

大连理工大学《程序设计基础 A 课程设计》设计报告

校园疫情防控信息管理系统

装

订

线

学 生 姓 名: 白岩浩

院系、班级: 电子信息与电气工程学部 电计 2004

学 号: 20201072017

联 系 电 话: 13821820539

Email: 2062400572@qq.com

完 成 日 期: 2022.1.8

一、设计任务

2019 年新冠病毒发生以来，党中央迅速采取防控措施，全国各地迅速行动，联防联控，全力以赴，展开了一场全国防疫阻击战。经过全国人民持续的严防死守，无数医护人员的坚持和努力，我国疫情防控取得了突出成效。但是由于人员流动、病毒变异等原因，疫情仍然愈演愈烈。尤其是近期的奥密克戎病毒猖獗，给疫情增加了诸多不确定性。

高校人员密集，人员结构复杂，师生来自全国各地，人员流动性大，给高校疫情防控带来了极大的挑战。及时、精准和全面掌握学生每日健康信息、核酸检测统计、疫苗接种情况、离校申请报备等信息，保持信息畅通，通过信息化手段构建一条校内信息传输通道，可以给高校的疫情防控工作带来极大的便利。基于此，“大连理工大学校园疫情防控信息管理系统”的建立便格外具有意义。

该系统的目标是做到可靠地存储学生疫情防控的各方面信息。这些信息包括五部分：学生个人信息（ID、姓名、电话、邮箱、院系、班级、通讯地址）、每日健康填报记录（日期、体温、是否咳嗽、是否发烧）、核酸检测统计记录（日期、检测地、检测结果）、疫苗接种信息记录（接种次数序号、接种时间、接种地点、疫苗型号）、离校申请记录（离校原因、出校时间、返校时间、行程轨迹、审核状态）。除了添加、查找、编辑、删除等基本要求外，该系统还可以实现排序、柱状统计图统计、密码登陆等更进一步的功能。

二、程序使用说明

2.1 运行前准备

运行环境：Windows 10 – Microsoft Visual C++ 2010 Express.

运行路径：将压缩包解压后，在磁盘任何目录下均可运行主程序.

文件组成：大连理工大学疫情防控信息管理系统.exe 主程序、password.txt 用于保存密码，data.txt 用于保存学生个人信息. 所有文件均在同一目录下.

2.2 操作说明

2.2.1 用户操作逻辑

程序会首先检测本次运行是否为初次运行。如果为初次运行，则重置登陆密码为“123”。然后进入登陆界面。如果为非初次运行，则直接进入登陆界面。在登陆界面中，使用者最多有三次输入密码的机会。若密码正确，程序则显示界面。用户可以使用程序预先提供的功能。使用完毕后，用户即可推出该系统。

2.2.2 登陆界面

用户至多可以输入三次密码。超过指定次数后，用户将失去进入系统的权限，只能

再次运行本程序才能重置剩余机会。

2.2.3 菜单界面

菜单分为三个栏目，分别是“编辑”、“统计数据”、“关于”。输入 T 或 t 可以实现不同栏目之间的切换。

2.2.4 查看所有学生疫情防控填报记录

在菜单界面输入字符“1”并按下回车键，即可使用该功能。程序将一行行地输出存储的所有学生的 ID、姓名、电话、邮箱、院系、班级、通讯地址，每日健康填报日期、体温(早)、体温(中)、体温(晚)、是否咳嗽、是否发烧，核酸检测日期、检测地点、检测结果，疫苗接种次数序号、接种时间、接种地点、疫苗型号，离校申请次数序号、离校原因、出校时间、返校时间、行程轨迹起点、行程轨迹终点、审核状态。若程序检测到目前没有任何信息可供输出，则提示用户“当前内存中没有存储学生信息”！

2.2.5 查询学生个人信息

在菜单界面输入字符“2”并按下回车键，即可使用该功能。程序将提示用户输入学生 ID（例如 202101）。在用户输入完毕并按下回车键后，程序会从存储的数据中将输入的字符串与已存储的学生 ID 逐个匹配。若匹配成功，则输出仅包含该学生的疫情防控填报记录。

2.2.6 编辑学生个人信息

在菜单界面输入字符“3”并按下回车键，即可使用该功能。用户需先输入要编辑学生的 ID（例如 202101）。程序会先搜索该学生是否存在。如果不存在，则提示用户未找到，并返回菜单界面；如果存在，用户可以根据自身的要求来编辑该学生的 ID、姓名、电话、邮箱、院系、班级、通讯地址和简介，并且随时可以返回菜单。

2.2.7 删除学生个人信息

在菜单界面输入字符“4”并按下回车键，即可使用该功能。用户需先输入要删除学生的 ID。程序会先搜索该学生是否存在。如果不存在，则提示用户未找到，并返回菜单界面；如果存在，程序将删除该学生的所有信息。如果用户误触，可以随意输入一串字符串，以退出删除功能。

2.2.8 排序学生个人信息

在菜单界面输入字符“5”并按下回车键，即可使用该功能。程序将提示用户选择“降序排序”或者“升序排序”。排序完成后，用户可直接在当前界面看到排序后的结果。

2.2.9 添加学生个人信息

在菜单界面输入字符“6”并按下回车键，即可使用该功能。程序将提示用户依次输入学生的 ID、姓名、电话、邮箱、院系、班级、通讯地址。其中，输入日期时，应将年、

月、日三项数据之间添加空格，将这三个整数分隔开。用户需要注意的是，**本程序会对用户输入的数据进行合理性的检查，以实现加强输入的效果**：程序会检测用户输入的日期是否符合公历规则（例如 2077 年大于本年，2 月没有 30 日），会检测用户输入的字符串长度是否超过程序限制（例如姓名不能超过 4 个汉字）。若用户输入的数据不规范，则程序会强制要求用户重新输入，直到输入的数据符合程序限制为止。

2.2.10 添加每日健康记录

在菜单界面输入字符“x”并按下回车键，即可使用该功能。程序将提示用户依次输入学生的每日健康填报日期、体温(早)、体温(中)、体温(晚)、是否咳嗽、是否发烧。若用户输入的数据不规范，则程序会强制要求用户重新输入，直到输入的数据符合程序限制为止。

2.2.11 添加核酸检测统计

在菜单界面输入字符“y”并按下回车键，即可使用该功能。程序将提示用户依次输入学生的核酸检测日期、检测地点、检测结果。若用户输入的数据不规范，则程序会强制要求用户重新输入，直到输入的数据符合程序限制为止。

2.2.12 添加疫苗接种信息

在菜单界面输入字符“z”并按下回车键，即可使用该功能。程序将提示用户依次输入学生的疫苗接种次数序号、接种时间、接种地点、疫苗型号。若用户输入的数据不规范，则程序会强制要求用户重新输入，直到输入的数据符合程序限制为止。

2.2.13 添加离校申请统计

在菜单界面输入字符“w”并按下回车键，即可使用该功能。程序将提示用户依次输入学生的离校申请次数序号、离校原因、出校时间、返校时间、行程轨迹起点、行程轨迹终点、审核状态。若用户输入的数据不规范，则程序会强制要求用户重新输入，直到输入的数据符合程序限制为止。

2.2.14 清空内存和文件中的所有数据

在菜单界面输入字符“*”并按下回车键，即可使用该功能。在清空之前，程序再次询问用户是否确认删除，以防止用户误操作。若用户确认删除，则程序会清空文件和内存中存储的所有数据。

2.2.15 安全退出系统

在菜单界面输入字符“#”并按下回车键，即可关闭该系统。

2.2.16 以核酸检测为依据查看统计信息

在菜单界面输入字符“a”并按下回车键，即可使用该功能。该程序会尝试将目前文件中储存的所有学生的核酸检测次数进行统计，并以横向柱状统计图的形式直观反映各

个核酸检测次数所占据的比例。

2.2.17 以疫苗接种为依据查看统计信息

在菜单界面输入字符“b”并按下回车键，即可使用该功能。该程序会尝试将目前文件中储存的所有学生的疫苗接种次数进行统计，并以横向柱状统计图的形式直观反映各个疫苗接种次数所占据的比例。

2.2.18 以离校申请为依据查看统计信息

在菜单界面输入字符“c”并按下回车键，即可使用该功能。该程序会尝试将目前文件中储存的所有学生的离校申请次数进行统计，并以横向柱状统计图的形式直观反映各个离校申请次数所占据的比例。

2.2.19 修改密码

在菜单界面输入字符“m”并按下回车键，即可使用该功能。该程序会提示用户输入新密码，并将新密码以文本文件的形式储存在同级目录中。

2.2.20 系统空间占用情况

在菜单界面输入字符“n”并按下回车键，即可使用该功能。该程序将会显示用于存储学生疫情防控信息的文本文件，即“data.txt”的占用空间，单位为 kb。

2.2.21 作者信息

在菜单界面输入字符“r”并按下回车键，程序将显示当前系统的版本号和作者信息。

三、 程序设计与实现

1. 程序结构

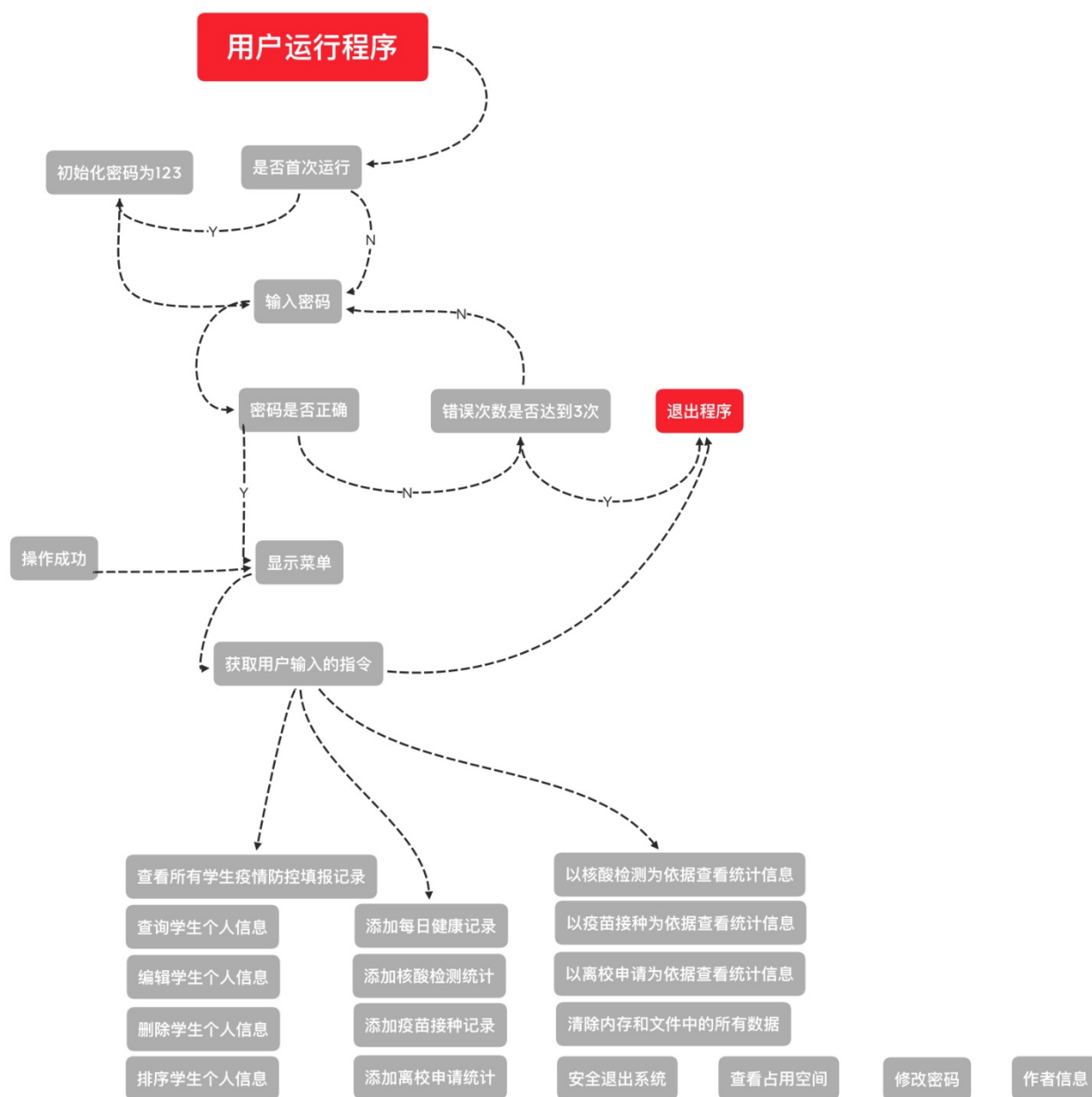


图 1. 本系统整体运行逻辑概览

2. 数据描述

本程序使用的全局变量有三类。一是结构体变量数组 `normalstudent[]`，用于存储学生信息；二是 `char` 型数组 `ProtectStr`，用作缓冲字符串，临时储存数据；三是 `StudentCheckData`，用于存储核酸检测次数的数据。

它们的定义如下代码所示。

//定义存储学生信息的结构体

struct NormalStudent

{

 char ID[IDNUMBER];//ID

 char name[NAMECHARACTER];//姓名

 int tel;//电话

 char email[EMAILCHARACTER];//邮箱

 char depart[DEPARTCHARACTER];//院系

 char clas[CLASCHARACTER];//班级

 char location[LOCATIONCHARACTER];//通讯地址

 char content[CONTENTCHARACTER];//简介

 int healthyyear;//健康填报年份

 int healthmonth;//健康填报月份

 int healthday;//健康填报日期

 float temp1;//体温(早)

 float temp2;//体温(中)

 float temp3;//体温(晚)

 char coughjudge[JUDGE];//是否咳嗽

 char feverjudge[JUDGE];//是否发烧

 int checkround;//核酸检测次数

 int checkyear;//核酸检测年份

 int checkmonth;//核酸检测月份

 int checkday;//核酸检测日期

 char checklocation[CHECKLOCATIONCHARACTER];//检测地点

 char checkjudge[CHECKJUDGE];//检测结果

 int vaccineround;//接种次数序号


```

int vaccineyear;//疫苗接种年份
int vaccinemonth;//疫苗接种月份
int vaccineday;//疫苗接种日期
char vaccinelocation[VACCINELOCATIONCHARACTER];//接种地点
char vaccinetype[VACCINETYPECHARACTER];//疫苗型号

int offround;//离校申请次数
char offreason[OFFREASONCHARACTER];//离校原因
int offyear;//出校年份
int offmonth;//出校月份
int offday;//出校日期
int backyear;//返校年份
int backmonth;//返校月份
int backday;//返校日期
char startingpoint[STARTINGPOINTCHARACTER];//行程轨迹起点
char destination[DESTINATIONCHARACTER];//行程轨迹终点
char supervision[SUPERVISIONCHARACTER];//审核状态
char about[ABOUTCHARACTER];//审核意见
}normalstudent[STUDENTNUMBER];

char ProtectStr[PROTECTSTRINGLENGTH];//缓冲区字符串
int StudentCheckData[STUDENTNUMBER][2]={0};//用于统计核酸检测次数分布的表格

```

3. 功能模块设计

为实现上述功能，系统实现自定义函数 52 个。具体如下：

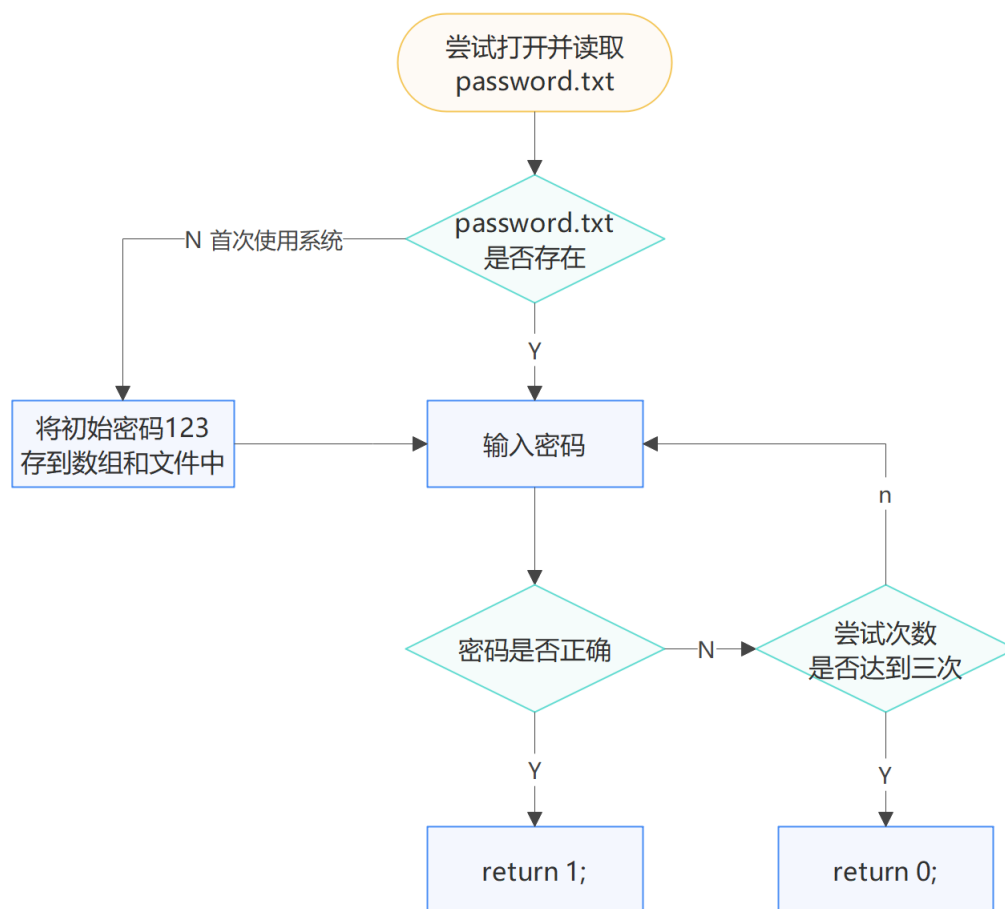
(1) 函数名：int login(char correctpassword[PASSWORDDIGIT], FILE *fp1)

函数功能：进行密码的检验与首次登陆时密码的初始化。

入口参数：char correctpassword[PASSWORDDIGIT]，用于储存正确的密码；FILE *fp1，用于读取文本文件中存储的密码。

出口参数：1 表示登陆成功，2 代表登陆失败。

算法描述：



(2) 函数名: `CountStudentIn(struct NormalStudent ns[STUDENTNUMBER])`

函数功能: 计算当前程序的结构体数组中存储了多少有效的学生信息

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址

出口参数: `i`, 表示校友的数量

算法描述: 使用循环算法, 对结构体数组的 ID 数组进行遍历, 每找到一个非空数组就执行 `i++` 语句, 直到遇到 ID 空数组为止。

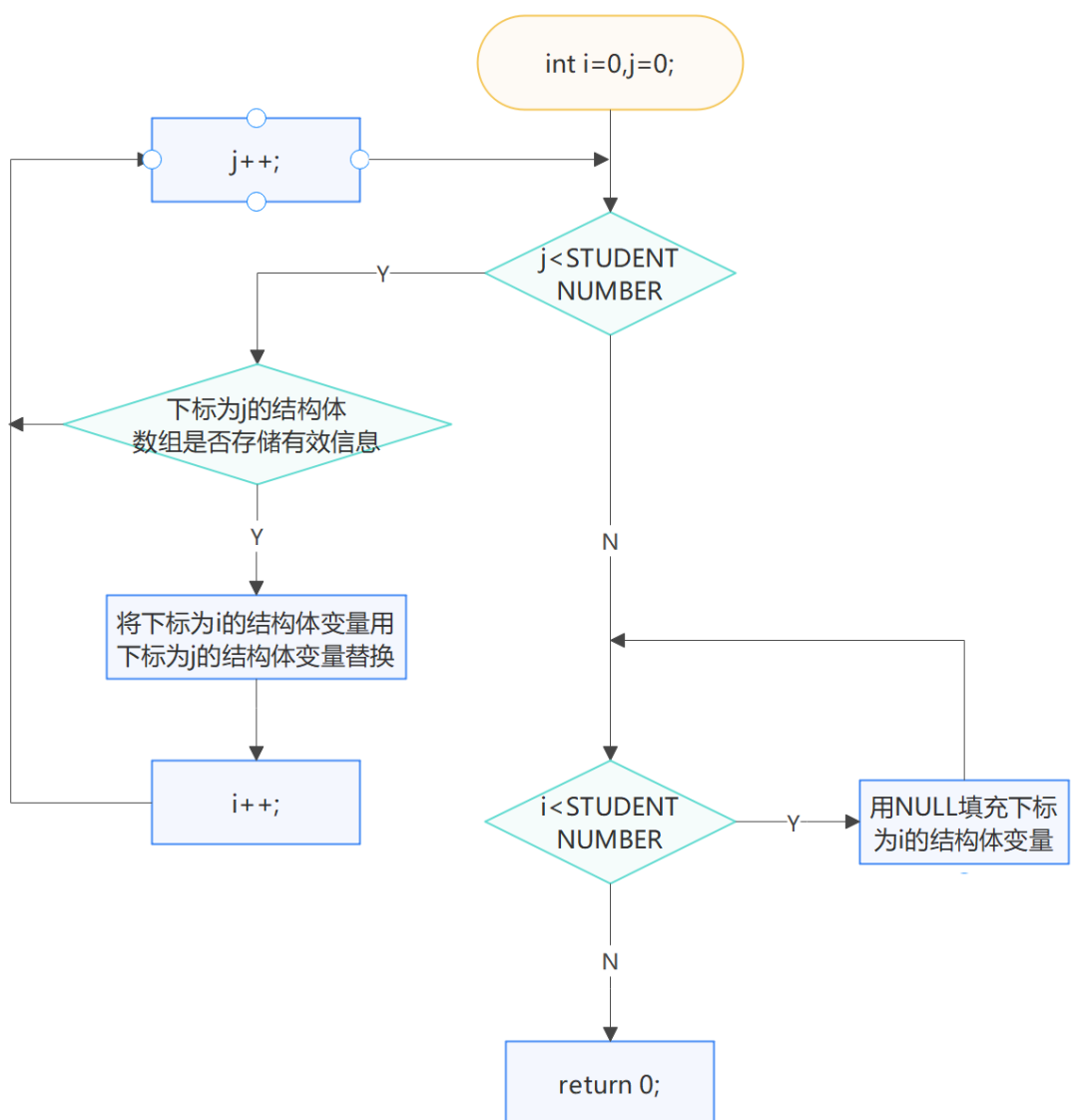
(3) 函数名: `int ClearNullStudent(struct NormalStudent ns[STUDENTNUMBER])`

函数功能: 将结构体数组中存储的有效校友信息向下标较小的方向移动

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址

出口参数: 0

算法描述: 类似于“删除字符串中指定的字符”, 只不过是把 `char` 型变量换成结构体变量, 把 `char[]` 数组换成结构体数组。



(4) 函数名: `int ReadData(struct NormalStudent ns[STUDENTNUMBER], FILE *fp)`

函数功能: 从 data.txt 中读取数据并保存至结构体变量中

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `FILE *fp`, 目标文件 data.txt 的指针

出口参数: 0

算法描述: 若文件不存在, 则提示用户读取错误; 若文件存在, 则逐行读取数据

(5) 函数名: `int SaveData(struct NormalStudent ns[STUDENTNUMBER], FILE *fp)`

函数功能: 将结构体变量中的数据保存至目标文件 data.txt

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; FILE *fp, 目标文件 data.txt 的指针

出口参数: 0

算法描述: 将结构体变量数组中的数据按下标递增的方式向目标文件 data.txt 中逐行写入。

(6) 函数名: int FunctionViewAll(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将结构体数组中存储的有效学生疫情防控填报数据(包含 ID、姓名、电话、邮箱、院系、班级、通讯地址, 每日健康填报日期、体温(早)、体温(中)、体温(晚)、是否咳嗽、是否发烧, 核酸检测日期、检测地点、检测结果, 疫苗接种次数序号、接种时间、接种地点、疫苗型号, 离校申请次数序号、离校原因、出校时间、返校时间、行程轨迹起点、行程轨迹终点、审核状态)逐行输出。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址

出口参数: 0

算法描述: 若结构体数组中没有有效学生疫情防控填报数据, 则提示用户“当前内存中没有存储学生信息!”; 反之逐行输出上信息。

(7) 函数名: int SearchStudentIndex(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 在结构体数组中寻找与用户输入的 ID 相匹配的学生疫情防控填报数据, 并按照 FunctionViewAll()的格式输出学生疫情防控填报数据。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址

出口参数: 0

算法描述: 若结构体数组中没有相匹配的 ID, 则提示用户“很抱歉, 您查找的 ID 不在该数据库中”; 反之则调用 FunctionViewAll()函数, 逐行输出学生疫情防控填报数据。

(8) 函数名: int InputID(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的 ID 存入下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 现将用户输入的 ID 存到缓冲字符串 ProtectStr 中, 然后用 strlen()函数检测用户输入的 ID 长度是否大于限制。若超过限制, 则提示用户重新输入, 直到长度符合要求为止。

(9) 函数名: int InputName(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的姓名存入下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 现将用户输入的姓名存到缓冲字符串 ProtectStr 中, 然后用 strlen()函数检测用户输入的姓名长度是否大于限制。若超过限制, 则提示用户重新输入, 直到长度符合要求为止。

(10) 函数名: int InputTel(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的电话存入下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 现将用户输入的电话存到下标为 i 的结构体变量中, 然后检测该整数是否过大或过小, 如果出现异常, 则提示用户重新输入, 知道电话符合要求为止。

(11) 函数名: int InputEmail(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的邮箱存入下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 现将用户输入的邮箱存到缓冲字符串 ProtectStr 中, 然后用 strlen()函数检测用户输入的邮箱长度是否大于限制。若超过限制, 则提示用户重新输入, 直到长度符合要求为止。

(12) 函数名: int InputDepart(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的院系存入下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述：现将用户输入的院系存到缓冲字符串 ProtectStr 中，然后用 strlen()函数检测用户输入的院系长度是否大于限制。若超过限制，则提示用户重新输入，直到长度符合要求为止。

(13) 函数名：int InputClas(struct NormalStudent ns[STUDENTNUMBER])

函数功能：将用户输入的班级存入下标为 i 的结构体变量中。

入口参数：struct NormalStudent ns[STUDENTNUMBER]，结构体数组首地址；int i，结构体变量下标数。

出口参数：0

算法描述：现将用户输入的班级存到缓冲字符串 ProtectStr 中，然后用 strlen()函数检测用户输入的班级长度是否大于限制。若超过限制，则提示用户重新输入，直到长度符合要求为止。

(14) 函数名：int InputLocation(struct NormalStudent ns[STUDENTNUMBER])

函数功能：将用户输入的通讯地址存入下标为 i 的结构体变量中。

入口参数：struct NormalStudent ns[STUDENTNUMBER]，结构体数组首地址；int i，结构体变量下标数。

出口参数：0

算法描述：现将用户输入的通讯地址存到缓冲字符串 ProtectStr 中，然后用 strlen()函数检测用户输入的通讯地址长度是否大于限制。若超过限制，则提示用户重新输入，直到长度符合要求为止。

(15) 函数名：int InputContent(struct NormalStudent ns[STUDENTNUMBER])

函数功能：将用户输入的简介存入下标为 i 的结构体变量中。

入口参数：struct NormalStudent ns[STUDENTNUMBER]，结构体数组首地址；int i，结构体变量下标数。

出口参数：0

算法描述：现将用户输入的简介存到缓冲字符串 ProtectStr 中，然后用 strlen()函数检测用户输入的简介长度是否大于限制。若超过限制，则提示用户重新输入，直到长度符合要求为止。

(16) 函数名：int AddStudent(struct NormalStudent ns[STUDENTNUMBER], int i)

函数功能：将用户输入的学生疫情防控信息输入到下标为 i 的结构体变量中。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 依次调用 InputID(), InputName(), InputTel(), InputEmail(), InputDepart(), InputClas(), InputLocation(), InputContent()函数。

(17) 函数名: int FunctionAdd(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 将用户输入的学生疫情防控填报信息输入到下标最小的空白结构体变量元素中

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址。

出口参数: 0

算法描述: 若结构体数组中的所有变量都已被填写有效学生疫情防控填报信息, 则提示用户“空间已满”, 拒绝用户输入, 反之则调用 CountStudentIn()获取变量 i, 再将变量 i 通过值传递的方式调用 AddStudent()函数。

(18) 函数名: int FunctionEdit(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 根据用户输入的 ID 编辑该学生的疫情防控填报信息。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址。

出口参数: 0

算法描述: 调用 FunctionSearch()寻找目标学生的下标 i, 再根据用户的要求有选择性地调用 InputID(), InputName(), InputTel(), InputEmail(), InputDepart(), InputClas(), InputLocation(), InputContent()中的一个或多个函数。

(19) 函数名: int FunctionDelete(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 根据用户输入的姓名删除该学生的疫情防控填报信息。

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址。

出口参数: 0

算法描述: 调用 FunctionSearch()寻找目标学生的下标 i, 再清空该结构体变量的数据, 最后用 ClearNullStudent()对结构体数组进行排序。

(20) 函数名: int DescendingIDSort(struct NormalStudent ns[STUDENTNUMBER])

函数功能: 在结构体数组中对学生 ID 进行降序排序

入口参数: struct NormalStudent ns[STUDENTNUMBER], 结构体数组首地址; int i, 结构体变量下标数。

出口参数: 0

算法描述: 选择排序。

(21) 函数名: `int AscendingIDSort(struct NormalStudent ns[STUDENTNUMBER])`

函数功能: 在结构体数组中对学生 ID 进行升序排序

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `int i`, 结构体变量下标数。

出口参数: 0

算法描述: 选择排序。

(22) 函数名: `int FunctionIDSort(struct NormalStudent ns[STUDENTNUMBER])`

函数功能: 让用户选择排序的方式 (升序/降序)

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `int i`, 结构体变量下标数。

出口参数: 0

算法描述: 分支结构。

(23) 函数名: `int EmptyFile(FILE *fp)`

函数功能: 清空 data.txt 中的内容。

入口参数: `FILE *fp`, 指向 data.txt 的文件指针。

出口参数: 0

算法描述: 利用 `fopen("data.txt","w")` 清空文件内容的特性。

(24) 函数名: `int Empty(struct NormalStudent ns[STUDENTNUMBER])`

函数功能: 清空结构体数组中的内容。

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址。

出口参数: 0

算法描述: 逐个遍历, 用 NULL 填充结构体数组中的所有变量。

(25) 函数名: `int FunctionEmpty(struct NormalStudent ns[STUDENTNUMBER], FILE *fp)`

函数功能: 让用户决定是否清空所有数据。

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `FILE *fp`, 指向 data.txt 的文件指针。

出口参数: 0

算法描述：最后向用户确认一次是否删除所有数据。若用户确认，则调用 EmptyFile()和 Empty()。若用户取消，则返回菜单界面。

(26) 函数名：int PrintDetail(struct NormalStudent ns[STUDENTNUMBER], int i)

函数功能：按规范格式逐行输出下标为 i 的学生的疫情防控填报信息。

入口参数：struct NormalStudent ns[STUDENTNUMBER]，结构体数组首地址；int i，要输出所有疫情防控填报信息的学生的下标。

出口参数：0

算法描述：用制表符以表格形式输出下标为 i 的学生的疫情防控填报信息。

(27) 函数名：int FunctionShowDetail(struct NormalStudent ns[STUDENTNUMBER])

函数功能：按规范格式逐行输出符合用户输入学生的疫情防控填报信息。

入口参数：struct NormalStudent ns[STUDENTNUMBER]，结构体数组首地址。

出口参数：0

算法描述：依次调用 FunctionSearch()和 PrintDetail()函数，用制表符以表格形式输出符合用户输入 ID 的所有信息。若该 ID 不存在于数据库中，则提示用户“未找到该学生”，并返回菜单界面。

(28) 函数名：int FunctionShutdown(FILE *fp)

函数功能：关闭系统

入口参数：FILE *fp，指向 data.txt 的文件指针。

出口参数：0

算法描述：利用 fclose()取消文件指针 fp 与 data.txt 的关联，再调用 exit()关闭程序。

(29) 函数名：int GetCheckRoundnMax(int sc[STUDENTNUMBER][2], int TotalCheckItem)

函数功能：返回 sc 数组中所占人数最多的核酸检测次数的人数。

入口参数：int sc[STUDENTNUMBER][2]，存储核酸检测次数统计数据的首地址；int TotalCheckItem，sc 数组中一共有多少个不同的核酸检测次数。

出口参数：max

算法描述：对 sc 数组遍历前 TotalCheckItem 项，实时更新最大人数 max。遍历完毕后返回 max 的值。

(30) 函数名：int ResetPreviousCheck(int sc[STUDENTNUMBER][2])

函数功能：重置 sc 数组中的数据至初始状态。

入口参数: `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址

出口参数: 0

算法描述: 对 `sc` 数组遍历, 将每一项都用 `NULL` 填充。

(31) 函数名: `int BlankIndexForPreviousCheck(int sc[STUDENTNUMBER][2])`

函数功能: 返回 `sc` 数组中所有空余位置的下标的最小值。

入口参数: `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址

出口参数: 0

算法描述: 对 `sc` 数组遍历, 每遍历一项就让局部变量 `i++`, 直到 `sc[i][0]==0` 为止。

(32) 函数名: `int IsPreviousCheck(int sc[STUDENTNUMBER][2], int y)`

函数功能: 判断核酸检测次数 `y` 是否已经被数组 `sc` 保存

入口参数: `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址;
`y`, 待判断的年份

出口参数: `i`, `i==1` 时说明存在, `i==0` 时说明不存在

算法描述: 初始时令 `i=0`, 对 `sc` 的前 `BlankIndexForPreviousCheck()` 进行遍历, 若发现有一项相同, 则让 `i=1`。遍历结束后, 返回整数 `i` 的值。

(33) 函数名: `int PreviousCheckIndex(int sc[STUDENTNUMBER][2], int y)`

函数功能: 获取当前核酸检测次数在数组 `sc` 中的位置

入口参数: `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址;
`int y`, 待查找的核酸检测次数

出口参数: `i`

算法描述: 对 `sc` 数组遍历, 每遍历一项就让局部变量 `i++`, 直到 `sc[i][0]==y` 为止。

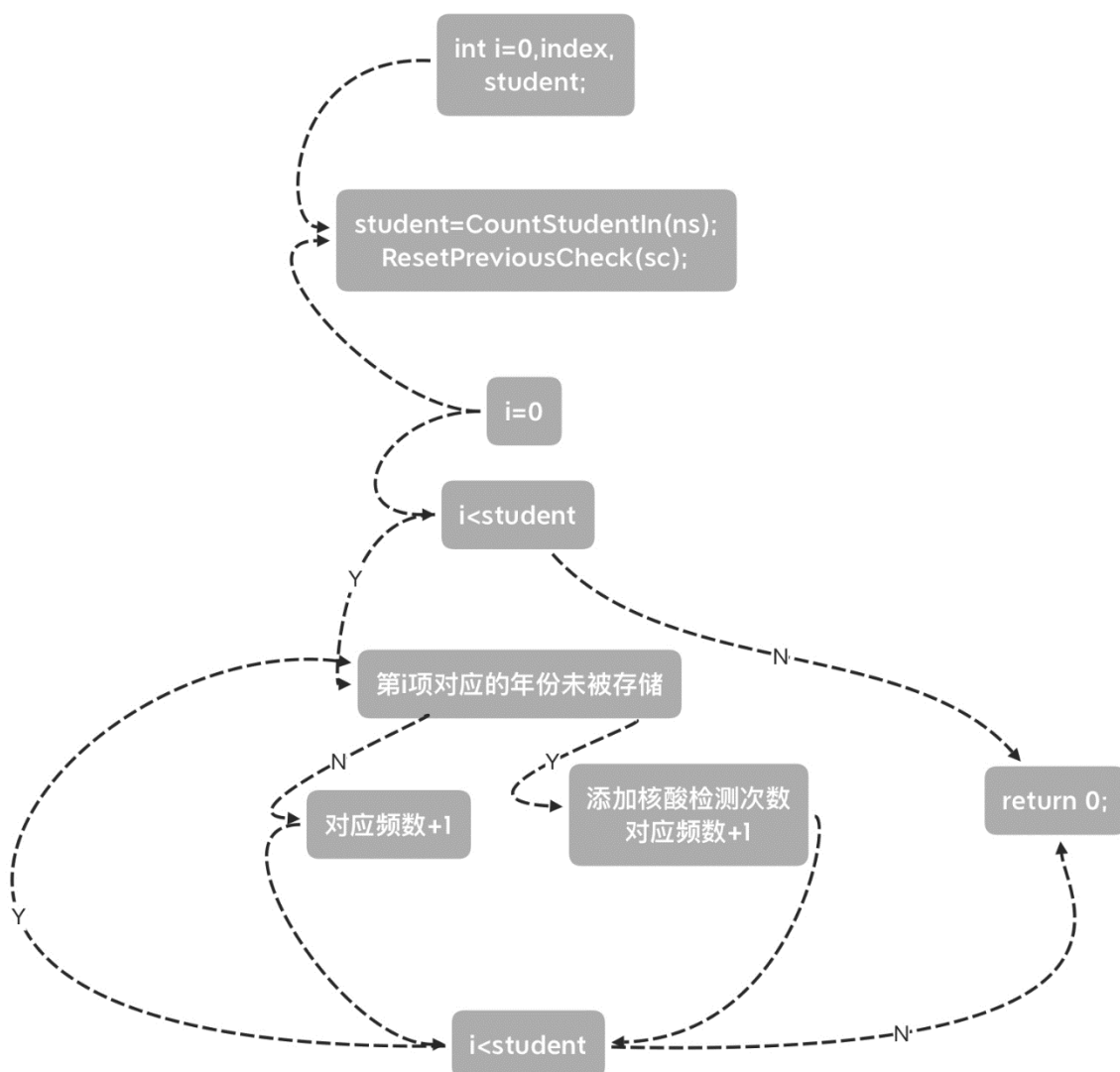
(34) 函数名: `int CountStudentCheck(struct NormalStudent ns[STUDENTNUMBER], int sc [STUDENTNUMBER][2])`

函数功能: 对核酸检测次数信息进行更新

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址

出口参数: 0

算法描述:



(35) 函数名: `int SortStudentCheck(int sc[STUDENTNUMBER][2])`

函数功能: 对存储核酸检测次数统计信息的数组按照人数降序排序

入口参数: `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址

出口参数: 0

算法描述: 选择排序

(36) 函数名: `int ShowCheckBarChart (struct NormalStudent ns[STUDENTNUMBER], int sc[STUDENTNUMBER][2])`

函数功能: 进行百分比的运算, 并按照特定比例转换。

入口参数: `struct NormalStudent ns[STUDENTNUMBER]`, 结构体数组首地址; `int sy[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址。

出口参数: 0

算法描述: 将所有百分比最大的一项乘以某个倍数变成 100%, 其他百分比也乘以该倍数, 最后将转换后的百分比作为实参。

(37) 函数名: `int FunctionShowStudentCheckStatics(struct NormalStudent ns [STUDENTNUMBER], int sc[STUDENTNUMBER][2])`

函数功能: 以柱状统计图的形式直观展示各核酸检测次数所占百分比。

入口参数: `struct NormalStudent ns [STUDENTNUMBER]`, 结构体数组首地址; `int sc[STUDENTNUMBER][2]`, 存储核酸检测次数统计数据的数组的首地址。

出口参数: 0

算法描述: 依次调用 `CountStudentCheck()`, `SortCheckStudent(sc)`, `ShowCheckBarChart(ns,sc)`

(38) 函数名: `int CountVaccineRound(struct NormalStudent ns [STUDENTNUMBER], int sc[STUDENTNUMBER][2])`

函数功能: 对疫苗接种次数进行更新

入口参数: `struct NormalStudent ns [STUDENTNUMBER]`, 结构体数组首地址; `int sc[STUDENTNUMBER][2]`, 存储疫苗接种次数统计数据的数组的首地址。

出口参数: 0

算法描述: 若疫苗接种次数存在, 则让对应频数++; 若疫苗接种次数不存在, 则调用 `BlankIndexForPreviousCheck()` 添加疫苗接种次数, 再让对应频数++。

(39) 函数名: `int FunctionShowVaccineRoundStatics(struct NormalStudent ns [STUDENTNUMBER], int sc[STUDENTNUMBER][2])`

函数功能: 以柱状统计图的形式直观展示各疫苗接种次数所占百分比。

入口参数: `struct NormalStudent ns [STUDENTNUMBER]`, 结构体数组首地址; `int sc[STUDENTNUMBER][2]`, 存储疫苗接种次数统计数据的数组的首地址。

出口参数: 0

算法描述: 依次调用 `CountStudentCheck()`, `SortCheckStudent(sc)`, `ShowCheckBarChart(ns,sc)`

(40) 函数名: `int FunctionChangePassword()`

函数功能: 更改密码

入口参数: 无

出口参数: 0

算法描述: 先调用 EmptyFile(), 清空 password.txt 中的内容, 再提示用户输入新密码, 并存储到 password.txt 中。

(41) 函数名: int FunctionGetOccupiedSpace(FILE *fp)

函数功能: 显示 data.txt 占用的空间

入口参数: FILE *fp, 指向 data.txt 的文件指针

出口参数: 0

算法描述: 使用 fseek()将文件光标指针移动到文件末端, 再用 ftell()函数返回总字节数, 将其除以 1024, 得到以 kb 为单位的大小, 最后用 rewind()函数对文件光标指针复位。

(42) 函数名: int FunctionShowAuthor()

函数功能: 展示软件版本号和作者信息

入口参数: 无

出口参数: 0

算法描述: 无算法

(43) 函数名: int PrintWelcome()

函数功能: 输出欢迎界面

入口参数: 无

出口参数: 0

算法描述: 无算法

(44) 函数名: int PrintMenuHead()

函数功能: 输出菜单上部分

入口参数: 无

出口参数: 0

算法描述: 无算法

(45) 函数名: int PrintMenuEdit()

函数功能: 输出菜单下部分的编辑栏

入口参数: 无

出口参数: 0

算法描述: 无算法

(46) 函数名: `int PrintMenuStatics()`

函数功能: 输出菜单下部分的统计数据栏

入口参数: 无

出口参数: 0

算法描述: 无算法

(47) 函数名: `int PrintMenuAccount()`

函数功能: 输出菜单下部分的关于栏

入口参数: 无

出口参数: 0

算法描述: 无算法

(48) 函数名: `int PrintFileCorrupted()`

函数功能: 输出文件损坏提示

入口参数: 无

出口参数: 0

算法描述: 无算法

(49) 函数名: `int ShowTab(short int *tab)`

函数功能: 负责顶部栏项目的显示

入口参数: 无

出口参数: 0

算法描述: 无算法

(50) 函数名: `int SwitchTab(short int *tab)`

函数功能: 负责顶部栏项目指数 `tab` 的切换

入口参数: 无

出口参数: 0

算法描述: 无算法

(51) 函数名: `int PrintShutDown()`

函数功能: 显示退出界面

入口参数: 无

出口参数: 0

算法描述：无算法

(52) 函数名：int PrintAuthor()

函数功能：输出软件版本号和作者信息

入口参数：无

出口参数：0

算法描述：无算法

四、设计难点与解决方法

4.1 表格设计

控制台是以纯文本的形式显示的，要在纯文本中显示表格，最好的方式是使用制表符。为此，我将代码的编码的格式改成了 GB2312（简体中文字符集），依次来兼容制表符字符。

现在还有一个问题：控制制表符输出的位置。经过精确计算，控制每一行的数量为 39 个字，之后输出一个制表符。若该行没有输出 39 个字就达到了 char 数组的'\0'处，就用表格将剩下的空白补上。

```
int PrintContent(struct NormalStudent ns[STUDENTNUMBER],int n){
    int i,j,k;//i 为总字数，j 为当前行中的总字数，k 用于补齐空格
    printf("\n | ");
    for(i=0;ns[n].content[i]!='\0';)
    {
        for(j=0;j<39&&ns[n].content[i]!='\0';i++,j++)
        {
            printf("%c",ns[n].content[i]);
        }
        if(ns[n].content[i]=='\0')
        {
            for(k=0;k<(39-j)*2+1;k++)
            {printf(" ");}
            printf(" | \n");
            break;
        }
        else
        {
```

```

        printf(" | \n | ");
    }
}
printf(" L");for(i=0;i<39*2+1;i++){printf("—");}printf("L ");
return 0;
}

```

五、 运行效果与不足之处

5.1 运行效果

以下是程序运行的部分界面。

大连理工大学疫情防控管理系统	
菜单	
编辑	统计数据
1 查看所有学生疫情防控填报记录	关于
2 查询学生个人信息	
3 编辑学生个人信息	
4 删除学生个人信息	
5 排序学生个人信息	
6 添加学生个人信息	
x 添加每日健康记录	
y 添加核酸检测统计	
z 添加疫苗接种信息	
w 添加离校申请统计	
* 清空内存和文件中的所有数据	
# 安全退出系统	

提示：菜单中每一项功能前面的字符是它的代号，“T”或“t”用于切换菜单的不同栏目。请输入操作对应的编号：

请输入操作对应的编号:1

-----学生个人信息记录-----						
ID	姓名	电话	邮箱	院系	班级	通讯地址
202101	白岩浩	1382182053	2062400572@qq.com	电信	电计2004	大连理工大学
202102	李明	1351285780	2058345178@163.com	能动	动力2104	大连理工大学
202103	单珊	1367206551	2871290562@foxmail.com	建艺	建筑2102	大连理工大学
202104	王新	1351249510	3782890452@126.com	土木	智建2103	大连理工大学
202105	陶然	1872253855	3761289471@dlut.com	软件	国软2103	大连理工大学

校园疫情防控信息管理系统

====每日健康填报记录=====						
ID	日期	体温(早)	体温(中)	体温(晚)	是否咳嗽	是否发烧
202101	2022/1/10	36.0	36.2	36	否	否
202102	2022/1/10	36.1	36.2	36	否	否
202103	2022/1/10	36.3	36.3	36	否	否
202104	2022/1/10	36.1	36.2	36	否	否
202105	2022/1/10	36.2	36.2	36	否	否

====核酸检测填报记录=====				
ID	日期	检测次数	检测地点	检测结果
202101	2022/1/08	5	大连理工大学	阴性
202102	2021/12/6	4	大连理工大学	阴性
202103	2022/1/08	5	大连理工大学	阴性
202104	2021/12/7	3	大连理工大学	阴性
202105	2022/1/08	5	大连理工大学	阴性

====疫苗接种填报记录=====				
ID	日期	接种序号	检测地点	检测结果
202101	2021/12/3	3	大连理工大学	北京科兴
202102	2021/5/17	2	大连理工大学	北京科兴
202103	2021/12/3	3	大连理工大学	北京科兴
202104	2021/12/3	3	大连理工大学	国药中生
202105	2021/5/28	2	大连理工大学	国药中生

====离校申请填报记录=====								
ID	出校日期	返校日期	申请次数	行程轨迹起点	行程轨迹终点	离校原因	审核状态	
202101	2021/12/2	2021/12/2	1	大连理工大学	黑石礁	购买书籍	同意	
202102	2021/12/8	2022/1/01	2	大连理工大学	医大二院	前往医院	同意	
202103	2021/12/8	2021/12/9	2	大连理工大学	海韵园	返回家中	同意	
202104	2021/12/9	2021/12/9	2	大连理工大学	海韵园	返回家中	同意	
202105	2021/12/7	2021/12/7	2	大连理工大学	黑石礁	购买书籍	同意	

按下回车键返回菜单.

请输入操作对应的编号:2

请输入您想查找的学生信息ID (例如 202101): 202102
搜寻到相关结果!

====每日健康填报记录=====						
ID	日期	体温(早)	体温(中)	体温(晚)	是否咳嗽	是否发烧
202102	2022/1/10	36.1	36.2	36	否	否

====核酸检测填报记录=====				
ID	日期	检测次数	检测地点	检测结果
202102	2021/12/6	4	大连理工大学	阴性

====疫苗接种填报记录=====				
ID	日期	接种序号	检测地点	检测结果
202102	2021/5/17	2	大连理工大学	北京科兴

====离校申请填报记录=====								
ID	出校日期	返校日期	申请次数	行程轨迹起点	行程轨迹终点	离校原因	审核状态	
202102	2021/12/8	2022/1/01	2	大连理工大学	医大二院	前往医院	同意	

该生体温不存在异常!
该生不存在咳嗽症状!
该生不存在发烧症状!

按下回车键以返回菜单.

请输入操作对应的编号:3

请输入您要编辑的学生的ID: 202103

请选择您要编辑【202103】的部分:

(1. ID 2. 姓名 3. 电话 4. 邮箱 5. 院系 6. 班级 7. 通讯地址 8. 简介 9. 退出编辑)

您的操作是(输入1-9之间的数字): 3

请输入电话: 1382182053

编辑成功!

请输入操作对应的编号:4

请输入您要删除的填报记录的学生ID: 202104

删除成功! 按下回车键以返回菜单. _

请输入操作对应的编号:5

请选择学生ID排序方式对应的数字(1. 降序排列 2. 升序排列): 1

=====每日健康填报记录=====						
ID	日期	体温(早)	体温(中)	体温(晚)	是否咳嗽	是否发烧
202105	2022/1/10	36.0	36.2	36	否	否
202104	2022/1/10	36.1	36.2	36	否	否
202103	2022/1/10	36.3	36.3	36	否	否
202102	2022/1/10	36.1	36.2	36	否	否
202101	2022/1/10	36.2	36.2	36	否	否

=====核酸检测填报记录=====				
ID	日期	检测次数	检测地点	检测结果
202105	2022/1/08	5	大连理工大学	阴性
202104	2021/12/6	4	大连理工大学	阴性
202103	2022/1/08	5	大连理工大学	阴性
202102	2021/12/7	3	大连理工大学	阴性
202101	2022/1/08	5	大连理工大学	阴性

=====疫苗接种填报记录=====				
ID	日期	接种序号	检测地点	检测结果
202105	2021/12/3	3	大连理工大学	北京科兴
202104	2021/5/17	2	大连理工大学	北京科兴
202103	2021/12/3	3	大连理工大学	北京科兴
202102	2021/12/3	3	大连理工大学	国药中生
202101	2021/5/28	2	大连理工大学	国药中生

=====离校申请填报记录=====							
ID	出校日期	返校日期	申请次数	行程轨迹起点	行程轨迹终点	离校原因	审核状态
202105	2021/12/2	2021/12/2	1	大连理工大学	黑石礁	购买书籍	同意
202104	2021/12/8	2022/1/01	2	大连理工大学	医大二院	前往医院	同意
202103	2021/12/8	2021/12/9	2	大连理工大学	海韵园	返回家中	同意
202102	2021/12/9	2021/12/9	2	大连理工大学	海韵园	返回家中	同意
202101	2021/12/7	2021/12/7	2	大连理工大学	黑石礁	购买书籍	同意

排序成功! 按下回车键以返回菜单.

大连理工大学疫情防控管理系统

菜单

编辑
■■■统计数据■■■
关于

a 以核酸检测为依据查看统计信息
b 以疫苗接种为依据查看统计信息
c 以离校申请为依据查看统计信息

提示：菜单中每一项功能前面的字符是它的代号，“T”或“t”用于切换菜单的不同栏目。
 请输入操作对应的编号：a

```
=====
次数  人数  百分比  柱状统计图
  5     3  60.00%  ██████████
  4     1  20.00%  █████
  3     1  20.00%  █████
=====
```

按下回车键以返回菜单. _

请输入操作对应的编号：b

```
=====
次数  人数  百分比  柱状统计图
  3     3  60.00%  ██████████
  2     2  40.00%  ████████
=====
```

按下回车键以返回菜单. _

请输入操作对应的编号：c

```
=====
次数  人数  百分比  柱状统计图
  2     4  80.00%  ██████████████
  1     1  20.00%  █████
=====
```

按下回车键以返回菜单.

5.2 不足之处

在编写程序的过程中，对于界面的美观设计是我一直尝试想办法却难以解决的问题，如实现密钥输入、鼠标点击菜单栏等等，这仍需我不断学习 C 语言程序设计相关知识，多向老师、学长请教！

六、 课程设计体会与收获

学习 C 语言程序设计的一个学期实在让人印象深刻。在这一个学期里，尤其是着手开始做程序设计的这两周，我的每一天都是无比充实的。也正因有了最后的期末考核和每次的程序设计作业，我的编程之路才算一路走了下来。

虽然是电子信息类的学生，课前自学的难度确实令我有些吃不消。但我深知在大学，除了自己，没人会有多在乎你学得怎么样，所以我必须对自己负责，有了问题也一定要尽量解决。性格内向的我一直不是很喜欢向老师提问，但在学习这门程序设计课之后，我发现问老师的确是解决速度最快的方法，自己查资料要走很多弯路。而我每次在课间向汪老师询问基础问题时，老师总是耐心地向我解答问题，大工老师的有求必应、传承奉献的精神确实令我感动。

虽然程序设计的课程算不上多苦多累，但这中间我确实产生过疲倦的念头。因为想要扎实掌握一门编程语言的应用操作、程序设计，确实需要消耗大量时间。但我深知，学习一门编程语言需要持久的毅力和自控力，方才能学有所得、学有所成。抱着这些想法，我一次又一次认真完成老师布置的各项任务。

这次的程序设计大作业，确实对我是一个质的提升。当我看到自己最后敲出的代码作品时，我由衷感到这一路的坚持和付出终是付有回报。在大学，学习固然是要靠自己，但是能每天和志同道合的同学在一起交流程序设计，感受到自己和别人的差距，也不失一种对自己的鞭策。

可能以后回顾这段编程经历，觉得并没有什么了不起的，但对于现阶段的我，能做到今天这一步，尤其是当看到自己写的程序能跑起来、顺利地运行时，心中涌出的成就感让我懂得了一分耕耘一份收获的学习程序语言的快乐。

能够有今天的学习领悟，首先要感谢电信学部和我的 C 语言程序设计老师汪老师和助教老师滕飞学长，能够给我提供宝贵的机会系统、扎实、深入地学习程序设计基础，让我深刻体会到求真 务实、传承奉献的大工精神。同时也万分感谢一路以来同我商讨编程学习、程序设计的 同学们，虽然交流不多，但是每次共同讨论的问题都能得到不同方式的解决和新的领悟，让我感受到了来自同门的关怀和鼓舞。最后还要感谢我的两个室友，我们相互鼓励、一起学习的劲头不断激励着我持续地学习程序设计、完备自己的知识储能。并且共同分享的诸多程序语言学习网站、视频和书籍也让我们彼此互相进步。

最后祝愿电信学部能够越办越好！