学生信息管理系统

PB20000296 郑滕飞

功能

- 初始化时可选择以顺序表存储方式建立文本数据文件或以单链表存储方式建立二进制数据文件，以文件为初始数据来源
- 分为需输入密码的管理员权限与无需密码的学生权限，可执行不同功能
- 学生权限可查询、统计数据，并输出成文件
- 管理员权限可修改数据（插入、删除、更改）、更改密码、全部初始化
- 以菜单形式执行上述功能，并含帮助项可查阅具体功能，当使用出现问题时将响铃并报错

数据

- 以本班学生数据为实验
- 包括序号（int iSerial）、学号（char sStuNum[11]）、姓名（char sStuName[10]）、性别（bool bGender）、成绩（short int siCScore）

平台

- 硬件平台：LAPTOP-TREDUDDB/64位操作系统/x64处理器
- 软件平台：Windows 10 家庭中文版

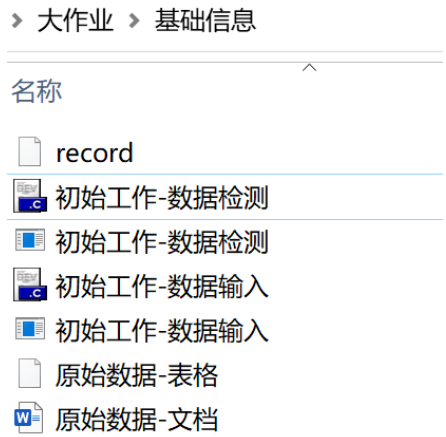
最终成果与部分代码展示

PART 1 公共部分

STEP I 原始信息输入（进入主程序前）

- 思路:
- 将原始的表格文档增添“性别”“成绩”项目，并输入数据
- 将数据直接复制入文本文档，接着替换制表符为空格
- 复制文本文档后利用合适的小程序可一次输入全部数据
- 利用另一单独小程序检测数据是否已正确输入

- 以下文件均在“基础信息”文件夹中
- 表格：原始数据-表格.xlsx
- 文档：原始数据-文档.docx
- 输入数据小程序：初始工作-数据输入.c/.exe
- 检测数据小程序：初始工作-数据检测.c/.exe
- 输出的文件名：record.dat



1.1.1-表格与文本文档

- 两幅图数据不同是因为表格生成伪成绩利用了随机函数，每次进入时随机。
- 最终数据以右侧的文本文档为准，录入dat文件与后续处理均是以后者为基础的。

3	序号	学号	姓名	性别	成绩（伪）
4	1	JL20010018	郑源	1	84
5	2	JL20010019	袁承毅	1	58
6	3	JL20010020	刘驰东	1	82
7	4	JL20010021	王一凡	1	95
8	5	PB20000075	魏莱	0	54
9	6	PB20000083	左雯杰	0	85
10	7	PB20000134	金瑜洋	0	50
11	8	PB20000187	钱文翰	1	100
12	9	PB20000241	杨浩然	1	76
13	10	PB20000296	郑滕飞	1	57
14	11	PB20000311	李鉴纯	1	81
15	12	PB20000334	曾其民	1	54
16	13	PB20010346	成政桦	1	82
17	14	PB20010347	崔扬波	1	78
18	15	PB20010349	董祥森	1	100
19	16	PB20010350	冯嘉源	1	58
20	17	PB20010354	何畅	1	60

1 JL20010018 郑源 1 73↵
2 JL20010019 袁承毅 1 99↵
3 JL20010020 刘驰东 1 80↵
4 JL20010021 王一凡 1 71↵
5 PB20000075 魏莱 0 93↵
6 PB20000083 左雯杰 0 72↵
7 PB20000134 金瑜洋 0 67↵
8 PB20000187 钱文翰 1 79↵
9 PB20000241 杨浩然 1 96↵
10 PB20000296 郑滕飞 1 76↵
11 PB20000311 李鉴纯 1 86↵
12 PB20000334 曾其民 1 78↵
13 PB20010346 成政桦 1 59↵
14 PB20010347 崔扬波 1 84↵
15 PB20010349 董祥森 1 77↵
16 PB20010350 冯嘉源 1 61↵
17 PB20010354 何畅 1 98↵
18 PB20010355 贺靖翔 1 57↵
19 PB20010358 李逸飞 1 97↵
20 PB20010359 李昭庆 1 60↵
21 PB20010361 林皓钧 1 57↵

1.1.2-数据输入与检测

D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\数据输入.exe

```
INPUT
106 PB20030774 谭午烜 1 85

INPUT
107 PB20030869 梅铁桦 1 73

INPUT
108 PB20061208 张称圆 1 95

INPUT
109 PB20061304 白寅岐 1 65

INPUT
110 PB20061322 徐思诚 1 51

INPUT
111 PB20061327 郑涛 1 57

INPUT
112 PB20111674 程千里 1 55

INPUT
113 PL20001001 张涵予 1 96

INPUT
114 PL20001002 陈嘉颖 0 85

--------------------------------
Process exited after 4.43 seconds with return value 0
请按任意键继续. . .
```

```c
typedef struct Data {
    int iSerial;
    char sStuNum[11];
    char sStuName[10];
    bool bGender;
    short siCScore;
} dat;

void main(void)
{
    dat info[114];
    int i;
    FILE *fp;
    fp = fopen("record", "wb");
    for(i = 0; i < 114; i++) {
        puts("\nINPUT");
        scanf("%d", &info[i].iSerial);
        scanf("%s", &info[i].sStuNum);
        scanf("%s", &info[i].sStuName);
        scanf("%hd", &info[i].bGender);
        scanf("%hd", &info[i].siCScore);
        fwrite(&info[i], sizeof(dat), 1, fp);
    }
    fclose(fp);
}
```

D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\基础信息\初始工作-数据检测.exe

```
89      PB20010451      何昌泽  1       89
90      PB20010453      李栋薪  1       82
91      PB20010454      李天语  1       98
92      PB20010455      梁居正  1       70
93      PB20010458      曲司成  1       98
94      PB20010461      王若暄  1       74
95      PB20010462      王喆卫  1       62
96      PB20010463      吴迪    1       86
97      PB20010464      夏小凡  1       56
98      PB20010465      熊易    1       82
99      PB20010466      徐潇彬  1       54
100     PB20010468      余少秋  1       98
101     PB20010470      张锦华  1       53
102     PB20010471      赵世博  1       51
103     PB20010472      邹子桢  1       59
104     PB20020572      齐俞迪
105     PB20020652      杜涵宇
106     PB20030774      谭午烜
107     PB20030869      梅铁桦
108     PB20061208      张称圆
109     PB20061304      白寅岐
110     PB20061322      徐思诚
111     PB20061327      郑涛
112     PB20111674      程千里
113     PL20001001      张涵予
114     PL20001002      陈嘉颖

--------------------------------
Process exited after 0.7274 seco
请按任意键继续. . .
```

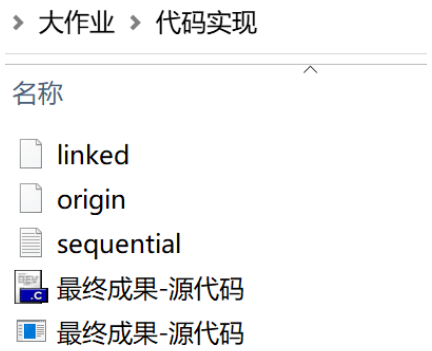
```c
typedef struct Data {
    int iSerial;
    char sStuNum[11];
    char sStuName[10];
    bool bGender;
    short siCScore;
} dat;

void main(void)
{
    dat info[114];
    int i;
    FILE *fp;
    if(!(fp = fopen("record", "rb"))) return;
    for(i = 0; i < 114; i++) {
        fread(&info[i], sizeof(dat), 1, fp);
        printf("%d\t", info[i].iSerial);
        printf("%s\t", info[i].sStuNum);
        printf("%s\t", info[i].sStuName);
        printf("%hd\t", info[i].bGender);
        printf("%hd\n", info[i].siCScore);
    }
    fclose(fp);
}
```

- 原始数据共114条，经检测输入正确，故封装使用。

STEP II 选择与初始化

- 思路：
- 首先将上一步中制作的record.dat复制入主代码文件夹，改名为origin.dat
- 在主程序中，初始数据为只读，不可改写
- 每次初始化时可选择顺序表/链表形式，并设置密码
- 两种形式进入的菜单功能完全相同，但实现方式有极大区别

- 以下文件均在代码实现文件夹中
- 主程序：最终成果-源代码.c/.exe
- 初始数据：origin.dat



1.2.1-声明预处理与结构体

- Student 为链表形式准备
- Data 为顺序表形式准备
- <stdbool.h>用于性别
- <string.h>用于比较字符串等
- 定义N用于链表生成

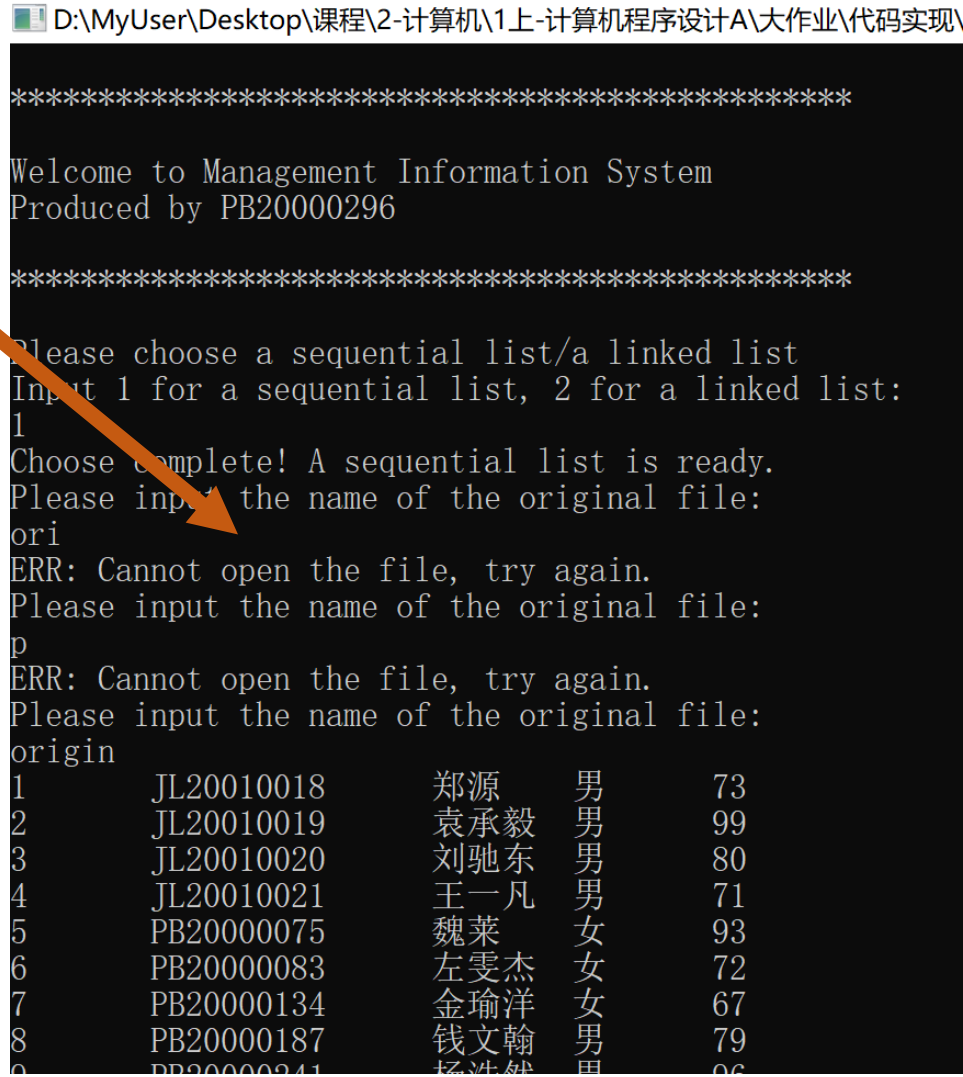
最终成果-源代码.c

```c
//学生信息管理系统
//制作者：郑滕飞
//学号：PB20000296

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#define N sizeof(stu)

typedef struct Student {
    int iSerial;
    char sStuNum[11];
    char sStuName[10];
    bool bGender;
    short siCScore;
    struct Student* next;
} stu;

typedef struct Data {
    int iSerial;
    char sStuNum[11];
    char sStuName[10];
    bool bGender;
    short siCScore;
} dat;
```

1.2.2-选择存储方式/输入初始数据

- 此处若输入不合要求字符，将响铃报错直到输入正确。
- （上方输入非1/2的数亦会如此）

```c
int choose(void)
{
    int c;
    puts("\n***********************************************\n");
    puts("Welcome to Management Information System");
    puts("Produced by PB20000296");
    puts("\n***********************************************\n");
    puts("Please choose a sequential list/a linked list");
    while (1) {
        puts("Input 1 for a sequential list, 2 for a linked list:");
        scanf("%d", &c);
        switch (c) {
            case 1:
                puts("Choose complete! A sequential list is ready.");
                return 0;

            case 2:
                puts("Choose complete! A linked list is ready.");
                return 1;

            default:
                puts("\aERR: Input error, try again.");
                while(getchar() != '\n');
        }
    }
}
```



1.2.3-设置密码

- 初始化细节将在后方叙述
- 此时密码已经被储存
- 确认密码失败将报错并重新输入

```
Reading complete!
Total: 114
Now please enter the new password, finishing with "enter"
now
Input again to confirm:
noww
ERR: Different input, try again.
Now please enter the new password, finishing with "enter"
now
Input again to confirm:
now
Initialization complete!
```

```
66  void enterPassword(char password[])
67  {
68      char s[20];
69      while(1) {
70          while(getchar() != '\n');
71          puts("Now please enter the new password, finishing with \"enter\"");
72          scanf("%[^\n]", password);
73          while(getchar() != '\n');
74          puts("Input again to confirm:");
75          scanf("%[^\n]", s);
76          if (!strcmp(password, s)) break;
77          puts("\aERR: Different input, try again.");
78      }
79  }
```

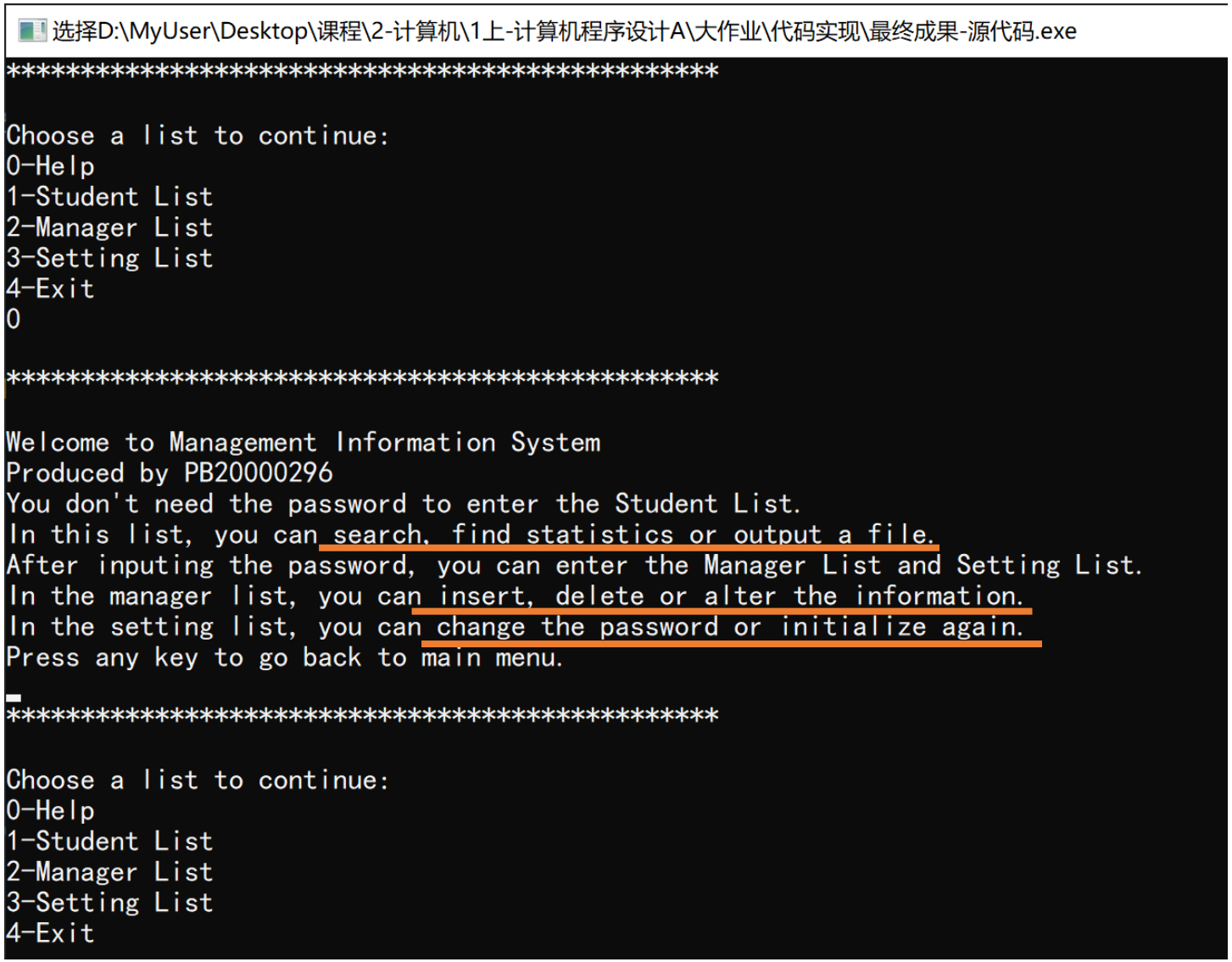
STEP III 主函数与主列表(Main Menu)

- 思路:
- 主列表可返回0或1，返回0则重新初始化，返回1则直接退出
- 一切_L为链表形式所专用的函数，一切_S为顺序表形式所专用的函数
- 存储顺序表时需要长度数据
- 存储链表时并不需要

```
900 int main(void) {
901     int n;
902     char password[20];
903     dat info[200];
904     stu *head;
905     while(1) {
906         if (choose()) {
907             head = initialize_L(password);
908             if(mainmenu_L(head, password)) return 0;
909         }
910         else {
911             n = initialize_S(info, password);
912             if(mainmenu_S(info, n, password)) return 0;
913         }
914     }
915 }
```

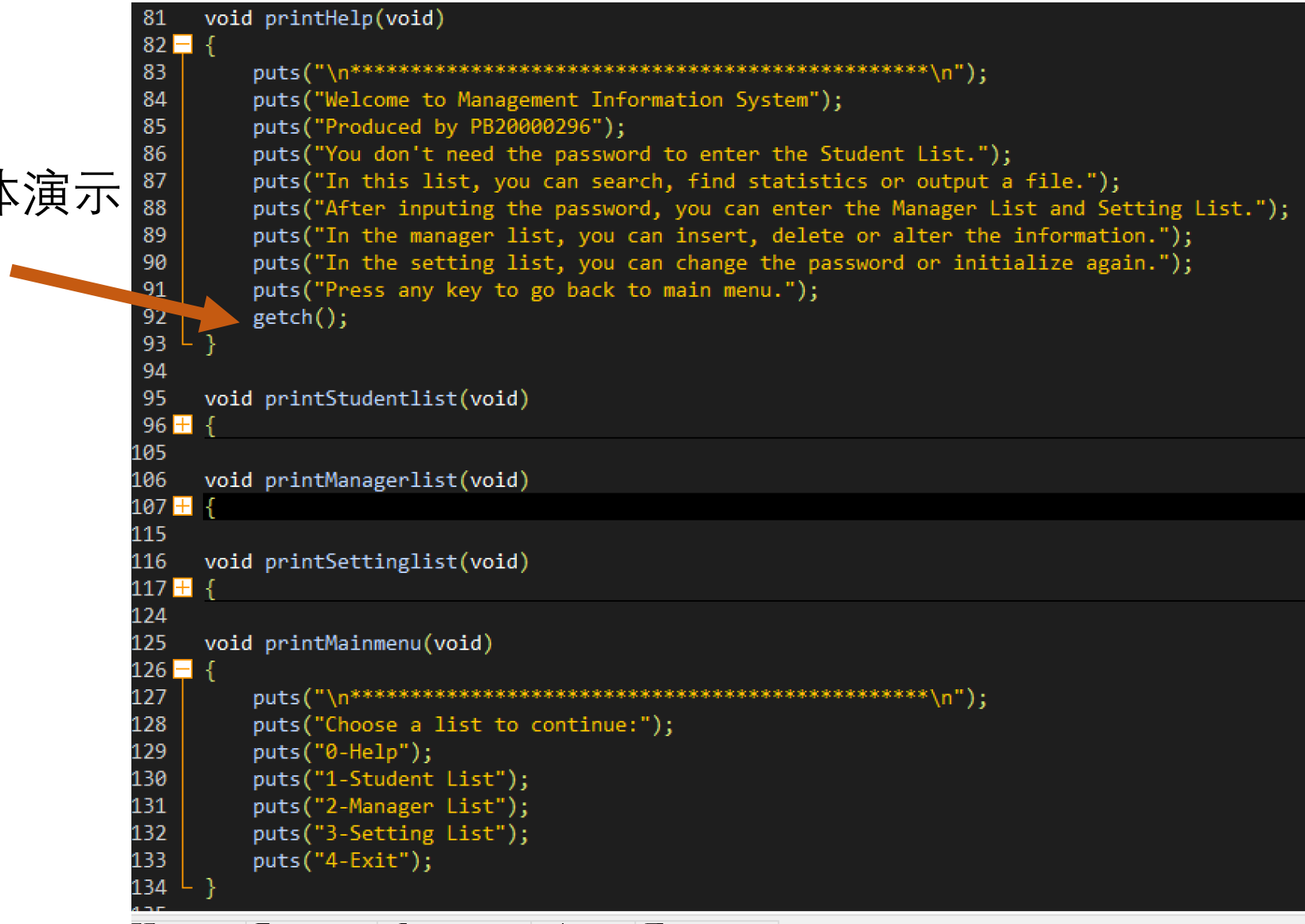
1.3.1-主列表与帮助

- 主列表可进入分列表
- 具体功能见帮助
- 学生列表无需密码
- 管理员与设置列表需密码



1.3.1-主列表与帮助

- 图为打印列表的函数
- 列表具体内容在后方具体演示
- 函数getch用于等待按键



```
81   void printHelp(void)
82   {
83       puts("\n***********************************************\n");
84       puts("Welcome to Management Information System");
85       puts("Produced by PB20000296");
86       puts("You don't need the password to enter the Student List.");
87       puts("In this list, you can search, find statistics or output a file.");
88       puts("After inputing the password, you can enter the Manager List and Setting List.");
89       puts("In the manager list, you can insert, delete or alter the information.");
90       puts("In the setting list, you can change the password or initialize again.");
91       puts("Press any key to go back to main menu.");
92       getch();
93   }
94
95   void printStudentlist(void)
96   {
105
106  void printManagerlist(void)
107  {
115
116  void printSettinglist(void)
117  {
124
125  void printMainmenu(void)
126  {
127      puts("\n***********************************************\n");
128      puts("Choose a list to continue:");
129      puts("0-Help");
130      puts("1-Student List");
131      puts("2-Manager List");
132      puts("3-Setting List");
133      puts("4-Exit");
134  }
```

1.3.2-分列表与键入密码

- 无密码列表直接进入
- 密码列表输入错误则无法进入
- 密码输入正确后正常进入

```
************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
2
Please enter the password
now

************************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
```

选择D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\代码实现

```
Now please enter the new password, finishing with "enter"
now
Input again to confirm:
now
Initialization complete!

************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
1

************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
5

************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
2
Please enter the password
no
ERR: Wrong password.

------------------------------------------------
```

1.3.2-分列表与键入密码

- 通过函数输出0/1实现判断输入正误

```
55   int inputPassword(char password[])
56   {
57       char s[20];
58       while(getchar() != '\n');
59       puts("Please enter the password");
60       scanf("%[^\n]", s);
61       if (!strcmp(password, s)) return 0;
62       puts("\aERR: Wrong password.");
63       return 1;
64   }
65
```

1.3.3-返回上级与主列表返回值

- 按要求输入即可返回
- 主列表选择退出后返回1，结束程序
- 进入设置列表后可初始化
- 初始化使主列表返回0，再次循环
- （接下来介绍设置列表会叙述初始化相关操作）

```
**********************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
4

**********************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
4

--------------------------------
Process exited after 624.3 seconds with return value 0
请按任意键继续. . .
```

```c
int mainmenu_L(stu* head, char password[])
{
    int t;
    while(1) {
        while(getchar() != '\n');
        printMainmenu();
        scanf("%d", &t);
        switch(t) {
            case 0:
                printHelp();
                break;

            case 1:
                studentlist_L(head);
                break;

            case 2:
                managerlist_L(head, password);
                break;

            case 3:
                if(settinglist_L(head, password)) return 0;
                break;

            case 4:
                return 1;

            default:
                puts("\aERR: Input error, try again.");
        }
    }
}
```

STEP IV 设置列表(Setting List)

- 设置列表包含两项功能：修改密码与全部初始化
- 链表与顺序表的设置列表基本一致，但链表初始化前需释放链表空间

```
Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
3
Please enter the password
now

**************************************************

Choose an action to continue:
1-Change the password
2-Initialize
3-Exit to main menu
```

1.4.1-修改密码

- 修改密码仍使用输入密码的函数

```
**************************************************

Choose an action to continue:
1-Change the password
2-Initialize
3-Exit to main menu
1
Now please enter the new password, finishing with "enter"
no
Input again to confirm:
no
Complete!
```

```
844  int settinglist_S(char password[])
845  {
846      int t;
847      if(inputPassword(password)) return 0;
848      while(1) {
849          while(getchar() != '\n');
850          printSettinglist();
851          scanf("%d", &t);
852          switch(t) {
853              case 1:
854                  enterPassword(password);
855                  puts("Complete!");
856                  getch();
857                  break;
858
```

```
66   void enterPassword(char password[])
67   {
68       char s[20];
69       while(1) {
70           while(getchar() != '\n');
71           puts("Now please enter the new password, finishing with \"enter\"");
72           scanf("%[^\n]", password);
73           while(getchar() != '\n');
74           puts("Input again to confirm:");
75           scanf("%[^\n]", s);
76           if (!strcmp(password, s)) break;
77           puts("\aERR: Different input, try again.");
78       }
79   }
```

1.4.2-全部初始化

```
844  int settinglist_S(char password[])
845  {
846      int t;
847      if(inputPassword(password)) return 0;
848      while(1) {
849          while(getchar() != '\n');
850          printSettinglist();
851          scanf("%d", &t);
852          switch(t) {
853              case 1:
854                  enterPassword(password);
855                  puts("Complete!");
856                  getch();
857                  break;
858
859              case 2:
860                  return 1;
861
862              case 3:
863                  return 0;
864
865              default:
866                  puts("\aERR: Input error, try again.");
867          }
868      }
869  }
```

```
871  int mainmenu_S(dat info[], int length, char password[])
872  {
873      int t;
874      while(1) {
875          while(getchar() != '\n');
876          printMainmenu();
877          scanf("%d", &t);
878          switch(t) {
879              case 0:
880                  printHelp();
881                  break;
882
883              case 1:
884                  studentlist_S(info, length);
885                  break;
886
887              case 2:
888                  length = managerlist_S(info, length, password);
889                  break;
890
891              case 3:
892                  if(settinglist_S(password)) return 0;
893                  break;
894
```

```
904  int main(void) {
905      int n;
906      char password[20];
907      dat info[200];
908      stu *head;
909      while(1) {
910          if (choose()) {
911              head = initialize_L(password);
912              if(mainmenu_L(head, password)) return 0;
913          }
914          else {
915              n = initialize_S(info, password);
916              if(mainmenu_S(info, n, password)) return 0;
917          }
918      }
919  }
```

1.4.2-全部初始化

- 链表时需先释放原空间

```
483  int settinglist_L(stu* head, char password[])
484  {
485      int t;
486      stu* p = head;
487      if(inputPassword(password)) return 0;
488      while(1) {
489          while(getchar() != '\n');
490          printSettinglist();
491          scanf("%d", &t);
492          switch(t) {
493              case 1:
494                  enterPassword(password);
495                  puts("Complete!");
496                  getch();
497                  break;
498
499              case 2:
500                  while (p->next) {
501                      p = p->next;
502                      free(head);
503                      head = p;
504                  }
505                  free(head);
506                  return 1;
507
508              case 3:
509                  return 0;
510
511              default:
512                  puts("\aERR: Input error, try again.");
513          }
514      }
515  }
```

PART 2 顺序表部分

STEP I 初始化

- 顺序表初始化需提前在主函数中定义出结构体数组（足够多的空间）
- 顺序表需要变量存储长度（记录实际使用的空间）

```
904  int main(void) {
905      int n;
906      char password[20];
907      dat info[200];
908      stu *head;
909      while(1) {
910          if (choose()) {
911              head = initialize_L(password);
912              if(mainmenu_L(head, password)) return 0;
913          }
914          else {
915              n = initialize_S(info, password);
916              if(mainmenu_S(info, n, password)) return 0;
917          }
918      }
919  }
```

2.1.1-打开文件

- 需要输入读取的文件名，错误则重新输入

```
**************************************************

Welcome to Management Information System
Produced by PB20000296

**************************************************

Please choose a sequential list/a linked list
Input 1 for a sequential list, 2 for a linked list:
1
Choose complete! A sequential list is ready.
Please input the name of the original file:
ori
ERR: Cannot open the file, try again.
Please input the name of the original file:
p
ERR: Cannot open the file, try again.
Please input the name of the original file:
origin
1       JL20010018      郑源    男      73
2       JL20010019      袁承毅  男      99
3       JL20010020      刘驰东  男      80
```

```c
int initialize_S(dat info[], char password[])
{
    char s[20];
    int i = 0;
    dat temp;
    FILE *fp;
    while(1) {
        while(getchar() != '\n');
        puts("Please input the name of the original file:");
        scanf("%[^\n]", s);
        if(!(fp = fopen(s, "rb"))) puts("\aERR: Cannot open the file, try again.");
        else break;
    }
```

2.1.2-读取过程

- 使用临时变量读取，为了舍弃最后一组EOF开头的实际不存在的数据
- 由于初始数据中有序号，故序号(int iSerial)可直接读入
- 读取过程中打印以确认数据

```
136   void printdat_S(dat info)
137   {
138       printf("%d\t", info.iSerial);
139       printf("%s\t", info.sStuNum);
140       printf("%s\t", info.sStuName);
141       if (info.bGender) printf("%s", "男\t");
142       else printf("%s", "女\t");
143       printf("%hd\n", info.siCScore);
144   }
```

```
159       fread(&temp, sizeof(dat), 1, fp);
160       while(!feof(fp)) {
161           info[i] = temp;
162           printdat_S(info[i]);
163           i++;
164           fread(&temp, sizeof(dat), 1, fp);
165       }
166       puts("\nReading complete!");
167       printf("Total: %d\n", i);
168       fclose(fp);
169       enterPassword(password);
170       puts("Initialization complete!");
171       getch();
172       return i;
173   }
```

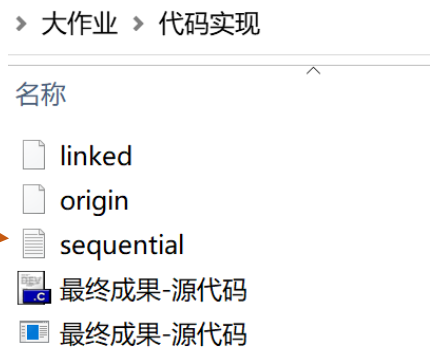
77	PB20010437	刘丰恺	男	88
78	PB20010438	吕继睿	男	50
79	PB20010439	罗宇帆	男	70
80	PB20010440	全孙嘉	男	54
81	PB20010441	王乐达	男	74
82	PB20010443	许宇澄	男	92
83	PB20010444	张可垚	男	79
84	PB20010445	陈鉴	男	77
85	PB20010446	陈少博	男	68
86	PB20010448	邓凯宁	男	99

STEP II 学生列表(Student List)

- 无需密码进入
- 拥有功能:
- 1-查找（输入学号/姓名，输出数据）
- 2-成绩查找（输入成绩区间，输出数据）
- 3-统计（各成绩区间人数、均分）
- 4-文件输出（顺序表为txt文件）

- 以下文件均在代码实现文件夹中
- 输出文件示例-sequential.txt

```
**********************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
1

**********************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
```



STEP II 学生列表(Student List)

- 由于不能改变数据于是无任何返回
- 四种功能分别对应函数

```
254  void studentlist_S(dat info[], int length)
255  {
256      int t;
257      while(1) {
258          while(getchar() != '\n');
259          printStudentlist();
260          scanf("%d", &t);
261          switch(t) {
262              case 1:
263                  search_S(info, length);
264                  break;
265
266              case 2:
267                  searchscore_S(info, length);
268                  break;
269
270              case 3:
271                  statistics_S(info, length);
272                  break;
273
274              case 4:
275                  output_S(info, length);
276                  break;
277
278              case 5:
279                  return;
280
281              default:
282                  puts("\aERR: Input error, try again.");
283          }
284      }
285  }
```

2.2.1-查找

- 查找支持学号与姓名
- 直接遍历全部长度以查找
- 若查找失败有报错提示

```
175   void search_S(dat info[], int length)
176   {
177       int i;
178       char s[20];
179       while(getchar() != '\n');
180       puts("Please input the student number or student name");
181       scanf("%s", s);
182       for (i = 0; i < length; i++) if (!(strcmp(s, info[i].sStuNum)*strcmp(s, info[i].sStuName))) {
183           printdat_S(info[i]);
184           getch();
185           return;
186       }
187       puts("\aERR: Data not found.");
188       getch();
189   }
```

```
**********************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
1
Please input the student number or student name
PB20000296
10      PB20000296      郑滕飞  男      76

**********************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
1
Please input the student number or student name
郑滕飞
10      PB20000296      郑滕飞  男      76

**********************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
1
Please input the student number or student name
PB20000000
ERR: Data not found.
```

2.2.2-成绩查找

- 两种可能出错方式
- 依然遍历查找
- 以变量统计总个数

```
Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
2
Input the beginning score:
75
Input the ending score:
75
ERR: Data not found.
```

```
191  void searchscore_S(dat info[], int length)
192  {
193      int begin, end, i, find = 0;
194      while(getchar() != '\n');
195      puts("Input the beginning score:");
196      scanf("%d", &begin);
197      while(getchar() != '\n');
198      puts("Input the ending score:");
199      scanf("%d", &end);
200      if (begin > end) {
201          puts("\aERR: begin is greater than end");
202          getch();
203          return;
204      }
205      for (i = 0; i < length; i++) if (info[i].siCScore >= begin && info[i].siCScore <= end) {
206          printdat_S(info[i]);
207          find++;
208      }
209      if (!find) puts("\aERR: Data not found.");
210      else printf("Total: %d", find);
211      getch();
212  }
```

2.2.2-成绩查找

- 当begin = end时输出特定成绩
- 当begin = 0, end = 100时输出全部列表

```
Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
2
Input the beginning score:
76
Input the ending score:
76
10      PB20000296      郑滕飞   男      76
Total: 1
```

```
**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
2
Input the beginning score:
0
Input the ending score:
100
1       JL20010018      郑源     男      73
2       JL20010019      袁承毅   男      99
3       JL20010020      刘驰东   男      80
4       JL20010021      王一凡   男      71
5       PB20000075      魏莱     女      93
6       PB20000083      左雯杰   女      72
7       PB20000134      金瑜洋   女      67
8       PB20000187      钱文翰   男      79
9       PB20000241      杨浩然   男      96
10      PB20000296      郑滕飞   男      76
```

```
98      PB20010465      熊易     男      82
99      PB20010466      徐潇彬   男      54
100     PB20010468      余少秋   男      98
101     PB20010470      张锦华   男      53
102     PB20010471      赵世博   男      51
103     PB20010472      邹子桢   男      59
104     PB20020572      齐俞迪   男      78
105     PB20020652      杜涵宇   女      50
106     PB20030774      谭午烜   男      85
107     PB20030869      梅铁桦   男      73
108     PB20061208      张称圆   男      95
109     PB20061304      白寅岐   男      65
110     PB20061322      徐思诚   男      51
111     PB20061327      郑涛     男      57
112     PB20111674      程千里   男      55
113     PL20001001      张涵予   男      96
114     PL20001002      陈嘉颖   女      85
Total: 114
**************************************************
```

2.2.3-统计数据

- 遍历方法将成绩基本按十位分为10段，输出每段人数
- 同时记录总和，计算出平均成绩

```
**********************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
3
1 to 10: 0
11 to 20: 0
21 to 30: 0
31 to 40: 0
41 to 50: 2
51 to 60: 23
61 to 70: 23
71 to 80: 22
81 to 90: 22
91 to 100: 22
Average: 75.21
```

```
214   void statistics_S(dat info[], int length)
215   {
216       int score[10], i;
217       long sum;
218       double average;
219       for (i = 0; i < 10; i++) score[i] = 0;
220       for (i = 0; i < length; i++) {
221           score[(info[i].siCScore - 1) / 10]++;
222           sum += info[i].siCScore;
223       }
224       for (i = 0; i < 10; i++) printf("%d to %d: %d\n", i * 10 + 1, i * 10 + 10, score[i]);
225       average = (double) sum / length;
226       printf("Average: %.2lf\n", average);
227       getch();
228   }
```

2.2.4-输出文件

- 由于输出txt文件，故使用fprintf
- 允许用户输入文件名，遍历输出

```
230   void output_S(dat info[], int length)
231   {
232       int i;
233       char s[20];
234       FILE *fp;
235       while(getchar() != '\n');
236       puts("Please input the name of the file:");
237       scanf("%s", s);
238       if (!(fp = fopen(strcat(s, ".txt"), "w"))) {
239           puts("\aERR: Fail to create the file.");
240           return;
241       }
242       for (i = 1; i < length; i++) {
243           fprintf(fp, "%d\t", info[i].iSerial);
244           fprintf(fp, "%s\t", info[i].sStuNum);
245           fprintf(fp, "%s\t", info[i].sStuName);
246           if (info[i].bGender) fprintf(fp, "%s", "男\t");
247           else fprintf(fp, "%s", "女\t");
248           fprintf(fp, "%hd\n", info[i].siCScore);
249       }
250       fclose(fp);
251       puts("Complete!");
252   }
```

```
**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
4
Please input the name of the file:
sequential
Complete!

**************************************************
```

sequential - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

29	PB20010371	乔雨豪	男	79
30	PB20010372	任宣霏	男	52
31	PB20010374	宋晨昊	男	88
32	PB20010375	宋京倍	男	72
33	PB20010376	孙守恒	男	86
34	PB20010377	孙思研	男	60
35	PB20010379	田浩乐	男	93
36	PB20010380	王昌盛	男	59
37	PB20010382	王溴然	男	88
38	PB20010383	王鹏翔	男	86
39	PB20010384	王枢臣	男	59
40	PB20010385	王帅坤	男	56

STEP III 管理员列表(Manager List)

- 需密码进入
- 拥有功能:
- 1-插入信息
- 2-删除信息
- 3-改变信息

- 可能改变长度，因此需要返回整数长度值

```
**********

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
2
Please enter the password
on

**********

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
```

```
400  int managerlist_S(dat info[], int length, char password[])
401  {
402      int t;
403      if(inputPassword(password)) return length;
404      while(1) {
405          while(getchar() != '\n');
406          printManagerlist();
407          scanf("%d", &t);
408          switch(t) {
409              case 1:
410                  length = insert_S(info, length);
411                  break;
412
413              case 2:
414                  length = delete_S(info, length);
415                  break;
416
417              case 3:
418                  change_S(info, length);
419                  break;
420
421              case 4:
422                  return length;
423
424              default:
425                  puts("\aERR: Input error, try again.");
426          }
427      }
428  }
```

2.3.1-插入数据

- 先判断是否已满，再判断输入是否合理
- 将插入位后全部从后往前起后移一位，且编号增加
- 输入插入的数据（依然纠错）
- 返回增加1的长度

```
287   int insert_S(dat info[], int length)
288   {
289       int i, t, read;
290       if (length == 200) {
291           puts("\aERR: Data is full.");
292           return length;
293       }
294       puts("Where do you want to insert? (0 means before the first data, etc.)");
295       scanf("%d", &t);
296       if (t > length || t < 0) {
297           puts("\aERR: Input error, try again.");
298           return length;
299       }
300       for (i = length - 1; i >= t; i--) {
301           info[i + 1] = info[i];
302           info[i + 1].iSerial++;
303       }
```

2.3.1-插入数据

```
**********************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
1
Where do you want to insert? (0 means before the first data, etc.)
15
Number:
PB00000000
Name:
TEST
Gender (1 for male, 0 for female)
1
Score:
98
Input complete!

**********
```

9	PB20000241	杨[illegible]	男	96
10	PB20000296	郑滕飞	男	76
11	PB20000311	李鉴纯	男	[illegible]
12	PB20000334	曾其民	男	78
13	PB20010346	成政桦	男	59
14	PB20010347	崔扬波	男	84
15	PB20010349	董祥森	男	77
16	PB00000000	TEST	男	98
17	PB20010350	冯嘉源	男	61
18	PB20010354	何畅	男	98
19	PB20010355	贺靖翔	男	57
20	PB20010358	李逸飞	男	97

```
304         puts("Number:");
305         scanf("%s", info[t].sStuNum);
306         puts("Name:");
307         scanf("%s", info[t].sStuName);
308         while(1) {
309             while(getchar() != '\n');
310             puts("Gender (1 for male, 0 for female):");
311             scanf("%d", &read);
312             if (!read || read == 1) {
313                 info[t].bGender = read;
314                 break;
315             }
316             puts("\aERR: Input error, try again.");
317         }
318         while(1) {
319             while(getchar() != '\n');
320             puts("Score:");
321             scanf("%d", &read);
322             if (read >= 0 && read <= 100) {
323                 info[t].siCScore = read;
324                 break;
325             }
326             puts("\aERR: Input error, try again.");
        }
        puts("Input complete!");
        return ++length;
}
```

- 键入的15表示第15位之后
- 即插入至16位

2.3.2-删除数据

- 先判断是否已空，再判断输入是否合理
- 显示此位数据，确认是否删除
- 将删除位后全部从前往后起前移一位
- 同时减少后部编号
- 返回减少1的长度

```
int delete_S(dat info[], int length)
{
    int i, t, read;
    if (length == 1) {
        puts("\aERR: Data can't be deleted to empty.");
        return length;
    }
    puts("Where do you want to delete?");
    scanf("%d", &t);
    if (t > length || t <= 0) {
        puts("\aERR: Input error, try again.");
        return length;
    }
    printdat_S(info[t - 1]);
    while(1) {
        while(getchar() != '\n');
        puts("Are you sure you want to delete this? (1 for yes, 0 for no)");
        scanf("%d", &read);
        if (!read) return length;
        if (read == 1) break;
        puts("\aERR: Input error, try again.");
    }
    for (i = t - 1; i < length; i++) {
        info[i] = info[i + 1];
        info[i].iSerial--;
    }
    puts("Delete complete!");
    return --length;
}
```

```
************************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
2
Where do you want to delete?
16
16      PB00000000      TEST    男      98
Are you sure you want to delete this? (1 for yes, 0 for no)
1
Delete complete!

************************************************
```

2.3.3-改变数据

- 判断输入是否合理
- 显示此位数据，并开始改写
- 改写时仍判断输入合理性
- 无返回

```
4       JL20010021      王一凡    男      71
5       PB20000075      魏莱      女      93
6       PB20000083      左雯杰    女      72
7       PB20000134      金瑜洋    女      67
8       PB20000187      钱文翰    男      79
9       PB20000241      杨浩然    男      96
10      PB20000296      郑滕飞    男      66
11      PB20000311      李鉴纯    男      86
12      PB20000334      曾其民    男      78
13      PB20010346      成政桦    男      59
```

```
**********************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
3
Where do you want to change?
10
10      PB20000296      郑滕飞    男      76
Input the new information
Number:
PB20000296
Name:
郑滕飞
Gender (1 for male, 0 for female):
1
Score:
66
Change complete!

**********************************************
```

```
void change_S(dat info[], int length)
{
    int i, t, read;
    puts("Where do you want to change?");
    scanf("%d", &t);
    if (t > length || t <= 0) {
        puts("\aERR: Input error, try again.");
        return;
    }
    printdat_S(info[t - 1]);
    puts("Input the new information");
    puts("Number:");
    scanf("%s", info[t - 1].sStuNum);
    puts("Name:");
    scanf("%s", info[t - 1].sStuName);
    while(1) {
        while(getchar() != '\n');
        puts("Gender (1 for male, 0 for female):");
        scanf("%d", &read);
        if (!read || read == 1) {
            info[t - 1].bGender = read;
            break;
        }
        puts("\aERR: Input error, try again.");
    }
    while(1) {
        while(getchar() != '\n');
        puts("Score:");
        scanf("%d", &read);
        if (read >= 0 && read <= 100) {
            info[t - 1].siCScore = read;
            break;
        }
        puts("\aERR: Input error, try again.");
    }
    puts("Change complete!");
}
```

PART 3 链表部分

- 除输出文件外效果与顺序表完全一致，故不再展示效果截图

STEP I 初始化

- 链表初始化需提前在主函数中定义出头结点
- 链表无需变量存储长度

```
904  int main(void) {
905      int n;
906      char password[20];
907      dat info[200];
908      stu *head;
909      while(1) {
910          if (choose()) {
911              head = initialize_L(password);
912              if(mainmenu_L(head, password)) return 0;
913          }
914          else {
915              n = initialize_S(info, password);
916              if(mainmenu_S(info, n, password)) return 0;
917          }
918      }
919  }
```

3.1.1-打开文件

- 需要输入读取的文件名，错误则重新输入
- 申请空间时判定是否成功

```
500   stu* initialize_L(char password[])
501   {
502       char s[20];
503       int i = 0;
504       dat temp;
505       stu *p, *head;
506       FILE *fp;
507       if (!(head = malloc(N))) puts("\aERR: Not enough space");
508       p = head;
509       while(1) {
510           while(getchar() != '\n');
511           puts("Please input the name of the original file:");
512           scanf("%[^\n]", s);
513           if(!(fp = fopen(s, "rb"))) puts("\aERR: Cannot open the file, try again.");
514           else break;
515       }
```

3.1.2-读取过程

- 使用临时变量读取，为了舍弃最后一组EOF开头的实际不存在的数据
- 由于读入为dat，实际需要stu，故有一项项赋值过程
- 读取过程中打印以确认数据
- 申请空间并建立链表

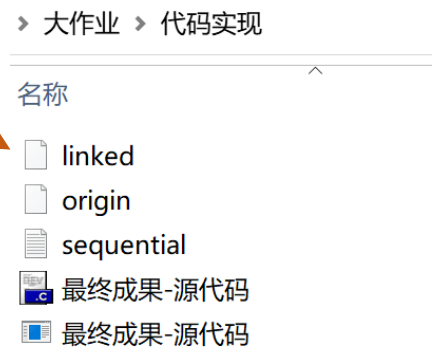
```
490   void printdat_L(stu *p)
491   {
492       printf("%d\t", p->iSerial);
493       printf("%s\t", p->sStuNum);
494       printf("%s\t", p->sStuName);
495       if (p->bGender) printf("%s", "男\t");
496       else printf("%s", "女\t");
497       printf("%hd\n", p->siCScore);
498   }
```

```
516       fread(&temp, sizeof(dat), 1, fp);
517       while(!feof(fp)) {
518           if (!(p->next = malloc(N))) puts("\aERR: Not enough space");
519           p = p->next;
520           p->iSerial = temp.iSerial;
521           strcpy(p->sStuNum, temp.sStuNum);
522           strcpy(p->sStuName, temp.sStuName);
523           p->bGender = temp.bGender;
524           p->siCScore = temp.siCScore;
525           printdat_L(p);
526           i++;
527           fread(&temp, sizeof(dat), 1, fp);
528       }
529       p->next = NULL;
530       puts("\nReading complete!");
531       printf("Total: %d\n", i);
532       fclose(fp);
533       enterPassword(password);
534       puts("Initialization complete!");
535       getch();
536       return head;
537   }
```

STEP II 学生列表(Student List)

- 无需密码进入
- 拥有功能：查找、成绩查找、统计、文件输出
- 文件输出为二进制文件
- 只需头节点一个参数

- 以下文件均在代码实现文件夹中
- 输出文件示例-linked.dat




```c
void studentlist_L(stu* head)
{
    int t;
    while(1) {
        while(getchar() != '\n');
        printStudentlist();
        scanf("%d", &t);
        switch(t) {
            case 1:
                search_L(head);
                break;

            case 2:
                searchscore_L(head);
                break;

            case 3:
                statistics_L(head);
                break;

            case 4:
                output_L(head);
                break;

            case 5:
                return;

            default:
                puts("\aERR: Input error, try again.");
        }
    }
}
```

3.2.1-查找

- 查找支持学号与姓名
- 直接遍历链表查找
- 若查找失败有报错提示
- 返回head时不改变
- 与顺序表几乎相同

```
539  void search_L(stu* head)
540  {
541      char s[20];
542      while (getchar() != '\n');
543      puts("Please input the student number or student name");
544      scanf("%s", s);
545      while (head->next) {
546          head = head->next;
547          if (!(strcmp(s, head->sStuNum)*strcmp(s, head->sStuName))) {
548              printdat_L(head);
549              getch();
550              return;
551          }
552      }
553      puts("\aERR: Data not found.");
554      getch();
555  }
556
```

3.2.2-成绩查找

- 两种可能出错方式
- 遍历查找
- 以变量统计总个数
- 与顺序表一致
- 头尾相同输出某成绩
- 头尾极值输出全部列表

```
557  void searchscore_L(stu* head)
558  {
559      int begin, end, find = 0;
560      while(getchar() != '\n');
561      puts("Input the beginning score:");
562      scanf("%d", &begin);
563      while(getchar() != '\n');
564      puts("Input the ending score:");
565      scanf("%d", &end);
566      if (begin > end) {
567          puts("\aERR: begin is greater than end");
568          getch();
569          return;
570      }
571      while(head->next) {
572          head = head->next;
573          if (head->siCScore >= begin && head->siCScore <= end) {
574              printdat_L(head);
575              find++;
576          }
577      }
578      if (!find) puts("\aERR: Data not found.");
579      else printf("Total: %d", find);
580      getch();
581  }
```

3.2.3-统计数据

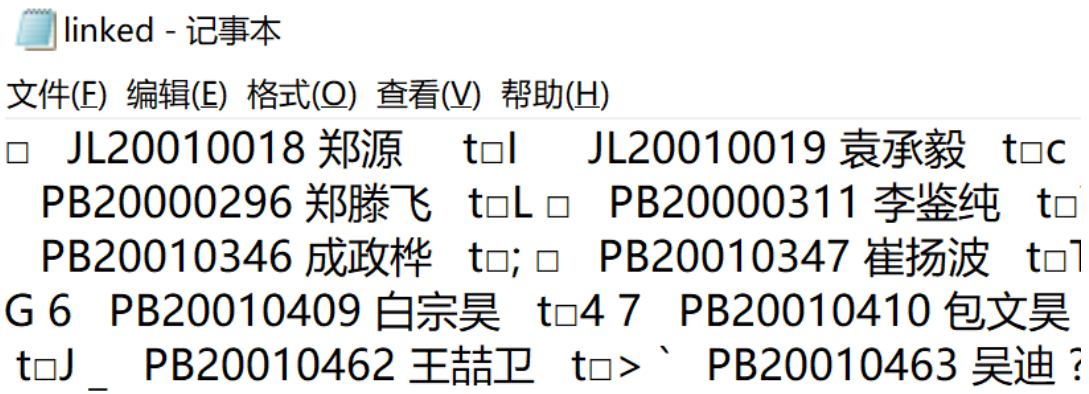
- 遍历方法将成绩基本按十位分为10段，输出每段人数
- 同时记录总和，计算出平均成绩
- 与顺序表一致

```
583  void statistics_L(stu* head)
584  {
585      int score[10], i, t;
586      long sum;
587      double average;
588      for (i = 0; i < 10; i++) score[i] = 0;
589      while(head->next) {
590          head = head->next;
591          t++;
592          score[(head->siCScore - 1) / 10]++;
593          sum += head->siCScore;
594      }
595      for (i = 0; i < 10; i++) printf("%d to %d: %d\n", i * 10 + 1, i * 10 + 10, score[i]);
596      average = (double) sum / t;
597      printf("Average: %.2lf\n", average);
598      getch();
599  }
```

3.2.4-输出文件

- 由于输出dat文件，故使用fwrite
- 允许用户输入文件名，遍历输出
- 利用临时变量存储输出
- 由于存储方式，记事本出现乱码

```
601  void output_L(stu* head)
602  {
603      int i;
604      char s[20];
605      FILE *fp;
606      dat temp;
607      while(getchar() != '\n');
608      puts("Please input the name of the file:");
609      scanf("%s", s);
610      if (!(fp = fopen(s, "wb"))) {
611          puts("\aERR: Fail to create the file.");
612          return;
613      }
614      while (head->next) {
615          head = head->next;
616          temp.iSerial = head->iSerial;
617          strcpy(temp.sStuNum, head->sStuNum);
618          strcpy(temp.sStuName, head->sStuName);
619          temp.bGender = head->bGender;
620          temp.siCScore = head->siCScore;
621          fwrite(&temp, sizeof(dat), 1, fp);
622      }
623      fclose(fp);
624      puts("Complete!");
625  }
626
```



STEP III 管理员列表(Manager List)

- 需密码进入
- 拥有功能:
- 1-插入信息
- 2-删除信息
- 3-改变信息

- 由于无需长度，依然不用返回
- 此时显著比顺序表方便

```
801  void managerlist_L(stu* head, char password[])
802  {
803      int t;
804      if(inputPassword(password)) return;
805      while(1) {
806          while(getchar() != '\n');
807          printManagerlist();
808          scanf("%d", &t);
809          switch(t) {
810              case 1:
811                  insert_L(head);
812                  break;
813
814              case 2:
815                  delete_L(head);
816                  break;
817
818              case 3:
819                  change_L(head);
820                  break;
821
822              case 4:
823                  return;
824
825              default:
826                  puts("\aERR: Input error, try again.");
827          }
828      }
829  }
```

3.3.1-插入数据

- 不用判断是否已满，直接判断输入是否合理
- 先不判断是否过大，到达指定位置前知是否过大
- 输入插入的数据（依然纠错）
- 链表中插入元素，后方全部编号增加

```
660  void insert_L(stu* head)
661  {
662      int i, t, read;
663      stu *q;
664      puts("Where do you want to insert? (0 means before the first data, etc.)");
665      scanf("%d", &t);
666      if (t < 0) {
667          puts("\aERR: Input error, try again.");
668          return;
669      }
670      for (i = 0; i < t; i++) {
671          if (!(head->next)) {
672              puts("\aERR: Input error, try again.");
673              return;
674          }
675          head = head->next;
676      }
```

```
    if (!(q = malloc(N))) puts("\aERR: Not enough space");
    puts("Number:");
    scanf("%s", q->sStuNum);
    puts("Name:");
    scanf("%s", q->sStuName);
    while(1) {
        while(getchar() != '\n');
        puts("Gender (1 for male, 0 for female):");
        scanf("%d", &read);
        if (!read || read == 1) {
            q->bGender = read;
            break;
        }
        puts("\aERR: Input error, try again.");
    }
    while(1) {
        while(getchar() != '\n');
        puts("Score:");
        scanf("%d", &read);
        if (read >= 0 && read <= 100) {
            q->siCScore = read;
            break;
        }
        puts("\aERR: Input error, try again.");
    }
    q->next = head->next;
    head->next = q;
    while (head->next) {
        head = head->next;
        t++;
        head->iSerial = t;
    }
    puts("Input complete!");
}
```

3.3.2-删除数据

- 先判断是否已空，再判断输入是否合理
- 显示此位数据，确认是否删除
- 将删除位后全部从前往后起前移一位
- 同时减少后部编号

```
712  void delete_L(stu* head)
713  {
714      int i, t, read;
715      stu *q;
716      if (!(head->next->next)) {
717          puts("\aERR: Data can't be deleted to empty.");
718          return;
719      }
720      puts("Where do you want to delete?");
721      scanf("%d", &t);
722      if (t <= 0) {
723          puts("\aERR: Input error, try again.");
724          return;
725      }
726      for (i = 0; i < t - 1; i++) {
727          if (!(head->next)) {
 28             puts("\aERR: Input error, try again.");
 29             return;
 30         }
 31         head = head->next;
 32     }
 33     if (!(head->next)) {
 34         puts("\aERR: Input error, try again.");
 35         return;
 36     }
```

```
737     printdat_L(head->next);
738     while(1) {
739         while(getchar() != '\n');
740         puts("Are you sure you want to delete this? (1 for yes, 0 for no)");
741         scanf("%d", &read);
742         if (!read) return;
743         if (read == 1) break;
744         puts("\aERR: Input error, try again.");
745     }
746     q = head->next;
747     head->next = head->next->next;
748     free(q);
749     while (head->next) {
750         head = head->next;
751         head->iSerial--;
752     }
753     puts("Delete complete!");
754 }
```

- 删除的同时释放空间

3.3.3-改变数据

- 判断输入是否合理
- 显示此位数据，并开始改写
- 改写时仍判断输入合理性

```
756  void change_L(stu* head)
757  {
758      int i, t, read;
759      puts("Where do you want to change?");
760      scanf("%d", &t);
761      if (t <= 0) {
762          puts("\aERR: Input error, try again.");
763          return;
764      }
765      for (i = 0; i < t; i++) {
766          if (!(head->next)) {
767              puts("\aERR: Input error, try again.");
768              return;
769          }
770          head = head->next;
771      }
```

```
772      printdat_L(head);
773      puts("Input the new information");
774      puts("Number:");
775      scanf("%s", head->sStuNum);
776      puts("Name:");
777      scanf("%s", head->sStuName);
778      while(1) {
779          while(getchar() != '\n');
780          puts("Gender (1 for male, 0 for female):");
781          scanf("%d", &read);
782          if (!read || read == 1) {
783              head->bGender = read;
784              break;
785          }
786          puts("\aERR: Input error, try again.");
787      }
788      while(1) {
789          while(getchar() != '\n');
790          puts("Score:");
791          scanf("%d", &read);
792          if (read >= 0 && read <= 100) {
793              head->siCScore = read;
794              break;
795          }
796          puts("\aERR: Input error, try again.");
797      }
798      puts("Change complete!");
799  }
```

PART 4 界面展示

D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\代码实现\最终成果-

```
Reading complete!
Total: 114
Now please enter the new password, finishing with "enter"
pol
Input again to confirm:
pol
Initialization complete!

**************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
2
Please enter the password
pol

**************************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
2
Where do you want to delete?
14
14      PB20010347      崔扬波  男      84
Are you sure you want to delete this? (1 for yes, 0 for no)
1
Delete complete!

**************************************************

Choose an action to continue:
1-Insert information
2-Delete information
3-Change information
4-Exit to main menu
4
```

D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\代

```
3-Change information
4-Exit to main menu
4

**************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
1

**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
2
Input the beginning score:
84
Input the ending score:
84
60      PB20010416      金维丰  男      84
Total: 1
**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
3
1 to 10: 0
11 to 20: 0
21 to 30: 0
31 to 40: 0
41 to 50: 2
51 to 60: 23
61 to 70: 23
71 to 80: 22
81 to 90: 21
```

D:\MyUser\Desktop\课程\2-计算机\1上-计算机程序设计A\大作业\代码实现\

```
**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
5

**************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
3
Please enter the password
pol

**************************************************

Choose an action to continue:
1-Change the password
2-Initialize
3-Exit to main menu
1
Now please enter the new password, finishing with "enter"
pk
Input again to confirm:
pk
Complete!

**************************************************

Choose an action to continue:
1-Change the password
2-Initialize
3-Exit to main menu
2

**************************************************

Welcome to Management Information System
Produced by PB20000296

**************************************************
```

```
Initialization complete!

**************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
1

**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
1
Please input the student number or student name
JL20010018
1       JL20010018      郑源    男      73

**************************************************

Choose an action to continue:
1-Search for information
2-Search by score
3-Find statistics
4-Output a file
5-Exit to main menu
5

**************************************************

Choose a list to continue:
0-Help
1-Student List
2-Manager List
3-Setting List
4-Exit
4

--------------------------------
Process exited after 21.22 seconds with return value 0
```

实验反思

- 开始总是试图直接读取文件导致形式不对而出错，二进制存储是更易于计算机处理的文件格式，转化为二进制后程序显著方便
- 制作顺序表时错误估计函数中应有的参数与返回值类型，功能无法实现或编译器报错时才意识到问题所在
- 制作链表时对形参、实参之间的传递方式分析不清，导致有时多建立变量，有时少建立变量
- 有时忘记处理返回值或关闭文件导致程序非正常关闭

实验报告至此结束

- 代码总行数：913
- 函数总个数：35
- 感谢阅读