

**《高级语言程序设计进阶》**

**综合设计实验报告**

|  |  |
| --- | --- |
| **姓 名** |  |
| **学 号** |  |
| **学 期** |  |
| **完成时间** |  |

二O二四 年四月

目录

[**一、实验目的3**](#_一、实验目的)

[**二、实验内容3**](#_二、实验内容)

[**三、系统设计**3](#_三、系统设计)

[1.系统功能框架4](#_1._系统功能框架)

[2. 数据结构设计4](#_2._数据结构设计)

[3. 函数说明4](#_3._函数说明)

[4. 完整的源代码4](#_4._完整的源代码)

[5.编译成功界面截图22](#_5.编译成功界面截图)

[6.运行界面22](#_6.运行界面)

[**(1)初始化执行**22](#_(1).初始化执行:)

[**(2)插入操作**22](#_(2).插入操作:)

[**(3)修改操作**23](#_(3).修改操作:)

[**(4)删除操作**23](#_(4).删除操作:)

**[四、总结](#_四、总结)**[24](#_四、总结)

[**五、操作说明**25](#_操作文档:)

\*目录的索引是为了方便查询，Ctrl加鼠标左键即可跳转相应页面

### 一、实验目的

本实验的目的在于通过设计一个学生信息管理系统，加深对数据结构与算法的理解，提高编程能力。

1、进一步掌握和利用C语言进行程序设计的能力；

2、进一步理解和运用结构化程序设计的思想和方法；

3、初步掌握开发一个小型实用系统的基本方法；

4、学会调试一个较长程序的基本方法；

5、学会利用流程图或N-S图表示算法；

6、掌握书写程序设计开发文档的能力。

### 二、实验内容

本实验设计了一个学生信息管理系统，具体功能包括：

**1.插入学生信息：**用户可以输入学生姓名、学号以及语文、数学、英语成绩，将学生信息插入系统中。

**2.查询学生信息：**用户可以根据学生姓名或学号查询学生信息，系统将显示相应学生的详细信息。

**3.删除学生信息：**用户可以根据学生姓名或学号删除学生信息。

**4.对学生信息进行排序：**用户可以按照语文、数学或英语成绩对学生信息进行排序，系统将按照用户选择的科目进行排序。

**5.按成绩范围查询学生信息：**用户可以输入一个成绩范围，系统将显示在该范围内的学生信息。

### 三、系统设计

## 1. 系统功能框架

该系统是一个学生信息管理系统，具有以下功能：

插入学生信息：可以插入学生的姓名、学号和各科成绩。

查询学生信息：可以根据姓名或学号查询学生的详细信息。

删除学生信息：可以根据姓名或学号删除学生的信息。

排序学生信息：可以按照语文、数学或英语成绩对学生信息进行排序。

按成绩范围查询：可以根据指定科目的成绩范围查询学生信息。

## 2. 数据结构设计

系统使用链表作为数据存储结构，其中：

**Student** 结构体表示一个学生，包含姓名、学号和各科成绩。

**Grade** 结构体表示学生的各科成绩。

**g\_head** 是链表的头指针，指向第一个学生节点。

## 3. 函数说明

**InsertStudent:** 插入学生信息到链表中。

**SearchStudent:** 查询学生信息并显示在输出框中。

**DeleteStudent:** 删除学生信息。

**Rewrite\_data:** 更新文件中的学生信息。

**Sort\_Student:** 对学生信息按照指定科目成绩进行排序。

**revise:** 检查是否有重复的姓名或学号。

**CompareStudent:** 比较两个学生的指定科目成绩，用于排序。

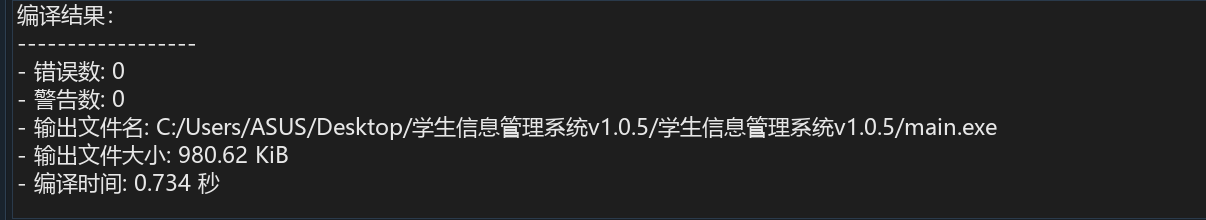
**Find\_range:** 根据指定科目和成绩范围查找学生信息。

**Range\_compare:** 比较学生的指定科目成绩是否在指定范围内。

## 4. 完整的源代码

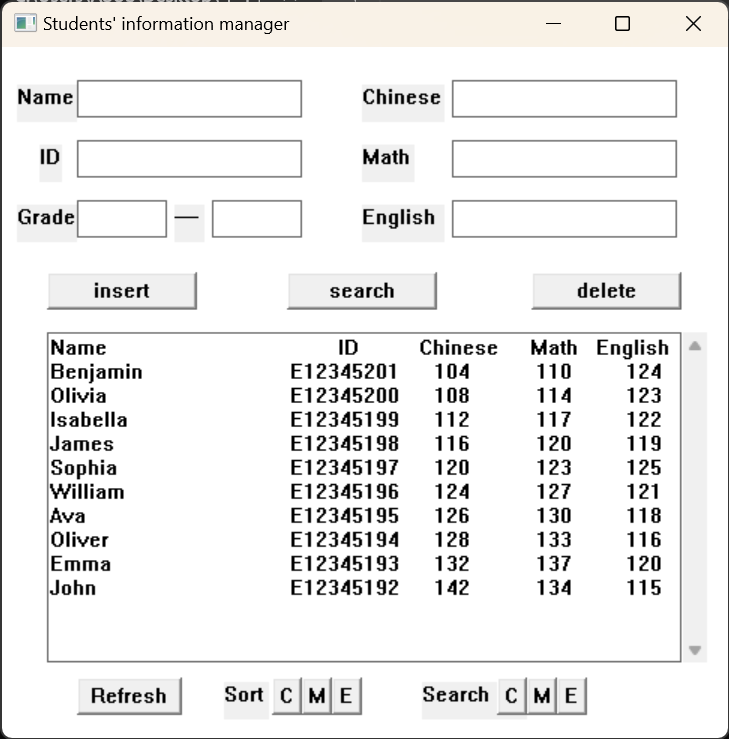
#include **<**windows**.**h**>**#include **<**stdio**.**h**>**#include **<**stdlib**.**h**>**#include **<**string**.**h**>***/\*整体思路是先读取文本里面的数据，然后处理数据用链表，每次数据处理完毕都写一次到数据里面，虽然有多余的擦除和写入，  
但是整体思路清晰，实现较容易\*/***struct** Grade **{**    **int** Chinese**;**    **int** Math**;**    **int** English**;  
};  
struct** Student **{**    **char** name**[**50**];**    **char** id**[**20**];**    **struct** Grade grade**;**    **struct** Student**\*** next**;  
};  
typedef** **struct** Student Student**;  
typedef** **struct** Grade Grade**;***//* *定义学生以及成绩结构体*FILE **\***student\_data**;***//定义文件的全局变量*HWND Name\_input**;**HWND ID\_input**;**HWND g\_hInsertButton**;**HWND g\_hSearchButton**;**HWND g\_hDeleteButton**;**HWND Output\_text**;**HWND Name\_lable**;**HWND ID\_lable**;**HWND Math\_lable**;**HWND Math\_input**;**HWND Chinese\_lable**;**HWND Chinese\_input**;**HWND English\_lable**;**HWND English\_input**;**HWND Grade\_input\_1**;**HWND Grade\_input\_2**;**HWND Grade\_lable\_1**;**HWND Grade\_lable\_2**;**HWND Sort\_lable**;**HWND Sort\_C**;**HWND Sort\_M**;**HWND Sort\_E**;**HWND Search\_lable**;**HWND Search\_C**;**HWND Search\_M**;**HWND Search\_E**;**HWND Refresh**;***//* *windows窗口要用的全局变量  
  
//* *这个是学生链表的头节点,本身不存储数据*Student**\*** g\_head **=** NULL**;***//* *这个是求某一范围成绩时单独调用的链表头节点*Student**\*** range\_list **=** NULL**;***//* *这个全局变量是为了实现每次显示6个节点,按next或者last来显示前或者后6个节点*Student**\*** current\_display\_node**;***//* *这个是每次显示的时候,要显示的那6个节点的头节点*Student**\*** display\_list**;**Student InsertStudent**(**Student temp\_student**,**Student **\***head**);  
void** SearchStudent**(char\*** name**,char\*** id**);  
void** DeleteStudent**(char\*** name**,char\*** id**);  
void** UpdateOutputBox**(**Student**\*** head**);  
int** Init\_linklist**(**Student **\***head**);  
void** Rewrite\_data**(**Student **\***head**);  
void** Sort\_Student**(**Student **\***head**,int** subject**);  
int** CompareStudent**(**Student**\*** a**,** Student**\*** b**,** **int** subject**);**Student**\*** Find\_range**(int** high**,int** low**,int** subject**,**Student**\*** head**,**Student**\*** range\_list**);  
int** Range\_compare**(**Student**\*** stu**,int** subject**,int** high**,int** low**);  
int** revise**(char\*** name**,char\*** id**,**Student**\*** head**);***//* *窗口过程函数*LRESULT CALLBACK WindowProc**(**HWND hwnd**,** UINT uMsg**,** WPARAM wParam**,** LPARAM lParam**)** **{**    **switch(**uMsg**)** **{**    **case** WM\_CREATE**:**        */\**        *窗口创建说明:*        *HWND变量* *=* *CreateWindow(*        *"控件类名"*  *EDIT,BUTTON等*        *"初始名称"*  *自定义*        *WS\_VISIBLE* *|* *WS\_CHILD* *|* *WS\_BORDER*     
        *（WS\_VISIBLE* *表示创建后立即显示该控件*        *WS\_CHILD* *表示该控件是子窗口的一部分*        *WS\_BORDER* *表示给窗口添加边框。）*        *50,* *50*  *控件左上角相对于父窗口左边框的水平距离（x* *坐标）和垂直距离（y* *坐标）。*        *150,* *25*  *这是控件的宽度和高度*        *hwnd*  *这是指定了父窗口的句柄，即将成为父窗口的子控件。*        *NULL：这是关于菜单的句柄，这里不涉及菜单，因此为* *NULL。*        *NULL：这是控件的实例句柄，这里不需要指定，所以为* *NULL。*        *NULL：这是传递给控件创建的其他参数，这里也不需要，所以为* *NULL。*        *\*/*          
        *//* *创建姓名输入框*        Name\_input **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 50**,** 22**,** 150**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建姓名输入框前的标签*        Name\_lable **=** CreateWindow**("STATIC",** **"Name",** WS\_VISIBLE **|** WS\_CHILD **,** 10**,** 25**,** 40**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建语文输入框*        Chinese\_input **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 300**,** 22**,** 150**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建语文输入框前的标签*        Chinese\_lable **=** CreateWindow**("STATIC",** **"Chinese",** WS\_VISIBLE **|** WS\_CHILD **,** 240**,** 25**,** 55**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建数学输入框*        Math\_input **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 300**,** 62**,** 150**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建数学输入框前的标签*        Math\_lable **=** CreateWindow**("STATIC",** **"Math",** WS\_VISIBLE **|** WS\_CHILD **,** 240**,** 65**,** 35**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建英语输入框*        English\_input **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 300**,** 102**,** 150**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建英语输入框前的标签*        English\_lable **=** CreateWindow**("STATIC",** **"English",** WS\_VISIBLE **|** WS\_CHILD **,** 240**,** 105**,** 55**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建学号输入框*        ID\_input **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 50**,** 62**,** 150**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建成绩输入框1*        Grade\_input\_1 **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 50**,** 102**,** 60**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建成绩输入框2*        Grade\_input\_2 **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER**,** 140**,** 102**,** 60**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建成绩输入框前的标签*        Grade\_lable\_1 **=** CreateWindow**("STATIC",** **"Grade",** WS\_VISIBLE **|** WS\_CHILD **,** 10**,** 105**,** 40**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建成绩输入框中间的标签*        Grade\_lable\_2 **=** CreateWindow**("STATIC",** **"----",** WS\_VISIBLE **|** WS\_CHILD **,** 115**,** 105**,** 20**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建学号输入框前的标签*        ID\_lable **=** CreateWindow**("STATIC",** **"ID",** WS\_VISIBLE **|** WS\_CHILD **,** 25**,** 65**,** 15**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建插入按钮*        g\_hInsertButton **=** CreateWindow**("BUTTON",** **"insert",** WS\_VISIBLE **|** WS\_CHILD**,** 30**,** 150**,** 100**,** 25**,** hwnd**,** **(**HMENU**)**1**,** NULL**,** NULL**);**        *//* *创建查询按钮*        g\_hSearchButton **=** CreateWindow**("BUTTON",** **"search",** WS\_VISIBLE **|** WS\_CHILD**,** 190**,** 150**,** 100**,** 25**,** hwnd**,** **(**HMENU**)**2**,** NULL**,** NULL**);**        *//* *创建Search\_lable*        Search\_lable **=** CreateWindow**("STATIC",** **"Search",** WS\_VISIBLE **|** WS\_CHILD **,** 280**,** 423**,** 70**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建Search\_C按钮*        Search\_C **=** CreateWindow**("BUTTON",** **"C",** WS\_VISIBLE **|** WS\_CHILD**,** 330**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**4**,** NULL**,** NULL**);**        *//* *创建Search\_M按钮*        Search\_M **=** CreateWindow**("BUTTON",** **"M",** WS\_VISIBLE **|** WS\_CHILD**,** 350**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**5**,** NULL**,** NULL**);**        *//* *创建Search\_E按钮*        Search\_E **=** CreateWindow**("BUTTON",** **"E",** WS\_VISIBLE **|** WS\_CHILD**,** 370**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**6**,** NULL**,** NULL**);**        *//* *创建删除按钮*        g\_hDeleteButton **=** CreateWindow**("BUTTON",** **"delete",** WS\_VISIBLE **|** WS\_CHILD**,** 353**,** 150**,** 100**,** 25**,** hwnd**,** **(**HMENU**)**3**,** NULL**,** NULL**);**        *//* *创建sort:的标签*        Sort\_lable **=** CreateWindow**("STATIC",** **"Sort",** WS\_VISIBLE **|** WS\_CHILD **,** 148**,** 423**,** 30**,** 25**,** hwnd**,** NULL**,** NULL**,** NULL**);**        *//* *创建sort* *Chinese的按钮*        Sort\_C **=** CreateWindow**("BUTTON","C",** WS\_VISIBLE **|** WS\_CHILD**,** 180**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**7**,** NULL**,** NULL**);**        *//* *创建sort* *Math的按钮*        Sort\_M **=** CreateWindow**("BUTTON","M",** WS\_VISIBLE **|** WS\_CHILD**,** 200**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**8**,** NULL**,** NULL**);**        *//* *创建sort* *English的按钮*        Sort\_E **=** CreateWindow**("BUTTON","E",** WS\_VISIBLE **|** WS\_CHILD**,** 220**,** 420**,** 20**,** 25**,** hwnd**,** **(**HMENU**)**9**,** NULL**,** NULL**);**        *//* *创建Main* *按钮,返回主页面*        Refresh **=** CreateWindow**("BUTTON","Refresh",** WS\_VISIBLE **|** WS\_CHILD**,** 50**,** 420**,** 70**,** 25**,** hwnd**,** **(**HMENU**)**14**,** NULL**,** NULL**);**        *//* *创建输出框*        Output\_text **=** CreateWindow**("EDIT",** **"",** WS\_VISIBLE **|** WS\_CHILD **|** WS\_BORDER **|** ES\_MULTILINE **|** ES\_AUTOVSCROLL **|** WS\_VSCROLL**,** 30**,** 190**,** 440**,** 220**,** hwnd**,** NULL**,** NULL**,** NULL**);**        **break;**    **case** WM\_COMMAND**:**        **switch** **(**LOWORD**(**wParam**))** **{**            *//为了减少代码,在search\_range的时候给subject赋值用的*            **int** temp\_for\_search**;**            **int** temp\_for\_sort**;**            **case** 1**:** *//* *插入按钮被点击*            **{**                *//* *获取姓名输入框中的文本*                **char** name**[**50**];**                GetWindowText**(**Name\_input**,** name**,** **sizeof(**name**));**                *//* *获取学号输入框中的文本*                **char** id**[**20**];**                GetWindowText**(**ID\_input**,** id**,** **sizeof(**id**));**                *//检查是否有与数据库中重复的name或者id*                **int** check **=** revise**(**name**,**id**,**g\_head**);**                *//* *获取成绩输入框中的三个文本*                **char** chinese\_get**[**5**],**math\_get**[**5**],**english\_get**[**5**];**                GetWindowText**(**Chinese\_input**,**chinese\_get**,sizeof(**chinese\_get**));**                GetWindowText**(**Math\_input**,**math\_get**,sizeof(**math\_get**));**                GetWindowText**(**English\_input**,**english\_get**,sizeof(**english\_get**));**                Student student\_get**;**                Grade grade\_get**;**                *//检测输入信息是否有空的*                **if(!**strcmp**(**name**,"")** **||** **!**strcmp**(**id**,"")** **||** **!**strcmp**(**chinese\_get**,"")** **||** **!**strcmp**(**math\_get**,"")** **||** **!**strcmp**(**english\_get**,""))**                **{**                    SetWindowText**(**Output\_text**,"Incomplete information, please complete the information before inserting.**\r\n**");**                    **break;**                **}**                **else**                **{**                *//转换三个文本到int*                    grade\_get**.**Chinese **=** atoi**(**chinese\_get**);**                    grade\_get**.**Math **=** atoi**(**math\_get**);**                    grade\_get**.**English **=** atoi**(**english\_get**);**                    strcpy**(**student\_get**.**name**,**name**);**                    strcpy**(**student\_get**.**id**,**id**);**                    student\_get**.**grade **=** grade\_get**;**                      
                *//根据上面的返回值做修改操作*                    **if(**check **==** 1**)**                    **{**                    *//先删除原有的,再插入新的,当作修改*                        DeleteStudent**(**name**,**id**);**                        *//* *插入学生信息*                        InsertStudent**(**student\_get**,**g\_head**);**                        *//* *写入学生信息*                        Rewrite\_data**(**g\_head**);**                        *//* *更新输出框内容*                        UpdateOutputBox**(**g\_head**);**                    **}**                    **else**                    **{**                        *//* *插入学生信息*                        InsertStudent**(**student\_get**,**g\_head**);**                        *//* *写入学生信息*                        Rewrite\_data**(**g\_head**);**                        *//* *更新输出框内容*                        UpdateOutputBox**(**g\_head**);**                    **}**                **}**                  
            **}**            **break;**            **case** 2**:** *//* *查询按钮被点击*            **{**                *//* *获取姓名输入框中的文本*                **char** name**[**50**],**id**[**20**];**                GetWindowText**(**Name\_input**,** name**,** **sizeof(**name**));**                GetWindowText**(**ID\_input**,** id**,** **sizeof(**id**));**                *//判断信息给的是不是空的*                **if(**strcmp**(**name**,"")==**0 **&&** strcmp**(**id**,"")==**0**)**                **{**SetWindowText**(**Output\_text**,** **"Error,haven't input information.");**                    **return** 0**;}**                *//* *查询学生信息*                SearchStudent**(**name**,**id**);**            **}**            **break;**            **case** 3**:** *//* *删除按钮被点击*            **{**                *//* *获取姓名输入框中的文本*                **char** name**[**50**];**                **char** id**[**40**];**                GetWindowText**(**Name\_input**,** name**,** **sizeof(**name**));**                GetWindowText**(**ID\_input**,** id**,** **sizeof(**id**));**                **if(!**strcmp**(**name**,"")** **&&** **!**strcmp**(**id**,""))**                **{**                    SetWindowText**(**Output\_text**,"Incomplete information, please complete the information before deleting.**\r\n**");**                    **break;**                **}**                *//* *删除学生信息*                DeleteStudent**(**name**,**id**);**                Rewrite\_data**(**g\_head**);**                *//* *更新输出框内容*                UpdateOutputBox**(**g\_head**);**            **}**            **break;**            **case** 4**:** *//* *Search\_C按钮被点击*            **{**                temp\_for\_search **=** 1**;**                **goto** loop**;**            **}**            **case** 5**:** *//* *Search\_M按钮被点击*            **{**                temp\_for\_search **=** 2**;**                **goto**  loop**;**            **}**            **case** 6**:** *//* *Search\_E按钮被点击*            **{**                temp\_for\_search **=** 3**;**            **}**              
                *//* *最好是不用goto语句,在这里为了减少重复代码*            loop**:**      
            **{**                **char** high\_0**[**5**],**low\_0**[**5**];**                **int**  high**,**low**;**                *//获取高低成绩中的信息*                GetWindowText**(**Grade\_input\_1**,** high\_0**,** **sizeof(**high\_0**));**                GetWindowText**(**Grade\_input\_2**,** low\_0**,** **sizeof(**low\_0**));**                **if(!**strcmp**(**high\_0**,"")** **||** **!**strcmp**(**low\_0**,""))**                **{**                    SetWindowText**(**Output\_text**,"Incomplete information, please complete the Grade range information before searching.**\r\n**");**                    **break;**                **}**                **if(**atoi**(**high\_0**)** **>=** atoi**(**low\_0**))**      
                **{**high **=** atoi**(**high\_0**);**                    low **=** atoi**(**low\_0**);}**                **else**                **{**                    high **=** atoi**(**low\_0**);**                    low **=** atoi**(**high\_0**);**                **}**                UpdateOutputBox**(**Find\_range**(**high**,**low**,**temp\_for\_search**,**g\_head**,**range\_list**));**            **}**            **break;**            **case** 7**:** *//* *Sort\_C按钮被点击*            **{**                temp\_for\_sort **=** 1**;**                **goto** loop\_for\_sort**;**            **}**            **case** 8**:** *//* *Sort\_M按钮被点击*            **{**                temp\_for\_sort **=** 2**;**                **goto** loop\_for\_sort**;**            **}**            **case** 9**:** *//* *Sort\_E按钮被点击*            **{**                temp\_for\_sort **=** 3**;**            **}**            loop\_for\_sort**:**            **{**                Sort\_Student**(**g\_head**,**temp\_for\_sort**);**                Rewrite\_data**(**g\_head**);**                UpdateOutputBox**(**g\_head**);**            **}**            **break;**            **case** 14**:** *//* *Refresh按钮被点击*            **{**                UpdateOutputBox**(**g\_head**);**            **}**        **default:**            **break;**        **}**        **break;**    **case** WM\_DESTROY**:**        PostQuitMessage**(**0**);**        **return** 0**;**    **default:**        **return** DefWindowProc**(**hwnd**,** uMsg**,** wParam**,** lParam**);**    **}**    **return** 0**;  
}***//* *插入学生信息*Student InsertStudent**(**Student temp\_student**,**Student **\***head**)** **{**    Student**\*** newStudent **=** **(**Student**\*)**malloc**(sizeof(**Student**));**    strcpy**(**newStudent**->**name**,** temp\_student**.**name**);**    strcpy**(**newStudent**->**id**,** temp\_student**.**id**);**    newStudent**->**grade **=** temp\_student**.**grade**;**    newStudent**->**next **=** NULL**;**    Student**\*** p **=** head**;**    **while(**p**->**next **!=** NULL**)**        p **=** p**->**next**;**    p**->**next **=** newStudent**;**    **return** **\***newStudent**;**    *//尾插法*      
**}***//* *查询学生信息***void** SearchStudent**(char\*** name**,char\*** id**)** **{**      
    *//* *在链表中查找姓名为name的学生信息*    Student**\*** current **=** g\_head**->**next**;**    **while** **(**current **!=** NULL**)**   
    **{**        **if** **(**strcmp**(**current**->**name**,** name**)** **==** 0**)** **{**            **char** output**[**512**];**            sprintf**(**output**,** **"Name:%s**\r\n**ID:%s**\r\n**Chinese:%d**\r\n**Math:%d**\r\n**English:%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**            SetWindowText**(**Output\_text**,** output**);**            **return;**        **}**        **if** **(**strcmp**(**current**->**id**,** id**)** **==** 0**)** **{**            **char** output**[**512**];**            sprintf**(**output**,** **"Name:%s**\r\n**ID:%s**\r\n**Chinese:%d**\r\n**Math:%d**\r\n**English:%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**            SetWindowText**(**Output\_text**,** output**);**            **return;**        **}**        current **=** current**->**next**;**    **}**    *//* *没有找到*    SetWindowText**(**Output\_text**,** **"Can't find this student.");  
}***//* *删除学生信息***void** DeleteStudent**(char\*** name**,char\*** id**)** **{**    *//* *特殊情况：头节点后面的第一个节点就是符合的节点*    **if** **(**strcmp**(**g\_head**->**next**->**name**,** name**)** **==** 0**)** **{**        Student**\*** temp **=** g\_head**->**next**;**        g\_head**->**next **=** temp**->**next**;**        free**(**temp**);**        **return;**    **}**    **else** **if** **(**strcmp**(**g\_head**->**next**->**id**,** id**)** **==** 0**)** **{**        Student**\*** temp **=** g\_head**->**next**;**        g\_head**->**next **=** temp**->**next**;**        free**(**temp**);**        **return;**    **}**    *//* *遍历链表找到要删除的节点的前一个节点*    Student**\*** current **=** g\_head**;**    **while** **(**current**->**next **!=** NULL**)** **{**        **if** **(**strcmp**(**current**->**next**->**name**,** name**)** **==** 0**)** **{**            Student**\*** temp **=** current**->**next**;**            current**->**next **=** current**->**next**->**next**;**            free**(**temp**);**            **return;**        **}**        current **=** current**->**next**;**    **}**    current **=** g\_head**;**    **while** **(**current**->**next **!=** NULL**)** **{**        **if** **(**strcmp**(**current**->**next**->**id**,** id**)** **==** 0**)** **{**            Student**\*** temp **=** current**->**next**;**            current**->**next **=** current**->**next**->**next**;**            free**(**temp**);**            **return;**        **}**        current **=** current**->**next**;**    **}  
}***//在文本中删除学生的信息***void** Rewrite\_data**(**Student **\***head**)  
{**    student\_data **=** fopen**("student\_data.txt","w");***//* *先清空文件，然后再根据链表处理的结果重新写入*    student\_data **=** fopen**("student\_data.txt","a+");**    **if** **(**student\_data **==** NULL**)** **{**        SetWindowText**(**Output\_text**,"Can't open the file**\n**");**    **}***//打开文件并且检查文件是否存在*    Student **\***temp **=** head**->**next**;**    **while(**temp **!=** NULL**)**    **{**        fprintf**(**student\_data**,** **"Name:%s**\t\t**ID:%s**\t\t**Grade.Chinese:%d**\t\t**Grade.Math:%d**\t\t**Grade.English:%d**\n**",**        temp**->**name**,**temp**->**id**,**temp**->**grade**.**Chinese**,**temp**->**grade**.**Math**,**temp**->**grade**.**English**);**        temp **=** temp**->**next**;***//一行太长了，换个行好看点*    **}**    fclose**(**student\_data**);**      
**}***//* *更新输出框内容，以表格形式显示学生信息***void** UpdateOutputBox**(**Student**\*** head**)** **{**    *//* *清空输出框内容*    SetWindowText**(**Output\_text**,"");**    **int** flag**=**0**;**    *//* *遍历链表，显示所有学生信息*    Student**\*** current **=** head**->**next**;**    *//flag的设置是为了让第一行显示名称*    **while** **(**current **!=** NULL**)** **{**          
        *//接下来一整段if运用是为了在第一行显示名称,空格和\t混用为了排版整齐*        **char** output**[**512**];**        **if(**flag**==**0**)**        **{**            sprintf**(**output**,"Name**\t\t\t**ID          Chinese**\t**Math   English**\r\n**");**            sprintf**(**output**+**strlen**(**output**),** **"%-20s**\t        **%s**\t**%d**\t **%d**\t**%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**            flag**=**1**;**        **}**        **else**        **{**            sprintf**(**output**,** **"%-20s**\t        **%s**\t**%d**\t **%d**\t**%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**        **}**        *//这里Name后面\t换成空格是因为如果name后面有两个一样的字母，\t就会被略去，原因不明*        SendMessage**(**Output\_text**,** EM\_REPLACESEL**,** 0**,** **(**LPARAM**)**output**);**        current **=** current**->**next**;**    **}  
}***//每次都需要的初始化执行函数***int** Init\_linklist**(**Student **\***head**)  
{**    student\_data **=** fopen**("student\_data.txt",** **"r+");**    **if** **(**student\_data **==** NULL**)** **{**        SetWindowText**(**Output\_text**,"There's no file named 'student\_data.'**\r\n**Now successfully created a new one.");**        student\_data **=** fopen**("student\_data.txt",** **"w+");**   
    *//* *创建一个新文件*        **if** **(**student\_data **==** NULL**)** **{**            SetWindowText**(**Output\_text**,"Failed to creat a new file.**\n**");**            **return** 1**;**        **}**    **}**    *//* *文件已存在或者成功创建，继续处理*    **char** line**[**120**];** *//* *假设每行最多100个字符*    **char** name**[**40**],**id**[**20**];**    **int** chinese**,** math**,** english**;**    *//* *逐行读取文件内容并解析*    **while** **(**fgets**(**line**,** **sizeof(**line**),** student\_data**)** **!=** NULL**)** **{**        *//* *使用sscanf解析每行内容*        **if** **(**sscanf**(**line**,** **"Name:%39s**\t\t **ID:%19s**\t\t **Grade.Chinese:%d**\t\t **Grade.Math:%d**\t\t **Grade.English:%d**\n**",**            name**,** id**,** **&**chinese**,** **&**math**,** **&**english**)** **==** 5**)**   
        **{**            Student temp\_student**;**            strcpy**(**temp\_student**.**name**,**name**);**            strcpy**(**temp\_student**.**id**,**id**);**            Grade temp\_grade**;**            temp\_grade**.**Chinese **=** chinese**;**            temp\_grade**.**Math **=** math**;**            temp\_grade**.**English **=** english**;**            temp\_student**.**grade **=** temp\_grade**;**            InsertStudent**(**temp\_student**,**g\_head**);**        **}**   
        **else** **{**        SetWindowText**(**Output\_text**,** **"There's an error in reading the file.");**        **}**    **}**      
    *//这里直接抄用update的代码，但是防止之前的信息被刷掉，所以重改一点，不直接调用*    Student**\*** current **=** g\_head**->**next**;**    **int** flag**=**0**;**    *//flag的设置是为了让第一行显示名称*    **while** **(**current **!=** NULL**)** **{**          
        *//接下来一整段if运用是为了在第一行显示名称,空格和\t混用为了排版整齐*        **char** output**[**512**];**        **if(**flag**==**0**)**        **{**            sprintf**(**output**,"Name**\t\t\t**ID          Chinese**\t**Math   English**\r\n**");**            sprintf**(**output**+**strlen**(**output**),** **"%-20s**\t        **%s**\t**%d**\t **%d**\t**%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**            flag**=**1**;**        **}**        **else**        **{**            sprintf**(**output**,** **"%-20s**\t        **%s**\t**%d**\t **%d**\t**%d**\r\n**",** current**->**name**,** current**->**id**,**current**->**grade**.**Chinese**,**current**->**grade**.**Math**,**current**->**grade**.**English**);**        **}**        *//这里Name后面\t换成空格是因为如果name后面有两个一样的字母，\t就会被略去，原因不明*        SendMessage**(**Output\_text**,** EM\_REPLACESEL**,** 0**,** **(**LPARAM**)**output**);**        current **=** current**->**next**;**    **}**    *//* *设置滚动条位置到第一行开头*    SendMessage**(**Output\_text**,** EM\_SCROLL**,** SB\_TOP**,** 0**);**    *//* *关闭文件*    fclose**(**student\_data**);**    **return** 0**;  
}***//给链表排序用的函数  
//subject用来判断比较哪一门科目***void** Sort\_Student**(**Student **\***head**,** **int** subject**)** **{**    **if** **(**head**->**next **==** NULL **||** head**->**next**->**next **==** NULL**)**        **return;**      
*//* *创建一个新的头节点作为已排序链表的头部*    Student **\***sorted\_head **=** **(**Student **\*)**malloc**(sizeof(**Student**));**    sorted\_head**->**next **=** NULL**;**      
*//* *遍历原始链表，依次将节点插入到已排序链表中*    Student **\***current **=** head**->**next**;**    **while** **(**current **!=** NULL**)** **{**        Student **\***next\_node **=** current**->**next**;**   
*//* *保存下一个节点的指针*        current**->**next **=** NULL**;***//* *断开当前节点与原链表的连接*          
        *//* *找到插入位置*        Student **\***prev **=** sorted\_head**;**        **while** **(**prev**->**next **!=** NULL **&&** **!**CompareStudent**(**prev**->**next**,** current**,** subject**))** **{**            prev **=** prev**->**next**;**        **}**          
        *//* *插入节点*        current**->**next **=** prev**->**next**;**        prev**->**next **=** current**;**          
        *//* *继续处理下一个节点*        current **=** next\_node**;**    **}**      
    *//* *将排序后的链表接入到原始链表中*    head**->**next **=** sorted\_head**->**next**;**      
    *//* *释放新建的头节点*    free**(**sorted\_head**);  
}***//* *用于检测是否有重复的名称或者id,同时返回值以供修改信息***int** revise**(char\*** name**,char\*** id**,**Student**\*** head**)  
{**    Student **\***temp **=** head**;**    **while(**temp**->**next **!=** NULL**)**    **{**        **if(**strcmp**(**name**,**temp**->**name**)** **||** strcmp**(**id**,**temp**->**id**))**            **return** 1**;**        *//发现重复的id或者name就返回1,然后进行修改操作*        temp**=**temp**->**next**;**    **}**    **return** 0**;  
}***//前面比后面大就返回1,否则返回0***int** CompareStudent**(**Student**\*** a**,** Student**\*** b**,** **int** subject**)** **{**    *//* *根据指定科目进行比较*    **switch** **(**subject**)** **{**        **case** 1**:** *//* *Chinese*        **return** **(**a**->**grade**.**Chinese **>=** b**->**grade**.**Chinese**)** **?** 1 **:** 0**;**        **case** 2**:** *//* *Math*        **return** **(**a**->**grade**.**Math **>=** b**->**grade**.**Math**)** **?** 1 **:** 0**;**        **case** 3**:** *//* *English*        **return** **(**a**->**grade**.**English **>=** b**->**grade**.**English**)** **?** 1 **:** 0**;**    **default:**        *//* *默认按照姓名进行比较*        **return** strcmp**(**a**->**name**,** b**->**name**)** **>=** 0 **?** 1 **:** 0**;**    **}  
}**Student**\*** Find\_range**(int** high**,** **int** low**,** **int** subject**,** Student**\*** head**,** Student**\*** range\_list**)** **{**    Student**\*** p **=** head**->**next**;**    Student**\*** p\_range **=** range\_list**;**      
*//* *清空range\_list，释放之前可能存在的节点*    **while** **(**p\_range **!=** NULL**)** **{**        Student**\*** temp **=** p\_range**;**        p\_range **=** p\_range**->**next**;**        free**(**temp**);**    **}**      
*//* *初始化range\_list*    range\_list **=** NULL**;**      
    **while** **(**p **!=** NULL**)** **{**        **if** **(**Range\_compare**(**p**,** subject**,** high**,** low**))** **{**            Student**\*** new\_student **=** **(**Student**\*)**malloc**(sizeof(**Student**));**            **if** **(**new\_student **==** NULL**)** **{***//* *处理内存分配失败*                SetWindowText**(**Output\_text**,"Memory allocation failed.**\n**");**                exit**(**1**);**            **}**            **\***new\_student **=** **\***p**;**            new\_student**->**next **=** NULL**;**              
            **if** **(**range\_list **==** NULL**)** **{**                range\_list **=** new\_student**;**            **}** **else** **{***//* *将新节点添加到range\_list的末尾*                Student**\*** temp **=** range\_list**;**                **while** **(**temp**->**next **!=** NULL**)** **{**                    temp **=** temp**->**next**;**                **}**                temp**->**next **=** new\_student**;**            **}**        **}**        p **=** p**->**next**;**    **}**    **return** range\_list**;  
}  
  
  
  
int** Range\_compare**(**Student**\*** stu**,int** subject**,int** high**,int** low**)  
{**    **switch** **(**subject**)** **{**        **case** 1**:** *//* *Chinese*        **return** **(**stu**->**grade**.**Chinese **>=** low **&&** stu**->**grade**.**Chinese **<=** high**)** **?** 1 **:** 0**;**        **case** 2**:** *//* *Math*        **return** **(**stu**->**grade**.**Math **>=** low **&&** stu**->**grade**.**Math **<=** high**)** **?** 1 **:** 0**;**        **case** 3**:** *//* *English*        **return** **(**stu**->**grade**.**English **>=** low **&&** stu**->**grade**.**English **<=** high**)** **?** 1 **:** 0**;**    **}**    **return** 0**;  
}  
  
  
int** WINAPI WinMain**(**HINSTANCE hInstance**,** HINSTANCE hPrevInstance**,** LPSTR lpCmdLine**,** **int** nCmdShow**)** **{**    *//分配头指针内存,防止内存出问题*    g\_head **=** **(**Student**\*)**malloc**(sizeof(**Student**));**    **if** **(**g\_head **==** NULL**)** **{**        fprintf**(**stderr**,** **"Error memory of head**\n**");**        **return** 1**;**    **}**    g\_head**->**next **=** NULL**;***//* *注册窗口类*      
    WNDCLASS wc **=** **{**0**};**    wc**.**lpfnWndProc **=** WindowProc**;**    wc**.**hInstance **=** hInstance**;**    wc**.**lpszClassName **=** **"MyWindowClass";**    RegisterClass**(&**wc**);**      
    *//* *创建窗口*    HWND hwnd **=** CreateWindowEx**(**0**,** **"MyWindowClass",** **"Students' information manager",** WS\_OVERLAPPEDWINDOW**,** CW\_USEDEFAULT**,** CW\_USEDEFAULT**,** 500**,** 500**,** NULL**,** NULL**,** hInstance**,** NULL**);**      
    *//* *显示窗口*    ShowWindow**(**hwnd**,** nCmdShow**);**    Init\_linklist**(**g\_head**);**    *//* *消息循环*    MSG msg**;**    **while(**GetMessage**(&**msg**,** NULL**,** 0**,** 0**))** **{**        TranslateMessage**(&**msg**);**        DispatchMessage**(&**msg**);**    **}**      
    **return** 0**;  
}**

## 5.编译成功界面截图



## 6.运行界面

## (1).初始化执行:

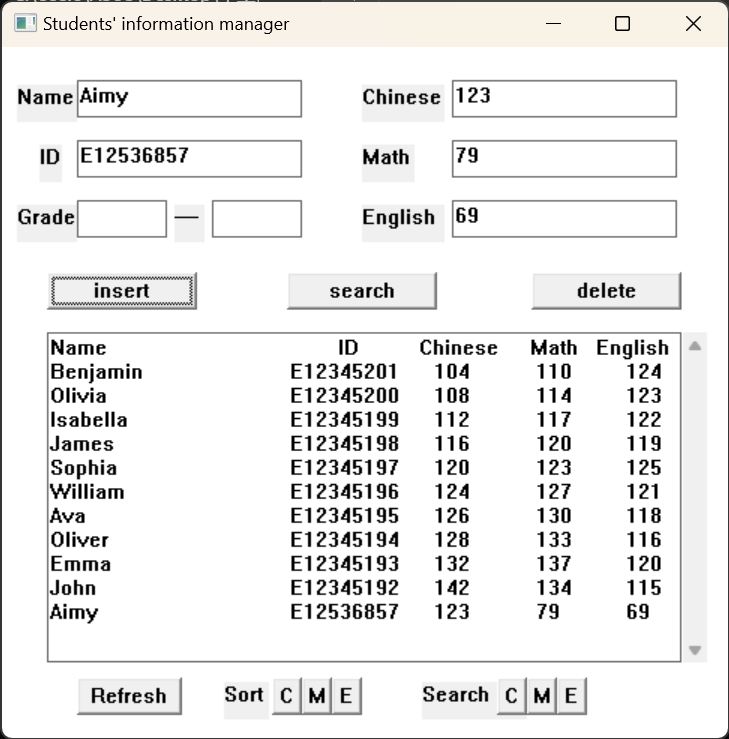


## (2).插入操作:

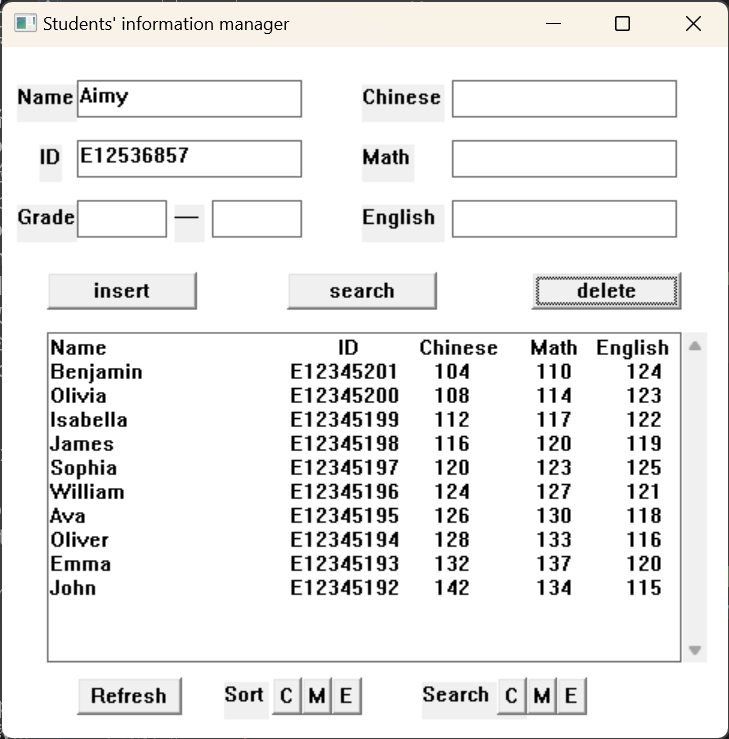
**//代码做了分辨,插入时可以检测信息是否输入完整,如果信息缺失,会显示相应提示**

## (3).修改操作:

**//代码特地做了个rivise函数,用来判断是否有相同的姓名或者ID,如果数据库中已经有了相同name或者ID的数据,那么将直接删除原有数据,并将新数据写入,以此来代替修改。**



## (4).删除操作:



### 四、总结

通过本次实验，我深入了解了链表数据结构的应用，掌握了Windows窗口程序设计的基本方法，提高了编程能力和问题解决能力。同时，通过实现学生信息管理系统，我进一步理解了数据结构与算法在实际项目中的应用，对软件开发流程有了更深入的了解。

# 操作说明:

**insert按钮:**在Name，ID，Chinese，Math，English,相关输入框中填入数据,便可以插入学生数据，如果有输入框是空的，就会返回提示，并且写入Student\_data.txt

**search按钮:**在Name，ID框中输入数据，优先查询Name,如果Name为空，则按照ID查询，都为空则返回提示。

**Delete 按钮:** 在Name，ID框中输入数据，优先按照Name删除,如果Name为空，则按照ID删除，都为空则返回提示。

**Refresh 按钮:** 在进行插入，查询，删除操作后，若界面未回到原界面，则刷新页面，回到最开始的数据界面。

**Sort C/M/E:**分别是按语文/数学/英语成绩从低到高对信息进行排列并且刷新界面。

**Search C/M/E:**分别是按按语文/数学/英语成绩范围查询学生信息,期中Grade后面的两个输入框都不能为空,否则返回提示,对于大小顺序没有要求,可以后面的框填较小的数,前面的框填较大的数。