

课程设计(论文)

题 目 名 称 用户软件信息管理系统

课 程 名 称 C语言程序课程设计

学 生 姓 名 杜嘉雯

学 号 2242001050

学院 、专业 国际学院、通信工程专业

指 导 教 师 成娅辉

2023年 5月 28 日

目 录

[1 前言 1](#_Toc453320180)

[2 需求分析 1](#_Toc453320181)

[2.1 课程设计目的 1](#_Toc453320182)

[2.2 课程设计任务 1](#_Toc453320183)

[2.3 设计环境 1](#_Toc453320184)

[2.4 开发语言 1](#_Toc453320185)

[3 分析和设计 1](#_Toc453320186)

[3.1 模块设计 1](#_Toc453320187)

[3.2 系统流程图 2](#_Toc453320188)

[3.3 主要模块的流程图 3](#_Toc453320189)

[4 具体代码实现 4](#_Toc453320190)

[5 课程设计总结 6](#_Toc453320191)

[5.1 程序运行结果 6](#_Toc453320192)

[5.2 课程设计体会 6](#_Toc453320193)

[参考文献 6](#_Toc453320194)

[致 谢 6](#_Toc453320195)

1 前言

编写一个程序来实现对用户所拥有的软件的“信息”进行查询，修改和管理；结构体来实现对用户和软件信息的记录，结构体数组来实现对大量用户的记录；首先将文件里的数据提取到内存中开辟好的结构体数组里以供访问，程序运行结束再将数据存回文件。

2 需求分析

2.1 课程设计目的

学生在教师指导下运用所学课程的知识来研究、解决一些具有一定综合性问题的专业课题。通过课程设计（论文），提高学生综合运用所学知识来解决实际问题、使用文献资料、及进行科学实验或技术设计的初步能力，为毕业设计（论文）打基础。

2.2 课程设计任务

用户软件信息管理系统（自选命题）

2.3 设计环境

（1）WINDOWS 11家庭版；

（2）VS2022。

2.4 开发语言

C语言

3 分析和设计

1. 准备工作：在程序开头定义了用户结构体和软件结构体，用户结构体中包含了用户id和以其所拥有的软件所组成的软件结构体数组，软件结构体中包含了软件名称，储存空间等信息；再然后是对一些函数模块的原型申明。
2. 主体：在主体main函数中，运用了printf函数将我们的软件界面借助命令提示符窗口展现出来；其后定义了一个用户结构体数组，用来存储用户信息；调用自定义函数load将文件里的数据从文件中提取到先前定义的用户结构体数组中；然后是使用switch语句，将用户输入的值与case值匹配，然后实行相对应的函数模块，借助while（1）循环，使得程序在不调用退出指令的情况下，在一个函数模块执行完的情况下可以接着接收值，进行下一次的函数调用。
3. 模块函数的完善：最后则是对实现各个功能的模块函数的内容进行编写，其中包括

void fu\_chu\_zhi(Person\_information); //标记

void Load(Person\_information); //加载至内存

void upload(Person\_information); //储存至硬盘

void Add\_person(Person\_information); //添加用户

void Modify\_person(Person\_information);//修改用户id

void Deleted\_person(Person\_information);//删除用户

void Add\_person\_app(Person\_information);//添加指定用户所有程序

void Modify\_person\_app(Person\_information);//修改用户软件信息

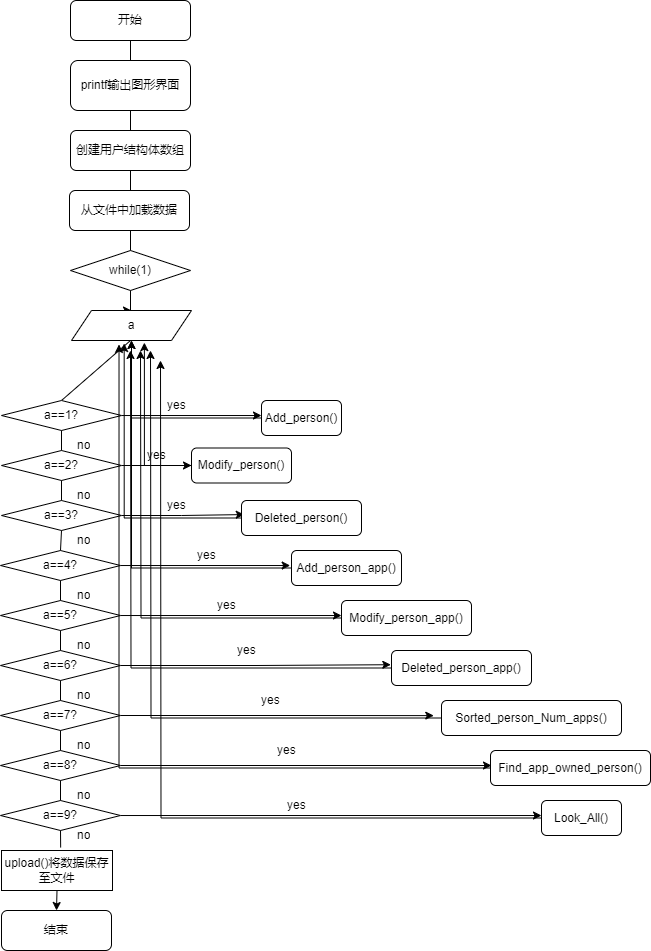
void Deleted\_person\_app(Person\_information);//删除用户软件

void Sorted\_person\_Num\_apps(Person\_information);//按照用户的所有软件数量进行降序排序

void Find\_app\_owned\_person(Person\_information);//寻找拥有指定软件用户的id

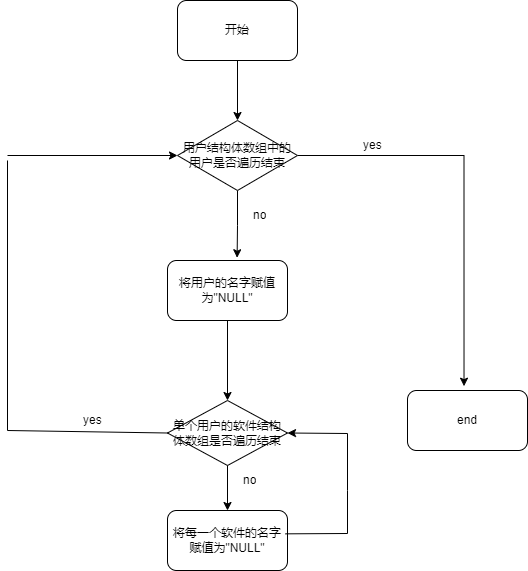
void Look\_All(Person\_information);//查看所有用户和拥有软件的信息

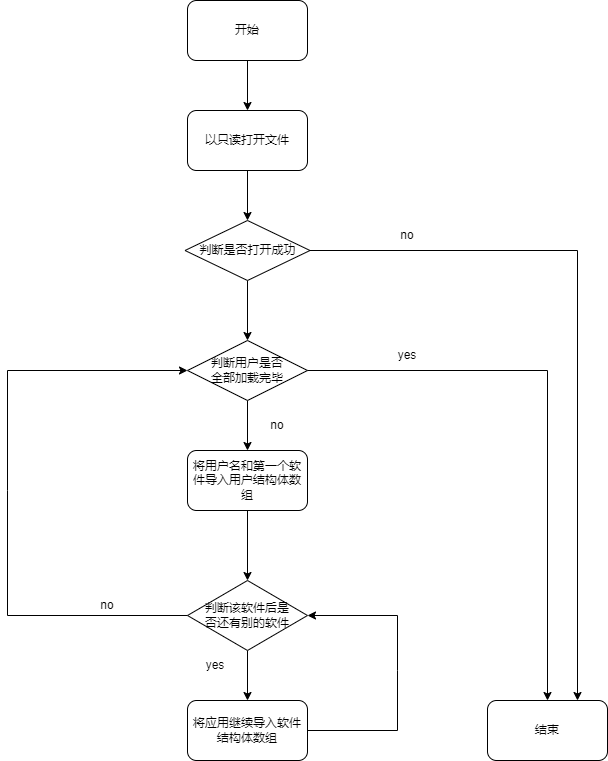
3.2 系统流程图



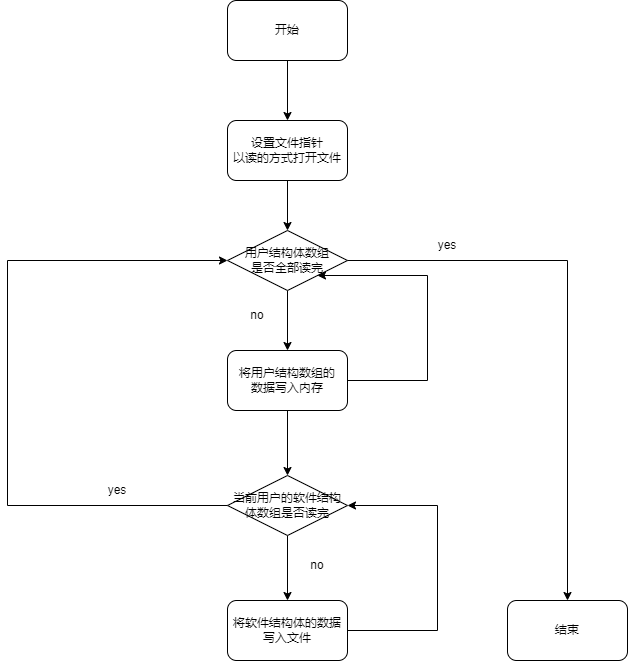
3.3 主要模块的流程图

（1）赋初值函数流程图（用于标记未被赋值的用户结构体数组值和软件结构体数组值，以确定数组中实际使用长度）

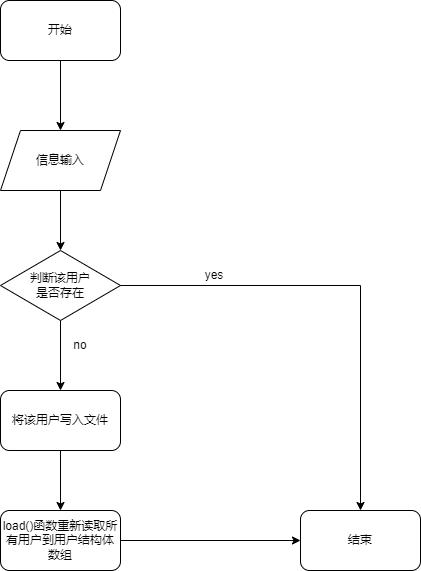
（1

（2）加载数据函数流程图（将数据从文件取到内存区）

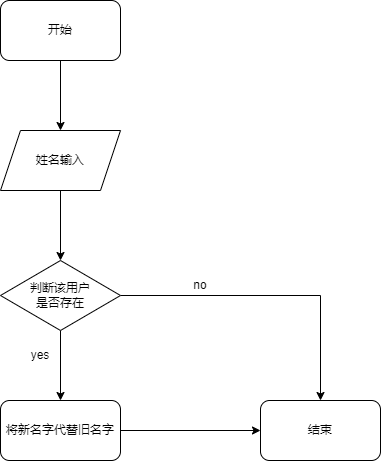
（3）保存数据函数流程图（将内存中分配的用户结构体数组保存到内存中）



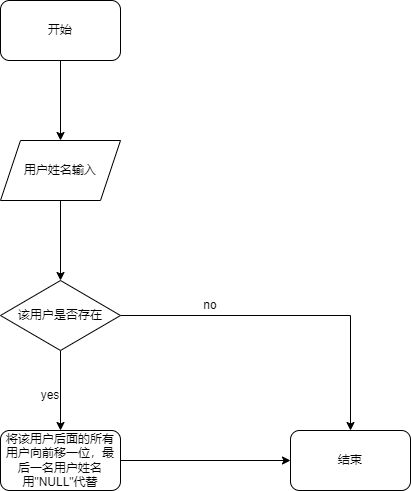
（4）添加用户函数流程图

****

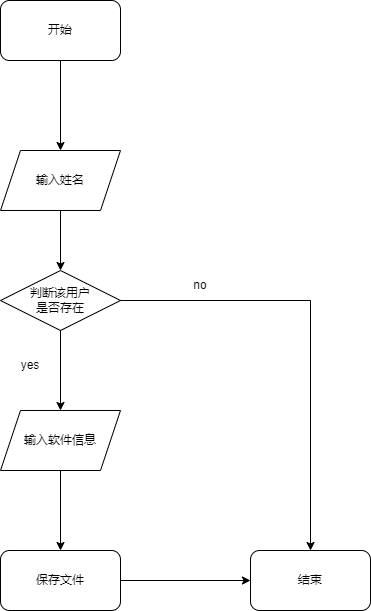
（5）修改用户姓名函数流程图



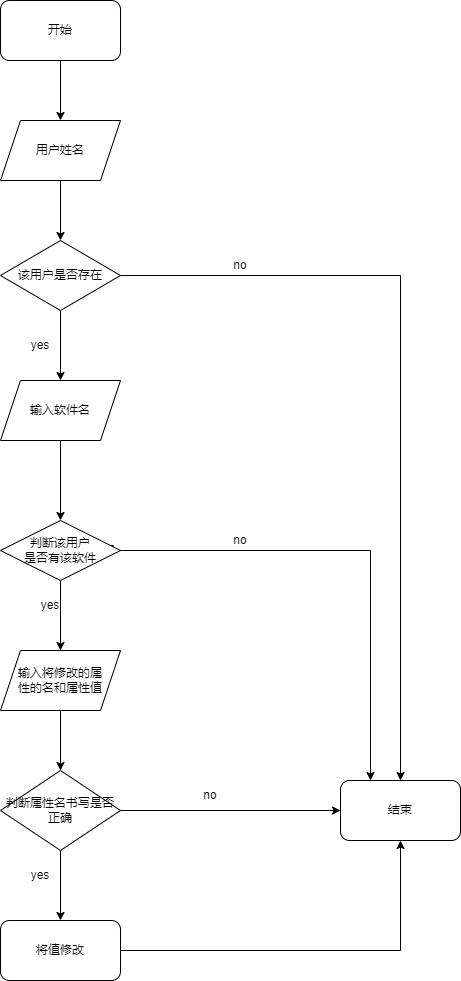
（6）删除用户函数流程图



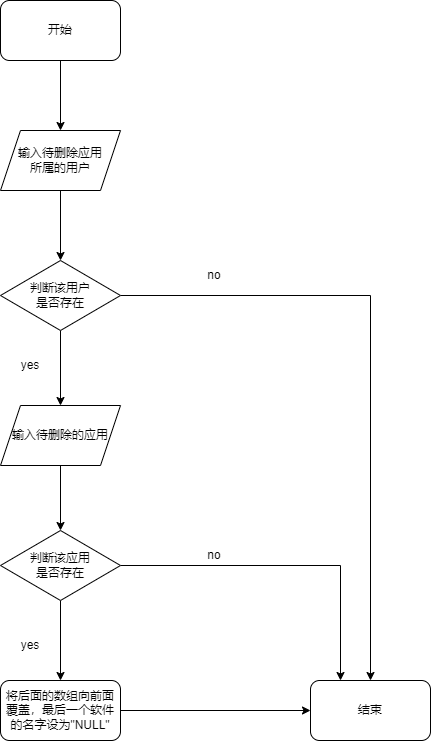
（7）添加用户软件函数流程图



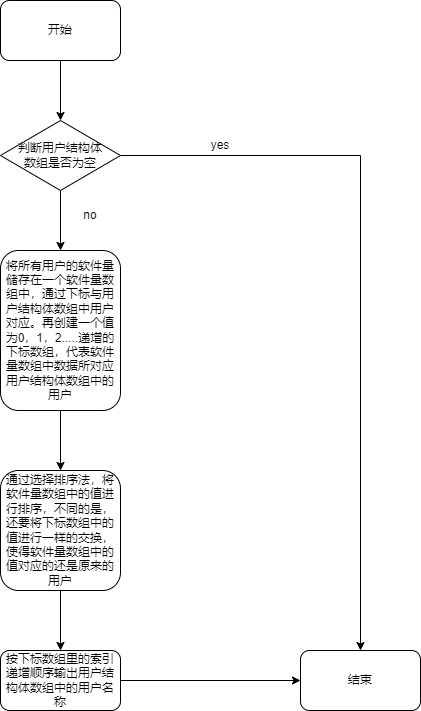
（8）修改软件信息函数流程图



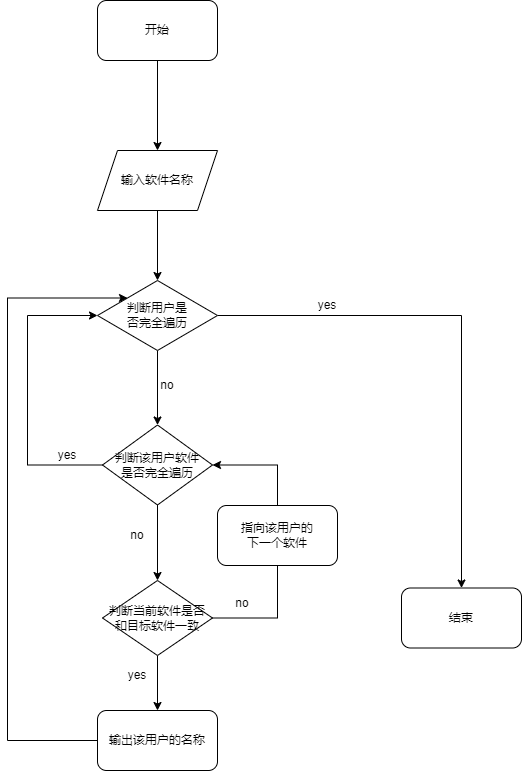
（9）删除用户软件函数流程图



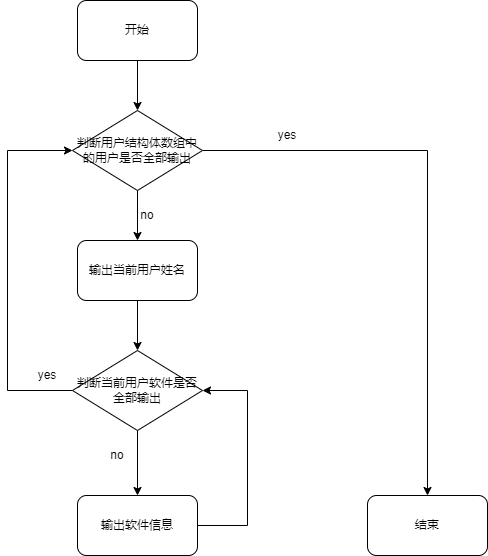
（10）按软件量将用户降序排序函数流程图



（11）找寻软件拥有者函数流程图



（12）浏览所有用户与其软件信息函数流程图



4 具体代码实现

#include <stdio.h>

#include <string.h>

#include <io.h>

void fu\_chu\_zhi(Person\_information);//模块函数声明

void Load(Person\_information);

void upload(Person\_information);

void Add\_person(Person\_information);

void Modify\_person(Person\_information);

void Deleted\_person(Person\_information);

void Add\_person\_app(Person\_information);

void Modify\_person\_app(Person\_information);

void Deleted\_person\_app(Person\_information);

void Sorted\_person\_Num\_apps(Person\_information);

void Find\_app\_owned\_person(Person\_information);

void Look\_All(Person\_information);

typedef struct App {//软件结构体，包含软件的信息

char Name[20];

float Storage;

char Developer[20];

char Version[10];

char Language\_CHINA[5];

}App;

typedef struct Person\_information {//用户结构体

char Name[20];

App Own\_App[10];//包含了一个大小为十的软件结构体数组，用于存储软件的信息

}Person\_information;

int main() {//图形界面的打印

printf("----------------------------------------\n各种可执行的操作及对应的编号如下：\n");

printf("1, 添加用户信息\n2, 修改用户姓名\n3, 删除用户\n");

printf("4,添加已有用户拥有软件\n5,修改已有用户拥有软件的信息\n6,删除已有用户软件信息\n");

printf("7,按每个用户拥有的应用数进行降序排序\n8,查找应用的所属用户\n");

printf("9,查看所有用户信息\n");

printf("输入其余任意键,退出程序\n----------------------------------------\n");

Person\_information All\_person[100];//在内存中开辟了一个大小为100的用户结构体数组，存储用户信息，便于数据的访问和操作

Person\_information\* All\_person\_pointer = &All\_person;//将用户结构体数组的地址赋值给All\_person\_pointer指针，便于后续函数传参

fu\_chu\_zhi(All\_person\_pointer);

Load(All\_person\_pointer);//将文件中的数据加载到原先开辟好的用户结构体数组中

while (1) {//按照用户输入的值进行匹配，while循环保证了在不进行人为终止的情况下程序会一直运行下去

printf("\n请按你想要执行的操作输入对应的编号：");

int a;

scanf\_s("%d", &a);

switch (a) {

case 1:Add\_person(All\_person\_pointer); break;

case 2:Modify\_person(All\_person\_pointer); break;

case 3:Deleted\_person(All\_person\_pointer); break;

case 4:Add\_person\_app(All\_person\_pointer); break;

case 5:Modify\_person\_app(All\_person\_pointer); break;

case 6:Deleted\_person\_app(All\_person\_pointer); break;

case 7:Sorted\_person\_Num\_apps(All\_person\_pointer); break;

case 8:Find\_app\_owned\_person(All\_person\_pointer); break;

case 9:Look\_All(All\_person\_pointer); break;

default:printf("程序运行结束");upload(All\_person\_pointer);return 0;//upload函数将用户结构体数组中的数据存回文件中，随后文件关闭

}

}

}

void fu\_chu\_zhi(Person\_information\* p) {//对用户结构体数组的部分位置做标记，将文件读取至用户结构体数组中时，未被覆盖的区域标记未变，意为“空”，可以确定数组的实际利用大小

for (int i = 0; i < 100; i++) {

strcpy\_s((p+i)->Name,5,"NULL");//将用户结构体数组的每一个用户名设置为“NULL”

for (int j = 0; j < 10; j++)

strcpy\_s((p + i)->Own\_App[j].Name ,5,"NULL");//将所有用户包含的软件结构体数组的所有软件名标记为“NULL”

}

}

void Load(Person\_information\* p) {//将文件中的数据输出到用户结构体数组中

FILE\* fp;

if (\_access("D:\project\c\Personal app system\Personapp", 0)) {//判断文件是否存在

fopen\_s(&fp, "D:\project\c\Personal app system\Personapp", "w");//文件不存在则以只读方式创建一个

fclose(fp);

FILE\* fp;

printf("该文件未存在，以为你创建了一个不含任何数据的新文件！\n此次运行软件，请执行操作1-添加用户，防止出现错误格式！\n");

return 0;

}

fopen\_s(&fp, "D:\project\c\Personal app system\Personapp", "r");//以只读方式打开文件，将文件地址赋值给fp指针

int i = 0;

while (1) {//while循环，当文件读取到末尾时，会被打破

if (feof(fp)) break; //读取到文件末尾时，feof(fp)会放回一个非零值，打破循环

else {//将文件中的数据读取到用户结构体数组中

fscanf\_s(fp, "%s", (p + i)->Name,20);

fscanf\_s(fp, "%s", (p + i)->Own\_App[0].Name,20);

fscanf\_s(fp, "%f", &(p + i)->Own\_App[0].Storage);

fscanf\_s(fp, "%s", (p + i)->Own\_App[0].Developer,20);

fscanf\_s(fp, "%s", (p + i)->Own\_App[0].Version,10);

fscanf\_s(fp, "%s", (p + i)->Own\_App[0].Language\_CHINA,5);

char endyn[5]={0};

fscanf\_s(fp, "%s",endyn, 5);//读取每个应用后特地设置的flag，用于确定当前应用接下来是当前用户的下一个应用还是下一个用户

for(int j=1;endyn[0] == 'n'; j++) {//当flag是以n开头时，可确定当前用户的软件还没有输入完毕；继续输入

fscanf\_s(fp, "%s", (p + i)->Own\_App[j].Name, 20);

fscanf\_s(fp, "%f", &(p + i)->Own\_App[j].Storage);

fscanf\_s(fp, "%s", (p + i)->Own\_App[j].Developer, 20);

fscanf\_s(fp, "%s", (p + i)->Own\_App[j].Version, 10);

fscanf\_s(fp, "%s", (p + i)->Own\_App[j].Language\_CHINA, 5);

fscanf\_s(fp, "%s", endyn, 5);//继续读取该软件后的flag

}

i++;//当前用户软件读取完后，将用户结构体数组的下标向后移动一位，继续输入下一个用户的数据

}

}fclose(fp);//关闭文件

}

void upload(Person\_information\* p) {//将结构体数组中的用户输入到文件中

FILE\* fp;

fopen\_s(&fp, "D:\project\c\Personal app system\Personapp", "w");//以读入的方式打开文件

for (int i = 0;; i++) {//循环输入每一个用户名和软件信息

fprintf(fp,"%s ", p[i].Name);

for (int j = 0;; j++) {//循环输入当前用户的每一个软件信息

fprintf(fp,"%s ", p[i].Own\_App[j].Name);

fprintf(fp,"%f ", p[i].Own\_App[j].Storage);

fprintf(fp,"%s ", p[i].Own\_App[j].Developer);

fprintf(fp,"%s ", p[i].Own\_App[j].Version);

fprintf(fp,"%s ", p[i].Own\_App[j].Language\_CHINA);

if (!strcmp(p[i].Own\_App[j + 1].Name, "NULL"))//当下一个软件的名称为“NULL”时，软件输入完毕

break;

fprintf(fp,"%s ", "no");//输入flag，接着输入当前用户的下一个软件

}

fprintf(fp,"%s ", "yes");//输入flag，寓意当前用户软件输入完毕

if (!strcmp(p[i + 1].Name, "NULL")) break;//下一个用户名为“NULL”时，用户输入完毕

}fclose(fp);

}

void Add\_person(Person\_information\* p) {//添加用户

FILE\* fp;

fopen\_s(&fp, "D:\project\c\Personal app system\Personapp", "a");//以追加写入的方式打开文件

char name[20];

int num;

printf("请输入需添加用户的姓名：");

scanf\_s("%s", name,20);

for (int i = 0;; i++) {//将输入的姓名与已存在的用户姓名进行对比，检测该用户是否存在

if (!strcmp(p[i].Name, name)) {

printf("该用户已存在！");

return 0;

}

if (!strcmp(p[i + 1].Name, "NULL"))break;//用户都遍历完都没有重名的，则退出当前循环

}

fprintf(fp, "%s ", name);//将用户姓名写入文件

printf("请输入该用户需添加的软件数目：");

scanf\_s("%d", &num);

for (int i = 1; i <= num; i++) {//将软件信息写入文件

printf("请分别输入第%i个软件的名字，容量，开发商，版本号，是否支持中文等情况\n", i);

char Name[20];

float Storage;

char Developer[20];

char Version[10];

char Language\_CHINA[5];

scanf\_s("%s", Name,20);

scanf\_s("%f", &Storage);

scanf\_s("%s", Developer,20);

scanf\_s("%s", Version,10);

scanf\_s("%s", Language\_CHINA,5);

fprintf(fp, "%s ", Name);

fprintf(fp, "%f ", Storage);

fprintf(fp, "%s ", Developer);

fprintf(fp, "%s ", Version);

fprintf(fp, "%s ", Language\_CHINA);

if(i==num)fprintf(fp, "%s ", "yes");//在每一个软件后做特殊标记，确定该软件接下来是别的软件还是别的用户

else fprintf(fp,"%s ","no");

}

fclose(fp);

Load(p);//将数据从已更新的文件中读取出来，确保用户结构体数组的同步更新

}

void Modify\_person(Person\_information\* p) {//修改用户姓名

printf("请输入你想修改用户的姓名：");

char Name\_modified[20];

scanf\_s("%s", Name\_modified,20);//输入修改前的姓名，以供查找

for (int i = 0;; i++) {//对用户姓名进行遍历匹配

if (!strcmp(p[i].Name, Name\_modified)) {//匹配成功

char Name\_to\_be\_modified[20];

printf("请输入修改后的用户姓名：");//输入修改后的姓名，以供修改

scanf\_s("%s", Name\_to\_be\_modified,20);

strcpy\_s(p[i].Name,20, Name\_to\_be\_modified);

printf("已将名为%s修改为%s", Name\_modified, Name\_to\_be\_modified);

upload(p);//存到文件中

break;

}

if (!strcmp(p[i + 1].Name, "NULL")) { //匹配失败

printf("不好意思，没有找到你的目标用户！");

break;

}

}

}

void Deleted\_person(Person\_information\* p) {//删除用户

printf("请输入你想删除用户的姓名:");

char name\_Deleted[20];

scanf\_s("%s", name\_Deleted,20);//输入用户姓名

for (int i = 0;; i++) {//循环遍历

if (!strcmp(p[i].Name, name\_Deleted)) {//匹配成功

for (int j = i;; j++) {

strcpy\_s(p[j].Name, 20, p[j + 1].Name);//从待删除的用户开始，将当前用户姓名用下一个用户姓名覆盖

for (int k = 0;k<10; k++) {//从待删除的用户开始，将当前用户软件结构体数组对应下标位置用下一个用户软件覆盖

strcpy\_s(p[j].Own\_App[k].Name, 20, p[j + 1].Own\_App[k].Name);

p[j].Own\_App[k].Storage = p[j + 1].Own\_App[k].Storage;

strcpy\_s(p[j].Own\_App[k].Developer, 20, p[j + 1].Own\_App[k].Developer);

strcpy\_s(p[j].Own\_App[k].Version, 10, p[j + 1].Own\_App[k].Version);

strcpy\_s(p[j].Own\_App[k].Language\_CHINA, 5, p[j + 1].Own\_App[k].Language\_CHINA);

if (!strcmp(p[j + 1].Own\_App[k + 1].Name, "NULL")) {//当下一个用户的下一个软件名为“NULL”时

strcpy\_s(p[j].Own\_App[k + 1].Name,5, "NULL");//当前用户的下一个软件名设置为“NULL”，防止当前用户的软件数目多余下一个用户

break;

}

}

if (!strcmp(p[j + 2].Name, "NULL")) {//下下个用户姓名为“NULL”时

strcpy\_s(p[j + 1].Name, 20, "NULL");//将下一个用户姓名设置为“NULL”，因为下一个用户是最后一个用户，其后没有可覆盖他的

break;

}

}

return 0;

}

else if (!strcmp(p[i + 1].Name, "NULL")) {//用户姓名匹配失败

printf("对不起，没有找到该用户\n");

return 0;

}

}

}

void Add\_person\_app(Person\_information\* p) {//添加用户软件

printf("请输入你想添加软件的用户的姓名：");

char Name\_add\_app[20];

scanf\_s("%s", Name\_add\_app, 20);//输入目标用户姓名

for (int i = 0;; i++) {//循环遍历用户结构体中用户姓名，匹配目标用户姓名

if (!strcmp(p[i].Name, Name\_add\_app)) {//匹配成功

for (int j = 0;; j++)//将软件结构体数组的下标向后推进

if (!strcmp(p[i].Own\_App[j].Name, "NULL")) {//确保该用户结构体的软件结构体数组下标到达空的位置，进行新软件的添加操作

printf("请分别输入将添加软件的名字，容量，开发商，版本号，是否支持中文等情况\n", i);

char Name[20];

float Storage;

char Developer[20];

char Version[10];

char Language\_CHINA[5];

scanf\_s("%s", Name, 20);

scanf\_s("%f", &Storage);

scanf\_s("%s", Developer, 20);

scanf\_s("%s", Version, 10);

scanf\_s("%s", Language\_CHINA, 5);

strcpy\_s(p[i].Own\_App[j].Name, 20, Name);

p[i].Own\_App[j].Storage = Storage;

strcpy\_s(p[i].Own\_App[j].Developer, 20, Developer);

strcpy\_s(p[i].Own\_App[j].Version, 10, Version);

strcpy\_s(p[i].Own\_App[j].Language\_CHINA, 5, Language\_CHINA);

upload(p);

upload(p);

return 0;

}

}

if (!strcmp(p[i + 1].Name, "NULL")) {//匹配失败

printf("不好意思，没有找到你的目标用户！\n");

return 0;

}

}

}

void Modify\_person\_app(Person\_information\* p) {//修改用户软件信息

char Name[20], App[20], Attrrbute[20], Attrubute\_zhi[20];//代表用户名，软件名，属性名，修改后的属性值

printf("请输入你目标用户名：");

scanf\_s("%s", Name,20);

for (int i = 0;; i++) {//推进用户结构体数组，将目标用户名数组中的用户名匹配

if (!strcmp(p[i].Name, Name)) {//用户名匹配成功

printf("请输入将修改属性的软件名称:");

scanf\_s("%s", App, 20);

for (int j = 0;; j++) {//推进当前用户的软件结构体数组下标，将目标软件名和数组中的软件吗匹配

if (!strcmp(p[i].Own\_App[j].Name, App)) {//匹配成功

while (1) {

printf("请输入将修改的属性和修改后的值：");

scanf\_s("%s", Attrrbute,20);//输入属性名，将其与软件的每个属性的名字进行对比

if (!strcmp(Attrrbute, "Storage")) {//若输入的属性名与与“Storage”相同，则修改当前软件的容量大小，下列判断同意

float Attribute\_F\_zhi;

scanf\_s("%f", &Attribute\_F\_zhi);

p[i].Own\_App[j].Storage = Attribute\_F\_zhi;

upload(p);

printf("修改成功！");

return 0;

}

scanf\_s("%s", Attrubute\_zhi,20);

if (!strcmp(Attrrbute, "Name")) {

strcpy\_s(p[i].Own\_App[j].Name, 20, Attrubute\_zhi);

upload(p);

printf("修改成功！");

return 0;

}

if (!strcmp(Attrrbute, "Developer")) {

strcpy\_s(p[i].Own\_App[j].Developer, 20, Attrubute\_zhi);

upload(p);

printf("修改成功！");

return 0;

}

if (!strcmp(Attrrbute, "Version")) {

strcpy\_s(p[i].Own\_App[j].Version, 10, Attrubute\_zhi);

upload(p);

printf("修改成功！");

return 0;

}

if (!strcmp(Attrrbute, "Language\_CHINA")) {

strcpy\_s(p[i].Own\_App[j].Language\_CHINA, 5, Attrubute\_zhi);

upload(p);

printf("修改成功！");

return 0;

}//全都匹配失败

printf("系统检测你所输入的属性名有误，请按Name,Developer,Version,Language\_CHINA进行输入，谢谢");

}

}

if (!strcmp(p[i].Own\_App[j + 1].Name, "NULL")) {//软件名匹配失败

printf("不好意思，没有找到你的目标软件！\n");

return 0;

}

}

if (!strcmp(p[i + 1].Name, "NULL")) {//用户名匹配失败

printf("不好意思，没有找到你的目标用户！\n");

return 0;

}

}

}

}

void Deleted\_person\_app(Person\_information \* p) {//删除用户软件

printf("请输入你想删除的应用的所属用户：");

char App\_Name[20];

scanf\_s("%s", App\_Name,20);

for (int i = 0;; i++) {//循环推进用户结构体数组次下标，遍历用户名，与目标用户名匹配

if (!strcmp(App\_Name, p[i].Name)) {//匹配成功

char App\_deleted[20];

printf("请输入你想删除的应用:");

scanf\_s("%s", App\_deleted,20);

for (int j = 0;; j++) {//循环推进软件结构体数组下标，遍历软件名，与目标软件名相匹配

if (!strcmp(App\_deleted, p[i].Own\_App[j].Name)) {//匹配成功

for (int k = j;; k++) {//循环推进软件结构体数组下标，对每个软件实行覆盖操作

if (!strcmp(p[i].Own\_App[k + 1].Name,"NULL")) {

strcpy\_s(p[i].Own\_App[k].Name, 5, "NULL");//将最后一个软件的名字设置为空

return 0;

}//从当前软件开始，用后面的软件覆盖前一个软件的信息

strcpy\_s(p[i].Own\_App[k].Name, 20, p[i].Own\_App[k + 1].Name);

p[i].Own\_App[j].Storage = p[i].Own\_App[j].Storage;

strcpy\_s(p[i].Own\_App[k].Developer, 20, p[i].Own\_App[k + 1].Developer);

strcpy\_s(p[i].Own\_App[k].Version, 20, p[i].Own\_App[k + 1].Version);

strcpy\_s(p[i].Own\_App[k].Language\_CHINA, 20, p[i].Own\_App[k + 1].Language\_CHINA);

if (!strcmp(p[i].Own\_App[k + 2].Name, "NULL")) {

strcpy\_s(p[i].Own\_App[k+1].Name, 5, "NULL");

printf("%s已经被删除成功!", App\_deleted);

return 0;

}

}

}

else if (!strcmp(p[i].Own\_App[j + 1].Name, "NULL")) {//软件名匹配失败

printf("不好意思，没有找到该应用\n");

return 0;

}

}

}

else if (!strcmp(p[i + 1].Name, "NULL")) {//用户名匹配失败

printf("不好意思，没有找到该用户！");

return 0;

}

}

}

void Sorted\_person\_Num\_apps(Person\_information\* p) {//按用户拥有的软件量进行降排序

if(!strcmp(p[0].Name,"NULL")) {//检查用户结构体数组是否为空

printf("用户列表为空。\n");

return 0;

}

int Person\_Owned\_nums[100];//储存用户软件量的数组，与用户结构体数组里的用户一一对应，通过下标建立关系

for (int i = 0;; i++) {//循环遍历每个用户结构体，将用户所有软件量存入软件量数组中

if (strcmp(p[i].Name, "NULL")) {//用户名与“NULL”不相同时，证明用户未遍历结束，执行一下操作

for (int num = 0;;) {//将每个用户的软件量初始化为0

if (!strcmp(p[i].Own\_App[0].Name, "NULL")) {//用户第一个软件名为“NULL”

Person\_Owned\_nums[i] = 0;//设置该用户软件量为0

break;

}

num++;//数量加一

if (!strcmp(p[i].Own\_App[num].Name, "NULL")) {//第i个软件的名字若为“NULL”，则证明当前用户软件遍历结束

Person\_Owned\_nums[i] = num;//将软件量数组对应位置设置为num

break;

}

}

}

else {

int Xia\_Biao[100];//创建一个下标数组，其中的值为0，1，2，3......，目标是在将软件量数组进行排序时，将下标数组中对应位置的值进行相同的排序，确保软件量数组中的值在用户结构体数组对应的用户可以找得到

for (int k = 0; k < 100; k++) Xia\_Biao[k] = k;

int maxxiabiao = 0;//选择排序法

for (int z = 0; z < i; z++) {

int max = Person\_Owned\_nums[z];

for (int j = z; j < i; j++) {

if (Person\_Owned\_nums[j] > max) {

maxxiabiao = j;

max = Person\_Owned\_nums[maxxiabiao];

}

}

int tempt = Person\_Owned\_nums[z];//将软件量数组中的值进行降序排序

Person\_Owned\_nums[z] = Person\_Owned\_nums[maxxiabiao];

Person\_Owned\_nums[maxxiabiao] = tempt;

tempt = Xia\_Biao[z];//将下标数组中的值进行相同的操作，使得软件量数组和下标数组中的值的仍然保持一一对应

Xia\_Biao[z] = Xia\_Biao[maxxiabiao];

Xia\_Biao[maxxiabiao] = tempt;

}

printf("按每个用户拥有的应用数进行降序排序结果如下：\n");

for (int s = 0; s < i; s++) {//将软件量数组进行排序，其在下标数组中所对应的值--用户结构体数组中的用户下标，也会随之排好序

printf("%s:%d个应用\n", p[Xia\_Biao[s]].Name, Person\_Owned\_nums[s]);//将下标数组中的值对应的用户结构体数组中的用户一一输出即可

}

return 0;

}

}

}

void Find\_app\_owned\_person(Person\_information\* p) {//找到拥有指定软件的用户

char Name[20];

int num\_own = 0;

printf("请输入你想查询的软件：");

scanf\_s("%s", Name,20);

for (int i=0; i < 100; i++) {//遍历循环所有用户

for (int j = 0; j < 10; j++) {//遍历循环单个用户的所有软件

if (!strcmp(p[i].Own\_App[j].Name, Name)) {//匹配成功

printf("%s\t", p[i].Name);//输出当前用户姓名

num\_own++;//flag加一

}

if (!strcmp(p[i].Own\_App[j + 1].Name, "NULL"))

break;

}

if (!strcmp(p[i + 1].Name, "NULL")) break;//下一个用户的姓名为“NULL”，则说明遍历结束

}

if (num\_own == 0)printf("未查询到拥有此应用的用户！");//如果flag等于0；则说明没有匹配的用户

else printf("\n以上便是拥有此应用的用户");

}

void Look\_All(Person\_information\* p) {//输出所有用户及其所有软件的信息

printf("\n注：排版格式：\n用户序号.用户名：\n 软件名 储存容量 开发商 版本号 是否支持中文\n ............\n");//格式提示

printf("==================================================\n");

for (int i = 0; i < 100; i++) {//循环遍历所有用户

printf("%d.%s:\n",i+1, p[i].Name);//输出用户姓名

for (int j = 0; j < 10; j++) {//循环输出用户的软件结构体数组的软件信息

printf(" %s", p[i].Own\_App[j].Name);

printf(" %f ", p[i].Own\_App[j].Storage);

printf(" %s ", p[i].Own\_App[j].Developer);

printf(" %s ", p[i].Own\_App[j].Version);

printf(" %s ", p[i].Own\_App[j].Language\_CHINA);

if (!strcmp(p[i].Own\_App[j + 1].Name, "NULL"))//下一个软件名为空，软件结构体数组遍历结束

break;

printf("\b;\n");//格式控制

}

printf("\b。%c", '\n');//格式控制

if (!strcmp(p[i + 1].Name, "NULL")) break;//下一个用户姓名为“NULL”，用户结构体数组遍历结束

}

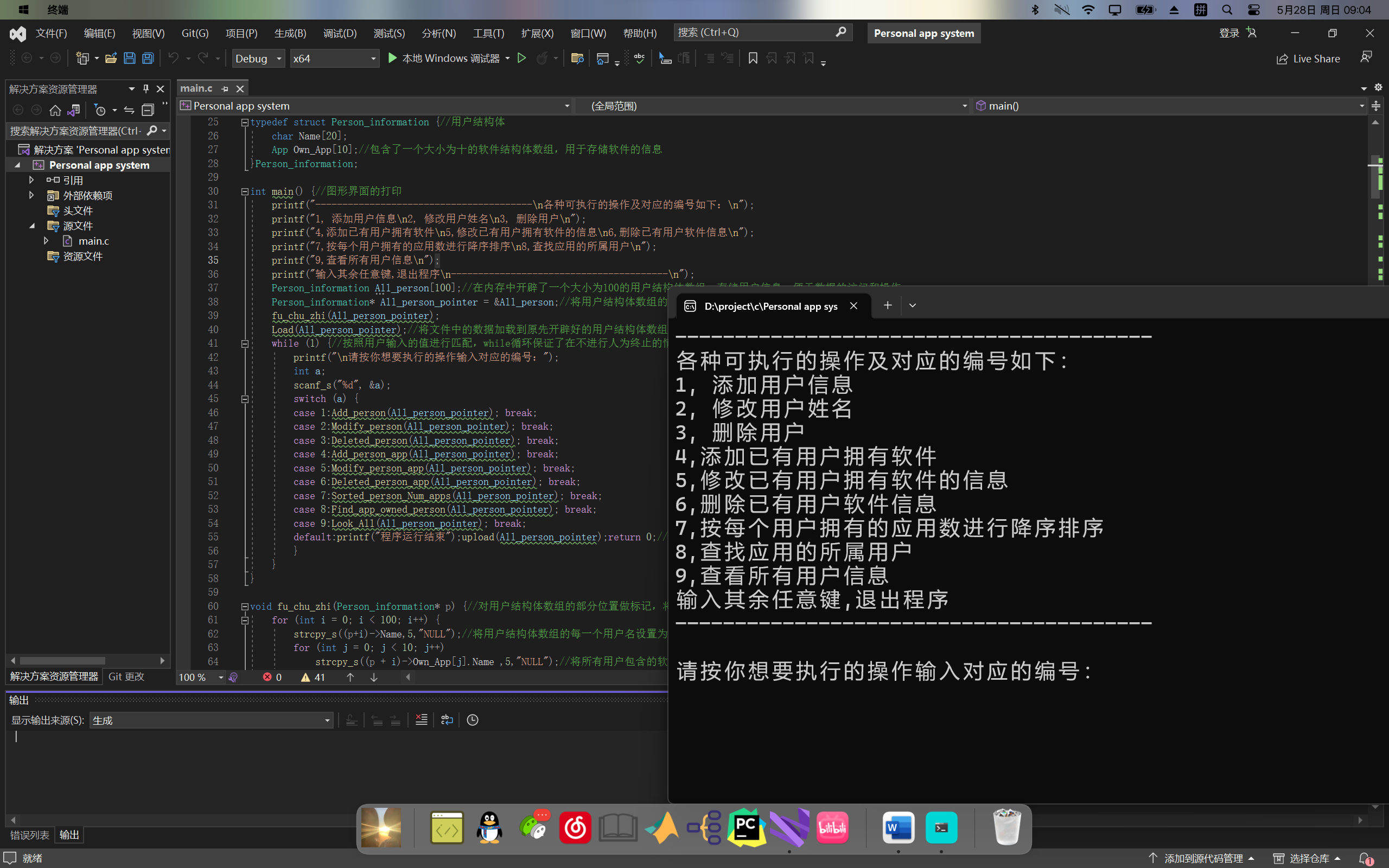
printf("==================================================\n");

}

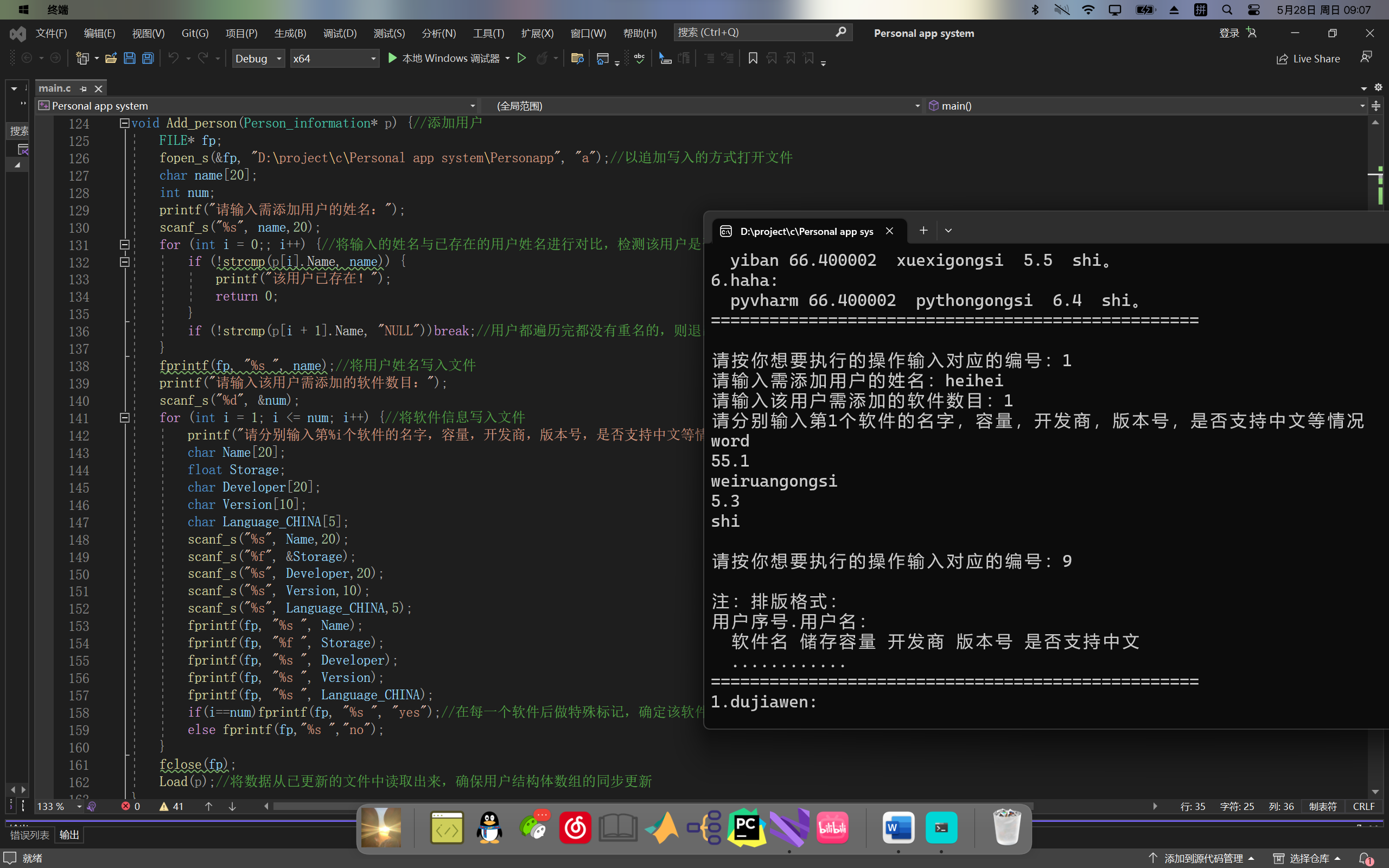
5 课程设计总结

5.1 程序运行结果

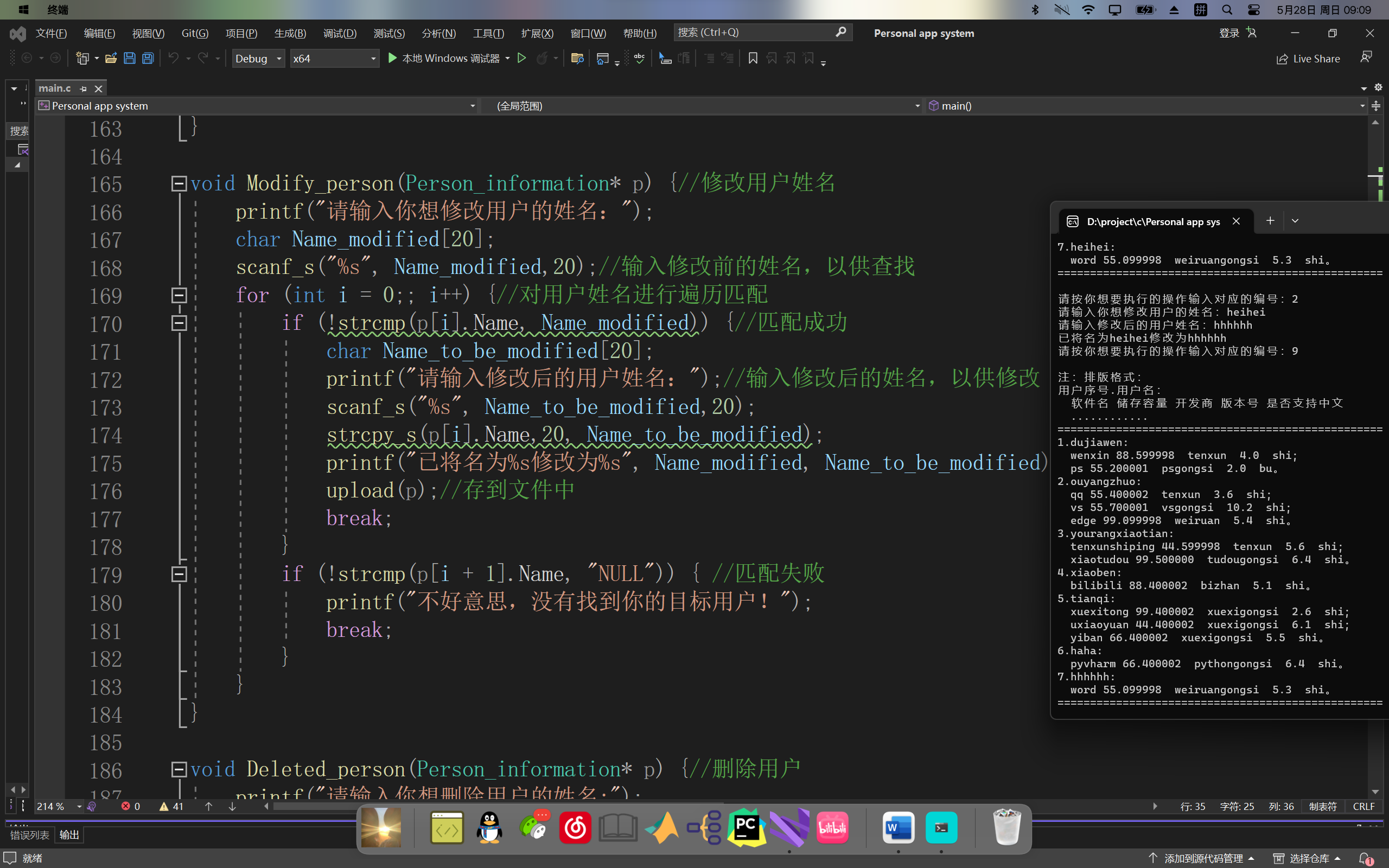
1，图文窗口的打印



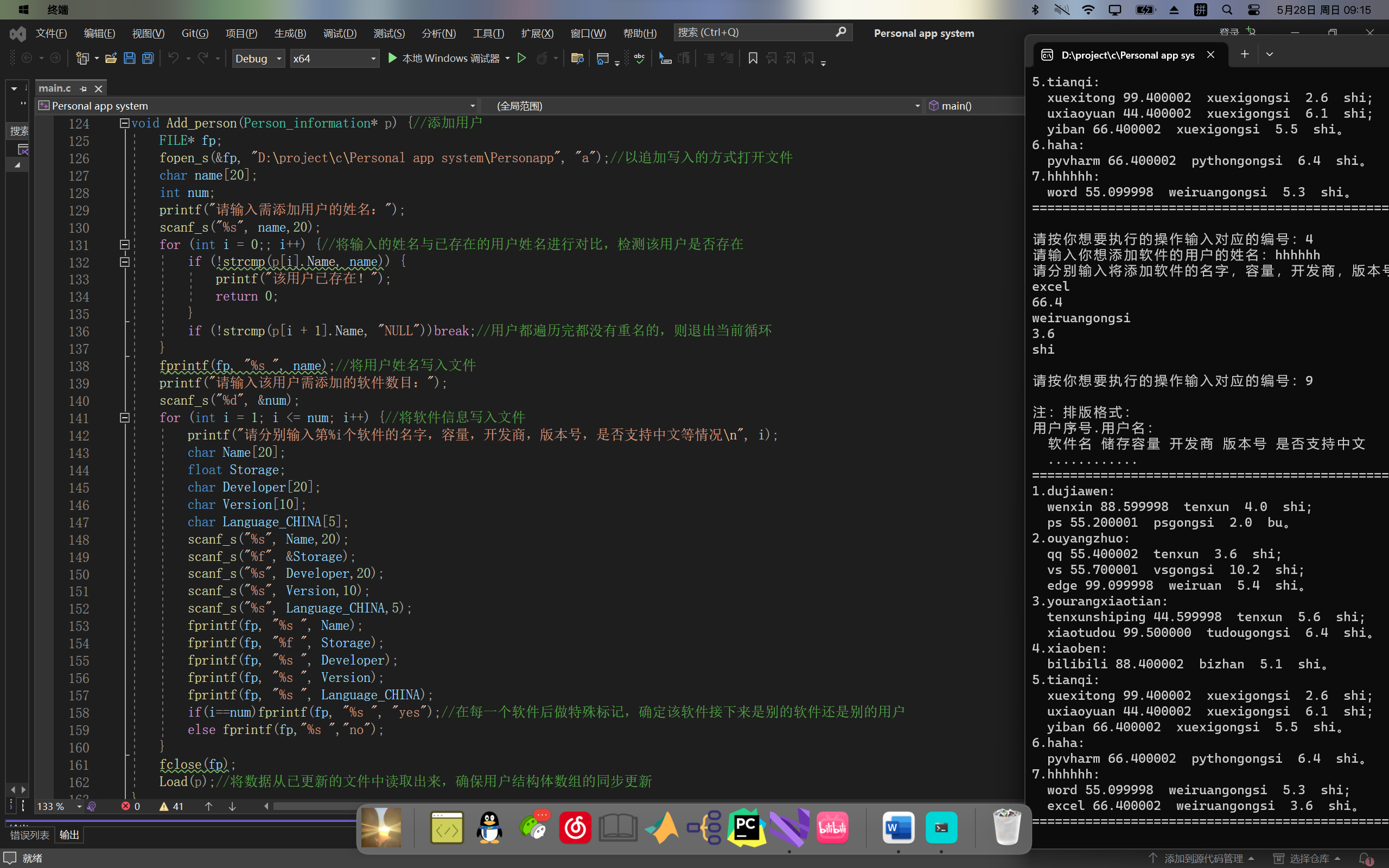
2，添加用户



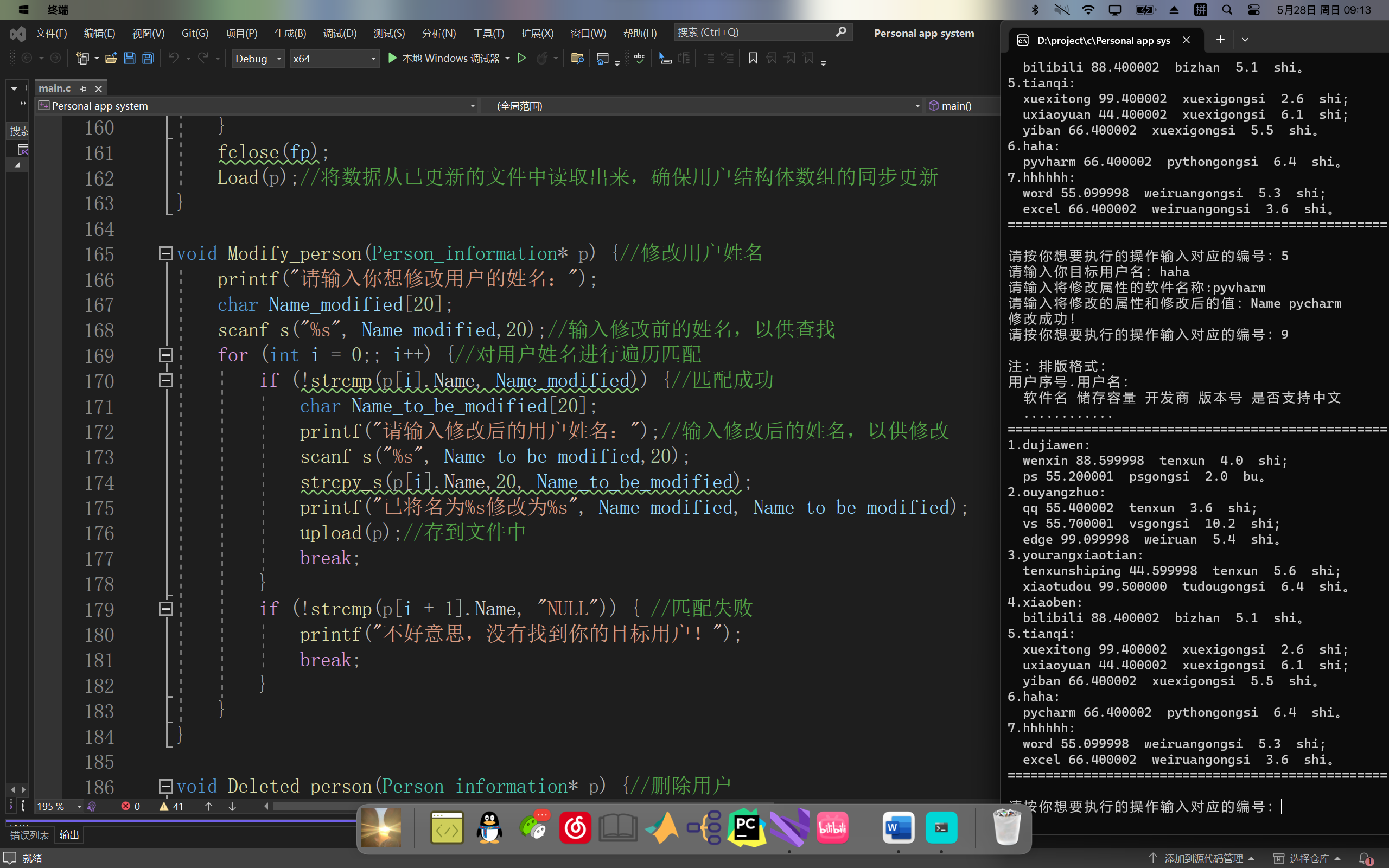
3，修改用户姓名



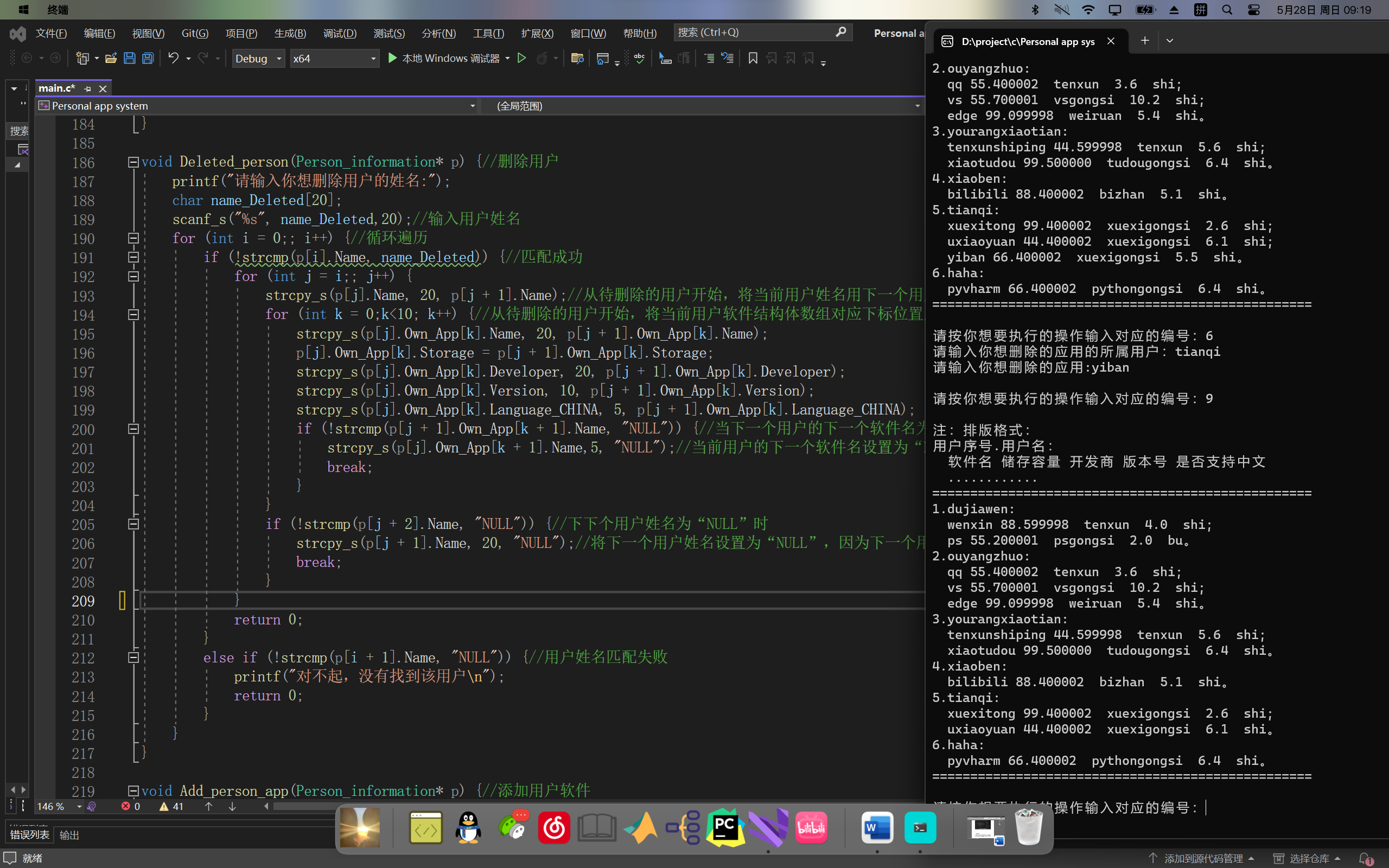
4，添加用户软件



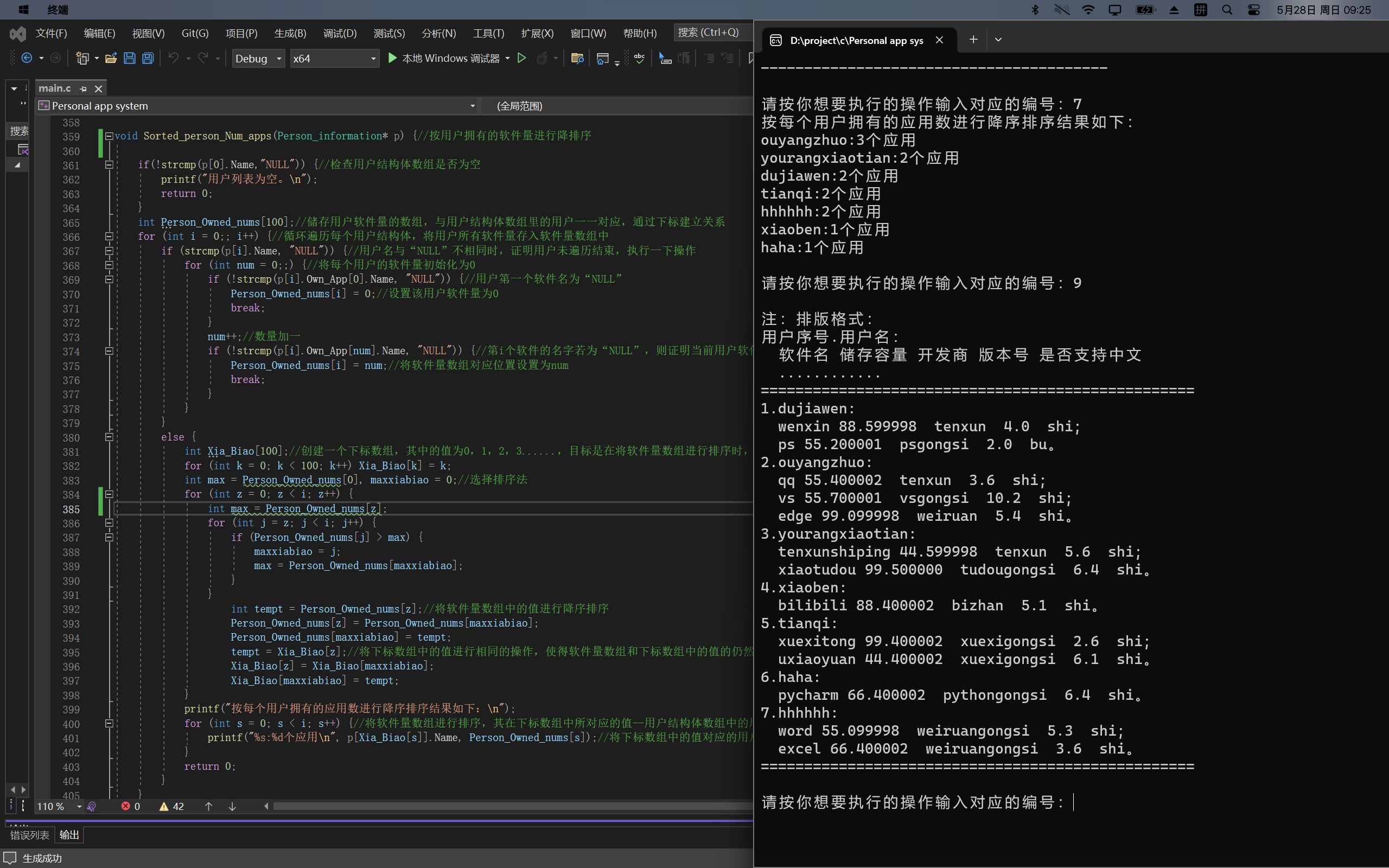
5，修改软件信息



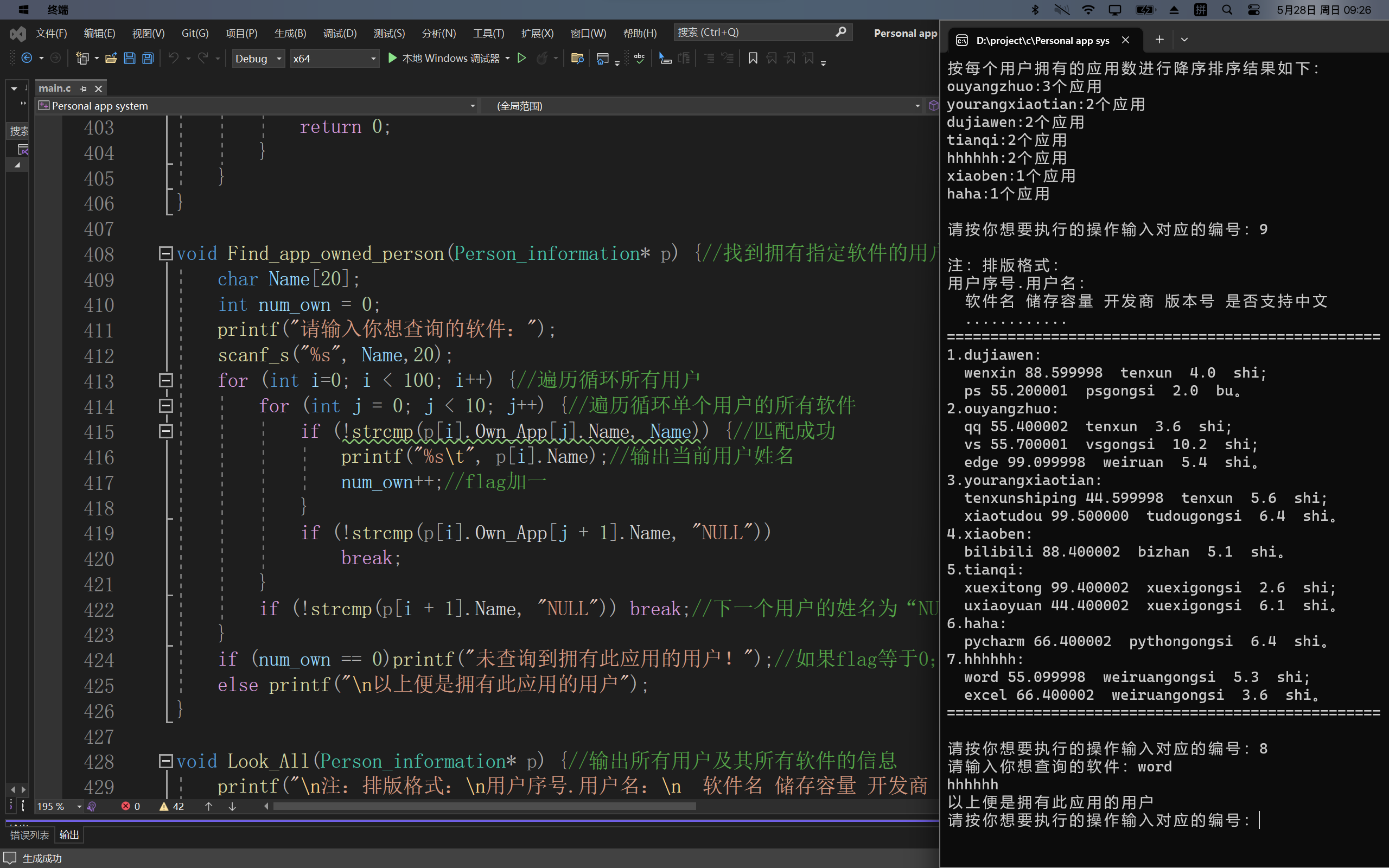
6，删除用户软件



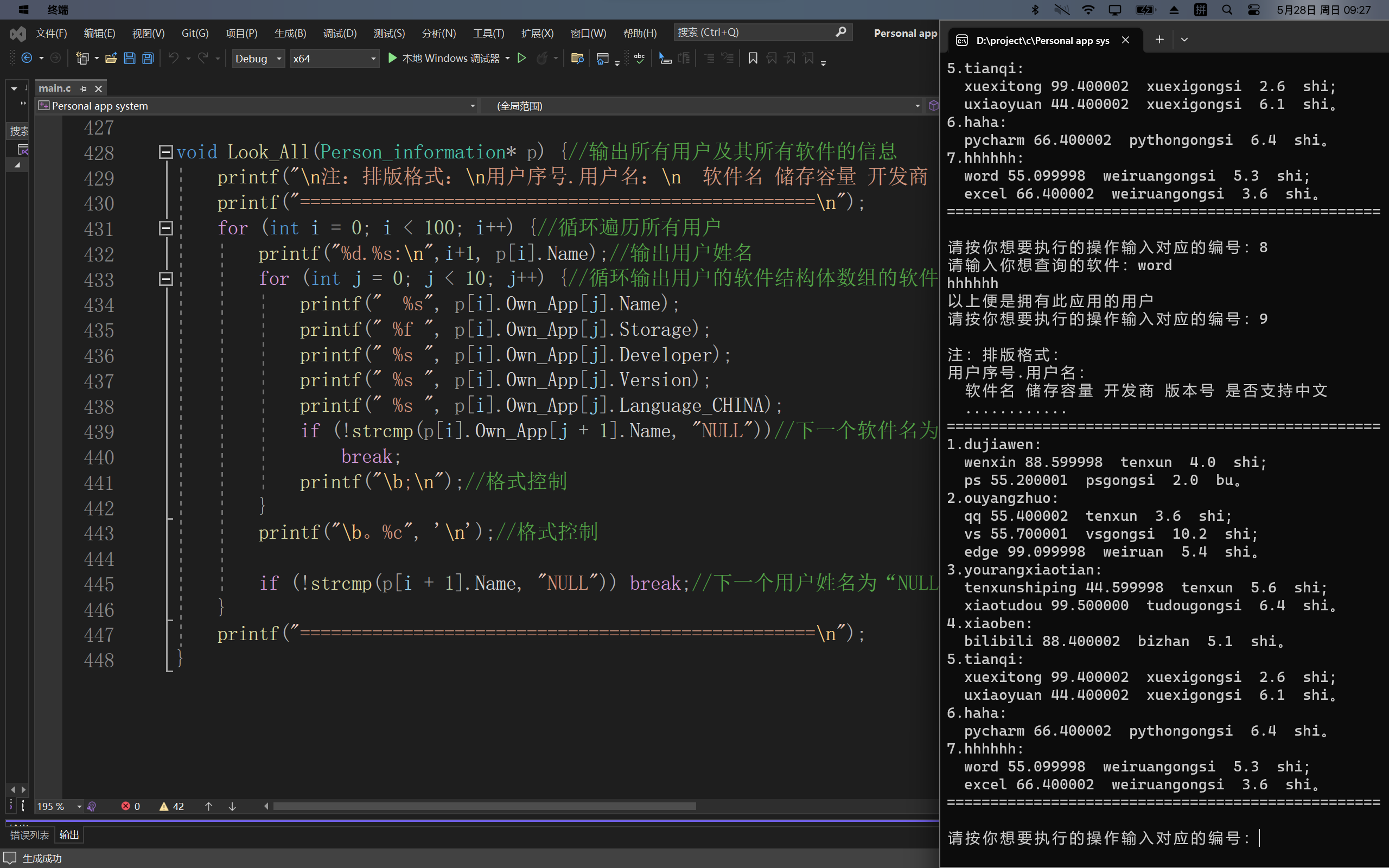
7，按用户软件拥有量进行降序排序



8，寻找拥有指定软件的用户



9，展示所有用户及其软件信息



5.2 课程设计体会

在本次项目编写过程中，我们将各种数据类型结合在了一起，各种数据类型的使用蕴含其中。程序编写的不足之处，有对涉及文件操作的项目有些许的不熟悉，本应该是将文件中的内容读取出来后，接下都是对内存中的数据进行操作，但是我们在将文件中的内容读取出来后，在进行添加用户时，是选择将用户添加到文件中，然后再将文件中的数据再次读取到内存中，大大降低的程序运行效率，在下次编写程序时会提起注意。

参考文献

[1]Stephen. Prata C++ Primer Plus：中文（第六版）.人民邮电出版社，2013.

[2]张玉生，刘炎，张亚红. C语言程序设计. 上海交通大学出版社，2018.

致 谢  
 首先由衷感谢我的指导老师。感谢您在整个用户软件信息管理系统项目中的耐心指导和教诲。您的专业知识、丰富经验以及对C语言的深入理解为我提供了宝贵的指导和启发。您的支持和鼓励让我能够克服困难，顺利完成这个项目。  
 我要特别感谢开发团队中的每一位成员。感谢你们的辛勤工作和卓越贡献。我们共同克服了一个又一个的技术难题，合作开发了这个用户软件信息管理系统。你们的专业知识和技术能力为系统的设计与实现提供了坚实的基础。  
 同时，我要感谢测试团队的成员。你们付出了大量的时间和精力来确保系统的质量和稳定性。通过你们的不懈努力，我们能够及时发现并解决了系统中的bug和问题，保证了系统的可靠性和用户体验。  
 我还要感谢用户界面设计师和用户体验研究员。你们的专业设计和用户研究为系统的界面和交互设计提供了宝贵的意见和建议。你们的努力使得系统界面美观、简洁，并且易于操作，提升了用户的满意度。  
 此外，我要感谢那些提供支持和帮助的同学和朋友。你们给予了我鼓励和支持，帮助我度过了开发过程中的困难和挑战。你们的理解和支持是我坚持下去的动力。  
 最后，我要感谢这个用户软件信息管理系统为我带来的成长和收获。通过这个项目，我学到了很多有关软件开发和团队合作的宝贵经验。这些经验将对我的未来发展产生深远的影响。  
 再次感谢每一位在用户软件信息管理系统项目中给予我支持和帮助的人。正是有了你们的支持和鼓励，我才能顺利完成这个项目。衷心感谢你们的付出和无私的奉献！  
祝愿用户软件信息管理系统能够为广大学生提供便捷和高效的服务，取得巨大的成功！