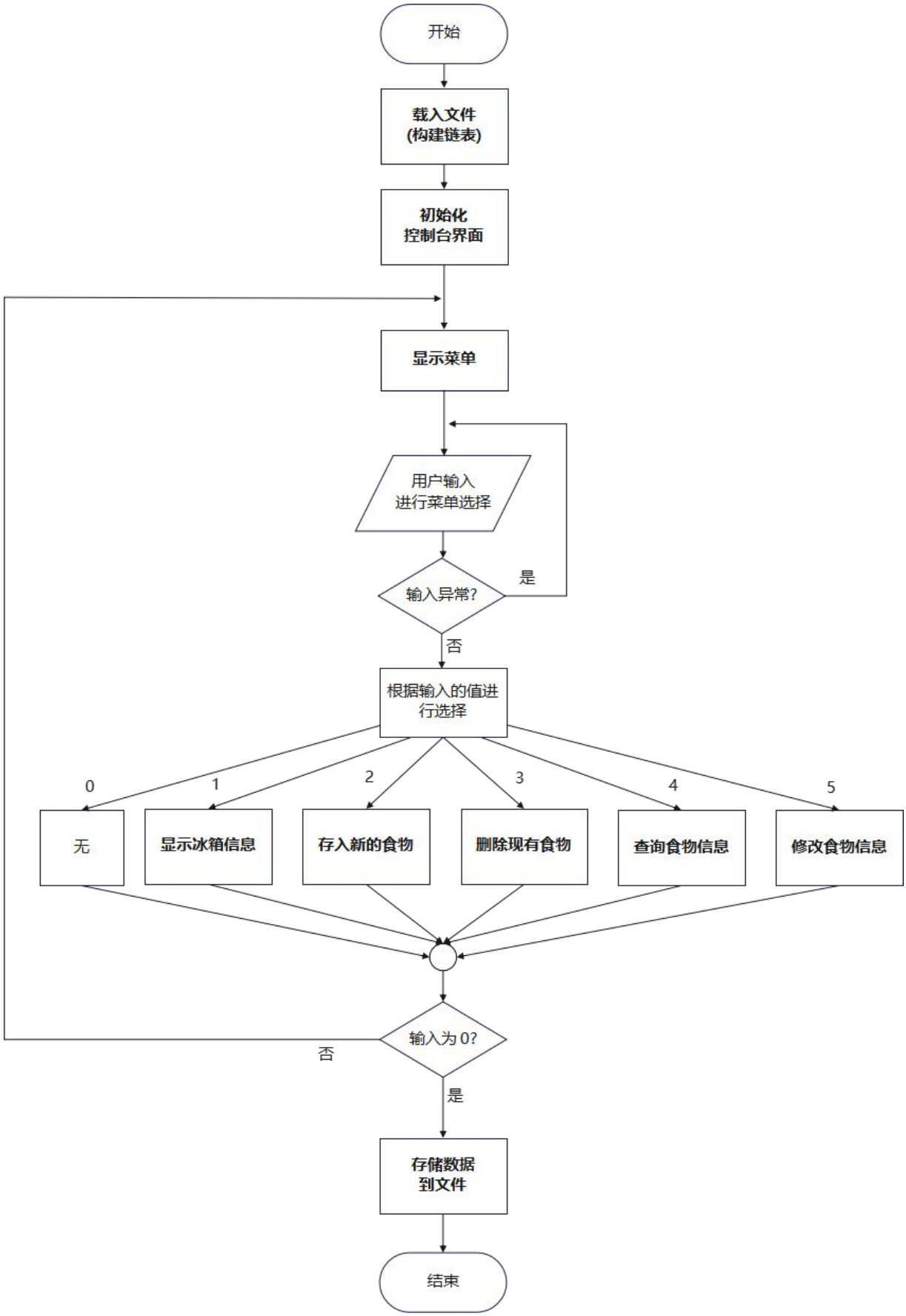
1．显示冰箱信息
2．存入新的食物
3．删除现有食物
4．查询食物信息
5．修改食物信息
6．文件处理：文件数据加载/存储
7．界面设置：初始化界面/展示菜单

其中，在各模块中还内置实现了如下功能点：

1．模块 2 中支持检测同种食物且可选合并操作；
2．模块 2～5 中对链表进行增删查改操作时，会即时对其按体积升序进行排序；
3．模块 6 中使用了双向动态链表来加载文件数据；
4．模块 7 中为用户提供了友好的可视化界面与详尽的操作指引；
5．各模块功能均支持中文显示与输入；
6．程序整体有较强的鲁棒性，对各类别的异常输入均能识别并反馈。

总体来说，用户在打开程序时，会首先进行文件数据的读取与加载（生成动态链表），同时控制台窗口会自动进行合适的大小、颜色等界面初始化，并显示菜单，提示与等待用户输入。根据用户输入的不同，程序会跳转到功能模块 1～5 中的一个或跳转到结束退出阶段：跳转到功能模块时，用户根据指引操作即可实现所选功能，功能模块执行完毕后，将跳回主菜单界面，重新提示与等待用户输入，如此往复循环直至用户选择退出；跳转到结束退出阶段时，程序会将此时链表中的数据写入文件中进行存储，存储完毕后将自动退出。

功能模块结构图如下页附图所示，其中粗体表示该部分功能已封装为函数模块。


开始
载入文件
(构建链表)
初始化
控制台界面
显示菜单
用户输入
进行菜单选择
输入异常?
是
否
根据输入的值进
行选择
0
1
2
3
4
5
无
显示冰箱信息
存入新的食物
删除现有食物
查询食物信息
修改食物信息
输入为 0?
否
是
存储数据
到文件
结束

三、详细设计

下面介绍各功能模块的实现思路。

模块 1：显示冰箱信息。

本模块的实现思路是先输出用全局变量存储的冰箱信息（当前温度与剩余容积），再输出链表中存储的食物信息。

输出食物信息时，设置一个临时的指针变量 now 指向头结点，用循环结构从头结点开始依次顺序遍历链表的所有结点，对遍历到的每个结点，输出其中的成员变量值（依次为食物名称、体积、保存温度和种类）。

输出完毕后，提示用户查看和按任意键返回即可。输出时利用了各类制表字符和 printf 函数的格式附加字符来进行排版，维持美观整齐的界面。

模块 2：存入新的食物。

为了提高程序的容错性，本模块的实现思路是先判断后入链。

判断部分将依次录入新食物的名称、体积、温度和种类，对于用户的意外输入，分以下两种情况进行处理。

第一类意外输入为无意义/不合法输入，当输入食物名称时遇到过长的字符串（本程序中设定为不超过 20 个汉字）、当输入体积和温度时遇到了非法字符（如温度为“???”）或无现实意义的输入（如体积为 0 或负数）、当输入种类时遇到非限定的字符（仅限 0/1/2，其中 0 代表蔬菜，1 代表肉类，2 代表水果），都会被认定为是无意义/不合法输入的情况，程序将输出提示用户“输入有误”，并要求用户重新输入直至不再遇到此类情况。

第二类意外输入为不符合冰箱信息条件的输入，此类输入有意义且合法，但因为输入的信息不符合冰箱信息条件，逻辑上不应允许放入冰箱。如输入的温度低于冰箱的最低温度，或输入的体积大于冰箱的当前剩余容积。对于此类输入，可以认为用户试图放入一个不允许放入的食物，应弹窗汇报错误信息，并中止此次录入，提前结束模块 2。

逐条判断并录入成功后，进行结点的入链操作。

此时经判断允许入链的信息已存储到一个由 malloc 函数申请得到的新结构体结点中，称之为“待入链结点”。

为了实现“检测同种食物并可选合并”的功能，程序将从头结点开始顺序遍历链表，当遇到同名称同温度同种类的食物时，认为这两种食物是同种食物，弹窗提示用户是否进行合并。若进行合并，则对原结点的成员变量“体积”进行增加，将原结点从链上移除，作为新的待入链结点参与后续入链过程；若不进行合并，则无任何变化，检测同种食物并可选合并的功能结束。

为了维持链表的有序性和实现冰箱温度的实时更新，程序将再一次遍历链表，当遇到成员变量体积大于待入链结点的结点时，将待入链结点插入当前结点的前面，即可保证链表按

体积升序有序。若遍历到链表尾结点时，尾结点的体积仍小于待入链结点的体积，则说明待入链结点的体积大于链表中所有结点的体积，此时将待入链结点插入到链表尾端作为新的尾结点既可。在遍历链表的同时将记录链表中的最低食物温度，将最低食物温度与待入链结点食物温度进行比较后取较小者作为新的冰箱温度，更新全局变量。

模块 3：删除现有食物。

本模块的实现思路是以食物名称作为关键字进行检索并删除。

本模块初始会先打印当前食物列表，并要求用户输入需要删除的食物名称。然后从头结点开始依次顺序遍历链表，用 strcmp 函数将结点的成员变量“食物名称”与输入的食物名称进行比对，若比对成功则删除该结点，若比对失败则遍历下一节点，如此直到遍历完整个链表。若检索完整个链表后仍匹配不到需要删除的食物名称，则提示检索失败，不存在用户输入的食物。

在遍历链表的同时将记录链表未删除结点中最低的食物温度，将遍历得到的最低食物温度作为新的冰箱温度，更新全局变量。

检索前设置了两个临时变量以进行本次删除食物数量的计数和释放冰箱体积总量的统计，将在检索完毕后输出告知用户。

模块 4：查询食物信息。

本模块的实现思路是以食物种类作为关键字进行检索并输出。

本模块会要求用户输入需要查询的食物种类代号（仅限 0/1/2，其中 0 代表蔬菜，1 代表肉类，2 代表水果），然后进行检索。在输入时也进行了异常输入的检测与处理，这一部分与模块 2 的实现基本相同。检索部分的实现与模块 3 大致相同，从头结点开始依次顺序遍历链表，将结点的成员变量“类型”与输入的食物类型进行比对，若比对成功则输出该结点信息，如此直到遍历完整个链表。

同样地，检索前设置了一个临时变量以进行本次查询食物数量的计数，将在检索完毕后输出以告知用户本次查询类型的食物数量。

模块 5：修改食物信息。

本模块功能是综合各子功能来实现的，具体子功能的实现思路与前述模块中子功能的实现思路基本一致。

本模块程序会先打印当前食物列表，并在各食物前标记唯一的序号，要求用户输入需要修改的食物的序号。确定需要修改的食物后，会再次要求用户输入需要修改的信息序号（注：0：食物名称，1：体积，2：保存温度，3：种类）。

若是 0/3 则直接修改结点成员变量值即可，若是 1/2 则需在确定需要修改的信息后，将修改后的结点从链表上移除，并作为新的“待入链结点”重新入链，具体实现细节与模块 1

中一致，如此一来才能保证链表的有序性和冰箱温度的实时更新。在本模块的多次输入时也进行了两类意外输入的检测与处理，这一部分与模块 2 的实现基本相同。

模块 6：文件处理——文件数据加载/存储。

本模块由两个函数实现，分别进行数据的加载和存储。

先介绍程序启动时执行的文件数据加载函数。本函数的实现思路是利用 fscanf 函数读取 Data.dat 数据文件内的内容，并根据其动态地构建一个双向链表，直至读取到文件尾。动态构建双向链表的简要思路是设置两个全局变量 head 和 tail 分别作为头指针和尾指针，并设置一个临时变量 pnew 用于指向新的待入链结点。每当一个新的结点入链，将减少冰箱剩余容积，并及时更新冰箱当前温度。

接下来介绍程序关闭时执行的文件数据存储函数。

向文件存储数据时，不直接写入 Data.dat 数据文件，而是利用 fopen 函数新生成一个 Data.tmp 临时文件，然后遍历链表将各结点成员变量的值写入该临时文件。临时文件写入完毕后将其关闭，调用 remove 函数删除旧的 Data.dat 数据文件，然后再调用 rename 函数将临时文件 Data.tmp 重命名为 Data.dat。如此一来便实现了向数据文件中存储链表信息的目的。

模块 7：界面设置——初始化界面/展示菜单

本模块也由两个函数实现，分别进行控制台界面的初始化和主菜单的打印展示。

该模块实现较为简单，初始化界面函数主要通过调用 system 函数来完成，而展示菜单函数则是由一个 system 函数调用的 cls 清屏命令和一连串的 printf 输出函数来完成。主要难点在于批处理命令的使用和菜单界面的设计和排版。

四、公用数据结构设计及用法说明

下面介绍数据结构（双向动态链表）的设计、自定义结构体类型与全局变量的含义及用法等。

```
typedef struct food
{
    char name[50];
    float size;
    float temper;
    unsigned int type;
    struct food* next;
    struct food* prev;
}food;
food *head=NULL,*tail=NULL;
float vol=MAX_VOL,nowtemper=MAX_TEMPERLIMIT;
const char foodtype[3][10]=
{
    "蔬菜",
    "肉类",
    "水果",
};
```

如图是源文件程序中自定义的结构体类型和所有的全局变量。

下面介绍双向链表中结构体结点的内容及含义。结构体类型 food 由 6 个成员变量组成，前四个变量从上至下分别是食物的名称、体积、保存温度和食物类型序号，最后两个指针变量分别是指向后一个节点的后继指针和指向前一个节点的前驱指针。

下面介绍全局变量的含义。head、tail 分别为双向链表的头指针和尾指针。浮点型变量 vol 和 nowtemper 分别代表冰箱当前剩余容积和当前设定温度。

有必要对 foodtype 数组进行说明。常变量字符型二维数组 foodtype 存储了三个字符串常变量，代表的是食物类型的名称。当需要输出某结点所代表的食物的类型 type 时，只需调用 printf 函数以%s 格式符输出 foodtype[type]即可将无符号整型变量转换成为可供用户阅读的食物类型名称。如此保存食物类型的名称方便随时进行修改和后续“增添新的食物类型”的新功能开发。

五、参数与细节说明

下面介绍各宏定义参数、函数的含义及用法，同时说明一些细节。

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<windows.h>
#define MAX_VOL 50.0
#define MAX_TEMPERLIMIT 10.0
#define MIN_TEMPERLIMIT -20.0
#define MIN(a,b) (((a)<(b))?(a):(b))
#define MAX(a,b) (((a)>(b))?(a):(b))
#define NOT_NUM(x) (strspn((x), "-.0123456789")!=strlen((x)))
```

如图是源程序文件中包含的头文件和进行的宏定义声明。

下面说明一些在宏定义中体现的参数和函数。

MAX_VOL 的含义为冰箱的最大体积，根据选题要求设定为 50.0。MAX_TEMPERLIMIT 和 MIN_TEMPERLIMIT 的含义分别为冰箱的温度上限和下限，同样根据选题要求分别设定为 10.0℃和-20.0℃。可以根据需要随时进行修改。

类函数宏 MIN 和 MAX 为取最值函数，分别取最小值和最大值；类函数宏 NOT_NUM(x) 为检测 x 是否为只含有“-.0123456789”这些字符的字符串，若是则其值为 0，若否则其值为 1。

下面说明一些细节。

stdio.h、stdlib.h、string.h 为 C99 提供的标准库函数，windows.h 为微软发布的 windows 系统 API，被大多数 C 主流编译器所支持。本程序包含 windows.h 主要是为了调用其中的接口完成弹窗警告、无效化关闭按钮、逐行更改控制台输出颜色等功能，以提供更友好的用户体验和更美观的界面。在源程序文件中体现为 SetConsoleTextAttribute、MessageBox 等函数的调用。

关于中文显示与输入，部分情况下出现乱码的原因是 GBK 和 UTF-8 两种编码的不兼容。由于记事本的默认编码为 UTF-8，控制台的默认编码为 GBK，而 IDE 中编写源文件时的编码有可能为 GBK 也有可能为 UTF-8。要解决这个问题，只需要将三者统一即可。

六、试验结果

下面展示测试用例和运行结果截图。

Data.dat 数据文件原有内容：

苹果 3.200000 3.200000 2

速冻牛肉丸 4.000000 -15.000000 1

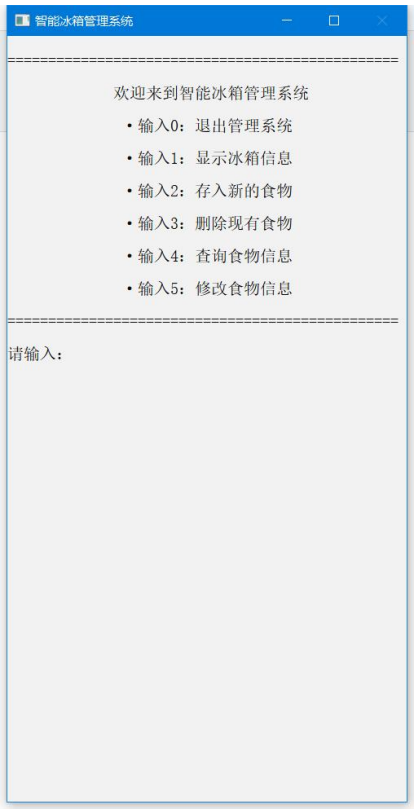
猪肉 5.500000 -2.300000 1

鸡翅 6.100000 -1.000000 1

娃娃菜 8.880000 2.500000 0

生菜 11.700000 3.500000 0

主菜单界面截图：



测试用例 1：

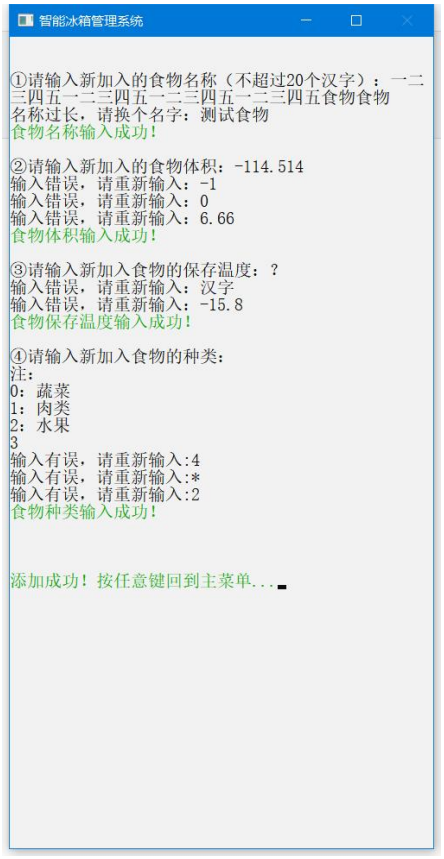
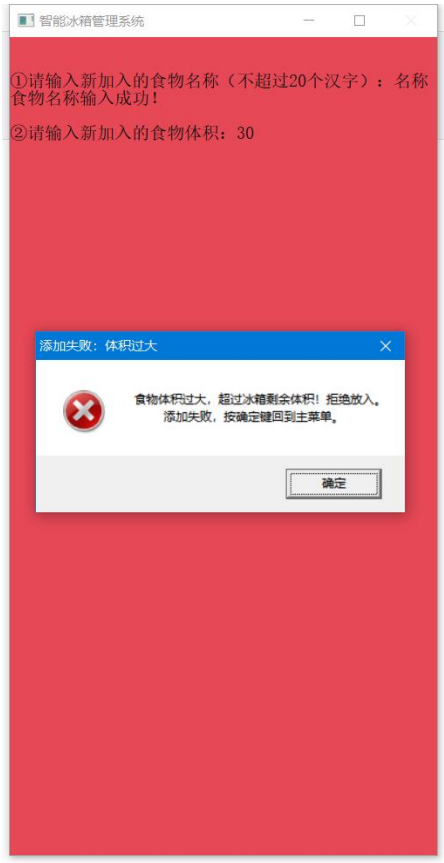
8

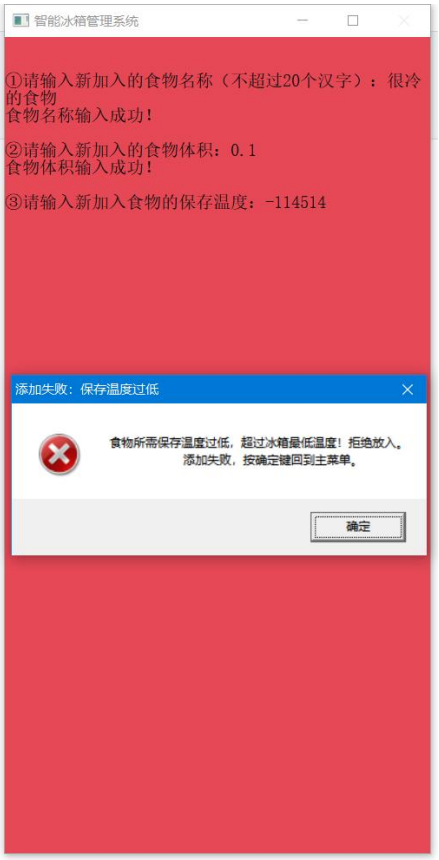
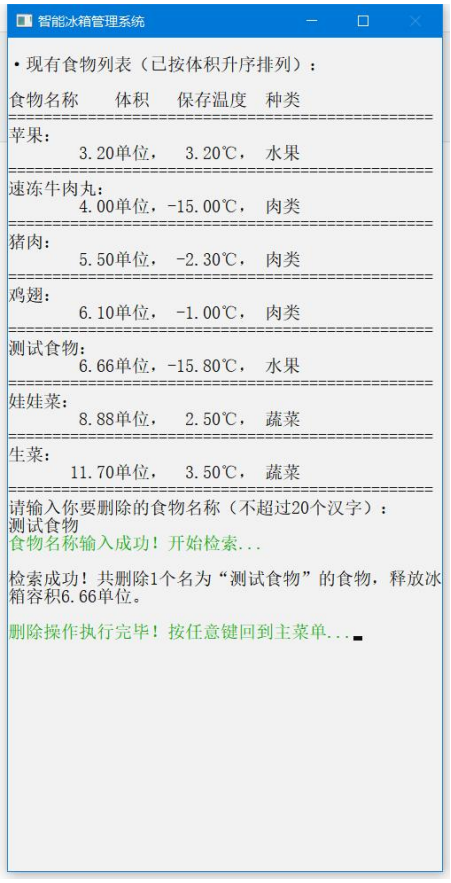
ok

汉字

1

当前运行结果：

测试用例 2： 2 一二三四五一二三四五一二三四五一二三四 五食物食物 测试食物 -114.514 -1 0 6.66 ？ 汉字 -15.8 3 4 * 2	**当前运行结果：** 
测试用例 3： 2 名称 30	**当前运行结果：** 

测试用例 4： 2 很冷的食物 0.1 -114514	**当前运行结果：** 
测试用例 5： 3 测试食物	**当前运行结果：** 

测试用例 6：

4

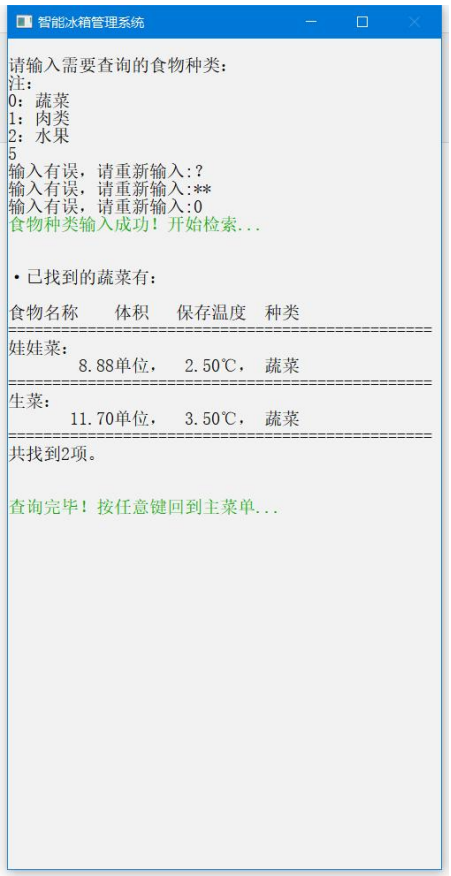
5

？

**

0

当前运行结果：



测试用例 7：

5

7

0

汉字

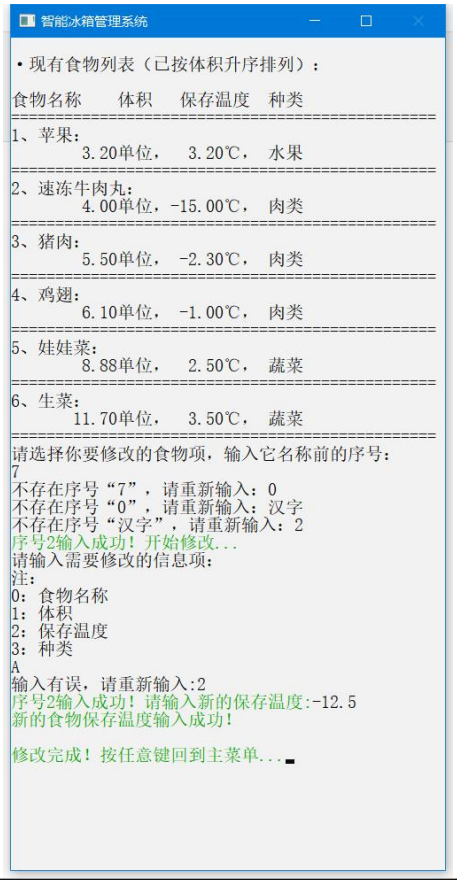
2

A

2

-12.5

当前运行结果：



测试用例 8.1：

2

苹果

3.2

3.200

2

测试用例 8.2：

（是）

1

Data.dat 数据文件现有内容：

速冻牛肉丸 4.000000 -12.500000 1

猪肉 5.500000 -2.300000 1

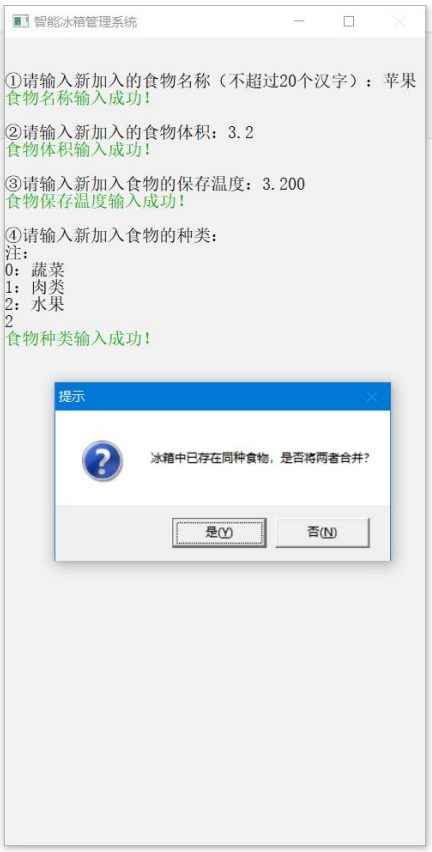
鸡翅 6.100000 -1.000000 1

苹果 6.400000 3.200000 2

娃娃菜 8.880000 2.500000 0

生菜 11.700001 3.500000 0

运行结果 8.1：



运行结果 8.2：

七、体会与总结

本次课程设计我选择了“冰箱管理系统”的选题，依照题目要求完成了显示冰箱信息、存入新的食物、删除现有食物、查询食物信息、修改食物信息、文件处理、动态链表构建七大基本功能，还附加了合并同种食物、排序、增强鲁棒性、友好交互界面等特色功能。

这次是我第一次尝试去着手完成一个稍大型的工程项目，也是第一次编写出超过 600 行代码的源文件。

量变产生质变，这次的课程设计与先前写代码的体验是很不一样的。我在这次课程设计中实践了许多新的知识，遇到了许多新的问题，接触了许多新的事物，也有了许多新的体会与收获。

这次课程设计是对这个学期以来学到的程序设计知识与技巧的一次综合考验。经过这次实践学习，我重温了函数、指针、结构体、链表等知识点，接触和尝试了文件处理、控制台界面设置、windowsAPI 调用等新技能，巩固和学习了许多知识。在实践的过程中，我也遇到了许多诸如显示乱码等令人头疼的问题，并最终经过自我摸索和查阅资料解决了问题，实现了程序的功能，收获了知识与经验。

这时回过头来重新审视源文件的代码，仍然会发现有许多可以改进的地方。实现功能上，或许可以再增添一个“增加用户自定义食物类型”的功能，让程序更完善，也更贴近现实；架构逻辑上，对于一些重复使用到的子功能（如异常输入判断、输出非默认颜色的信息、将一个待入链结点插入链表等），可以考虑封装成新的函数供重复调用，这样可以减少冗余的代码，使代码更清晰易读，也更具有模块化的特点；人机交互上，如果能将输入数字来选择对应功能改为移动鼠标点击按钮来选择对应功能，程序将会变得更加用户友好和直观易懂。

这些待改进的地方也恰给我指明了未来继续学习进步的方向：尝试编写带有可交互图形界面的程序，并进一步提升自己的工程代码能力。

这次课程设计的难点不在于算法的艰深，而在于功能模块的设计和程序整体的架构。应该要编写哪些函数？又应该如何将它们组织串联在一起？只有经过提前思考和布局，用自上而下的大局观和大视野来审视自己的代码，才有可能交出一份优美的答卷。如果拿到选题就立马着手埋头码字，往后势必要经过几次伤筋动骨的大改动，最后只能是身心俱疲、事倍功半。这是我从这次课程设计实践中总结出来的最大教训。而如何提升自己布局架构的工程能力，也就顺理成章地成为了我目前突破的关键点之一。

无他，唯手熟尔。

我想，无论是要提升工程代码能力还是学习图形界面编写，都离不开勤奋的训练和一次又一次的实践。只有在实践中不断遇到问题并解决问题，人的经验才能够丰富、技能才能够不断熟练与日益精湛。

未来的学习生活中，我打算尝试学习一些更适合进行图形界面编写的程序设计语言，并多尝试去做一些体量稍大的项目，用代码实现自己想要实现的功能，并最终封装形成实用的小程序。如果我的作品能为平凡的学习生活带来一些便捷或是欢乐，那就再好不过了。

八、参考文献

[1] 谭浩强. C 程序设计[M]. 第五版. 清华大学出版社, 2017.

[2] C 参考手册[EB/OL]. [2022-12-31]. zh.cppreference.com/w/c

[3] CSDN 社区[EB/OL]. [2022-12-31]. www.csdn.net

附录：程序清单

源程序文件：22 级计科 8 班陈煜祺 3122008883.c

可执行程序文件：22 级计科 8 班陈煜祺 3122008883.exe

数据文件：Data.dat

测试用例和编译运行结果文档：测试用例和编译运行结果.docx

程序设计课程设计报告文档：程序设计课程设计（3122008883 陈煜祺）.docx