目录

一、项目简介

（一）项目题目及内容简介

1.项目题目：《学生成绩分析与管理系统》

2.内容介绍：对不同的登录角色分配不同的操作界面与权限，根据友好的交互，通过简单的操作，对学生成绩进行管理和分析。

（二）项目模块划分

函数清单：

```
void init_node(void);//初始化链表
void load(void);//载入成绩数据
void save(void); //保存写入
void add_student(void);//添加成员
void write(void);//控制台写入信息
Node* search(char id[MAX]);//链表搜索
void print_one(Node* list);//打印输出信息
void delet(Node** list, char id[MAX]);//删除信息
void put_in_order(Node* phead, int code_data, int code_num);//排序函数
void f_rand(int a[], int* t); //随机数
void encrypt(char* location_1, char* code_location); //加密。加密前文本位置，密码位置
void decode(char* location_2, char* code_location); //解密。After_Encrypt 位置，密码位置
void insert_sort(Node* phead);//插入排序
int statistics(void);//人数统计
int rank(char id[MAX]);//排序
int extreme_value(Node* list, int choice, int code_num);//班级最值统计
double average_all(Node* list, int choice, int code_num);//班级均分统计
double average_one(Node* list, int choice);//个人均分统计
double variance_all(Node* phead);//班级波动指数
double variance_one(Node* p,int choice);//个人波动指数
void report(int code_num);//生成成绩报告
int analyze(void);//数据分析界面
void GPA(Node* phead);//GPA
void f_analyze(int change);//数据分析实现
void headview(void);//顶部图标
SMALL_RECT SizeOfWindow(HANDLE hConsoleOutput);//窗口调整
void modeset(int w, int h);//窗口设置大小
void surface(void);//窗口初始化
bool login(void);//登录界面
int awelcome(void);//管理员主菜单，返回选项
int swelcome(void);//学生主菜单，返回选项
void jump(void);//跳转界面
```

二、需求分析

（一）选题的依据

1、借助信息技术手段帮助管理和分析学生成绩。

2、现有的学生成绩管理系统功能过于单一，不能满足现实需求。

3、面向学生和老师，成绩管理系统应该具有不同的功能和权限，提供更加实际，更加多样化的服务。

4、成绩是一个需要相对较高安全保密的数据，而现有的成绩管理方式大多忽略了安全性这一特点，本程序为此提供了解决方案。

（二）功能需求

学生端

1、 考试成绩查询。

2、 GPA 的查询。

3、 成绩数据的分析，以提供引导性的改进方案。

   1）班级排名查询。

   2）多次考试平均分查询。

   3）多次考试波动情况查询。

   4）一键生成考试分析报告。

4、 登录密码的更改

教师端

1、学生成绩查询。

2、添加新学生信息。

3、修改学生信息和成绩。

4、删除学生数据。

5、查看成绩分析。

   1）班级平均分。

   2）班级最高最低分。

   3）班级成绩波动情况。

6、登录密码的修改

数据安全

1、 账户登录系统。

   1） 账户类型、权限独立分配。

   2） 账户密码验证，账户、密码独立存储。

   3） 登录失败保护，多次出错自动退出。

2、 关联文件加密存储。

   1） 加密算法使 $\lim P$ 破解 $= 0$;

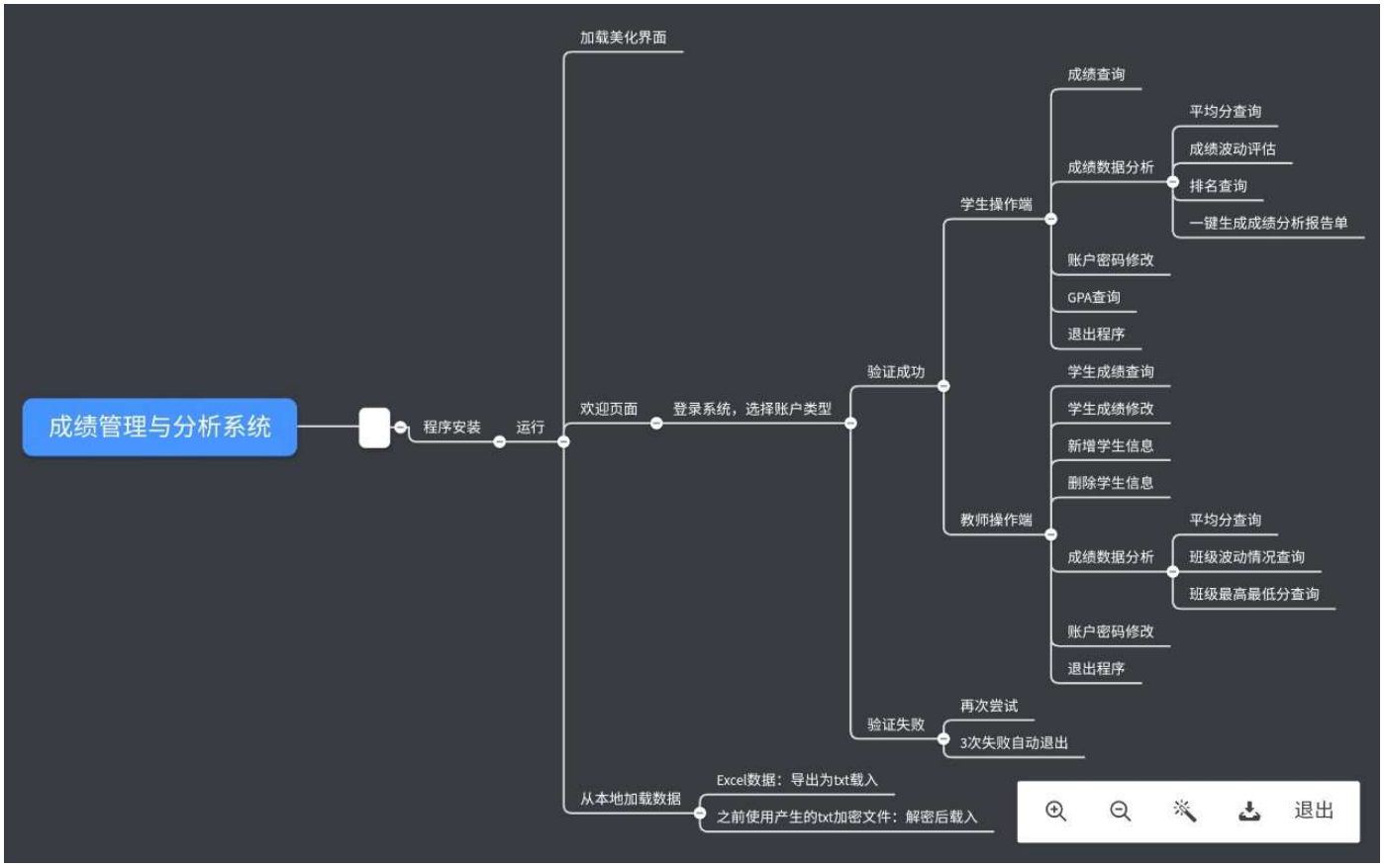
2） 双重随机处理，隐藏解密线索；
3） 使用一次性密钥，增强安全性能；
4） 每次运行随机加密，加大破解难度。

UI 设计

1、 界面简洁美观。
a) 从美学出发，带来清爽视觉体验。
b) 菜单排列简洁明了。
c) 窗口分辨率、比例适当。
d) 对称设计无处不在。
e) 无数次调试，无数次修改，只为舒适的视觉体验 。
2、 交互操作友好。
a) 从用户出发，优化细节交互方式。
b) 样式简洁，重点突出。
c) 信息丰富，既美观又实用。

三、系统设计

（一）总体设计（设计框图）



（二）模块设计

1、数据载入：实现同一电脑多次使用，和不同电脑间的转移。
   1） 第一次使用：通过 Excel 导出为 txt 文件，读取导出后的文本文件实现数据载入
   2） 多次使用：对使用后生成的加密文件进行解密操作载入数据。
2、界面美化
   1） 使操作界面简洁自然，调用 window.h 中的窗台控制函数对窗台进行自定义美化。
   2） 对每一个用户界面进行美化适配。
3、登录系统
   1） 用于选择账户类型，分配不同的操作界面和权限。
   2） 验证账号密码，识别登录用户。
4、教师端
   1） 学生成绩的查看。
   2） 学生成绩的修改。
   3） 新增学生信息。
   4） 删除学生信息。
   5） 学生信息分析。
   6） 账户密码修改。
   7） 退出保存、加密操作。
5、学生端
   1） 个人成绩查看。
   2） 个人成绩分析。
   3） 个人 GPA 查看。
   4） 账户密码修改。
   5） 退出程序。
6、成绩分析
   1） 班级、个人平均成绩统计。
   2） 班级最低分最高分统计。
   3） 班级、个人成绩波动评估。
   4） 个人成绩分析报告。
   5） 个人班级位次查询。

四、系统实现（包含模块流程图）

（零） 所有相关函数以及全局变量声明

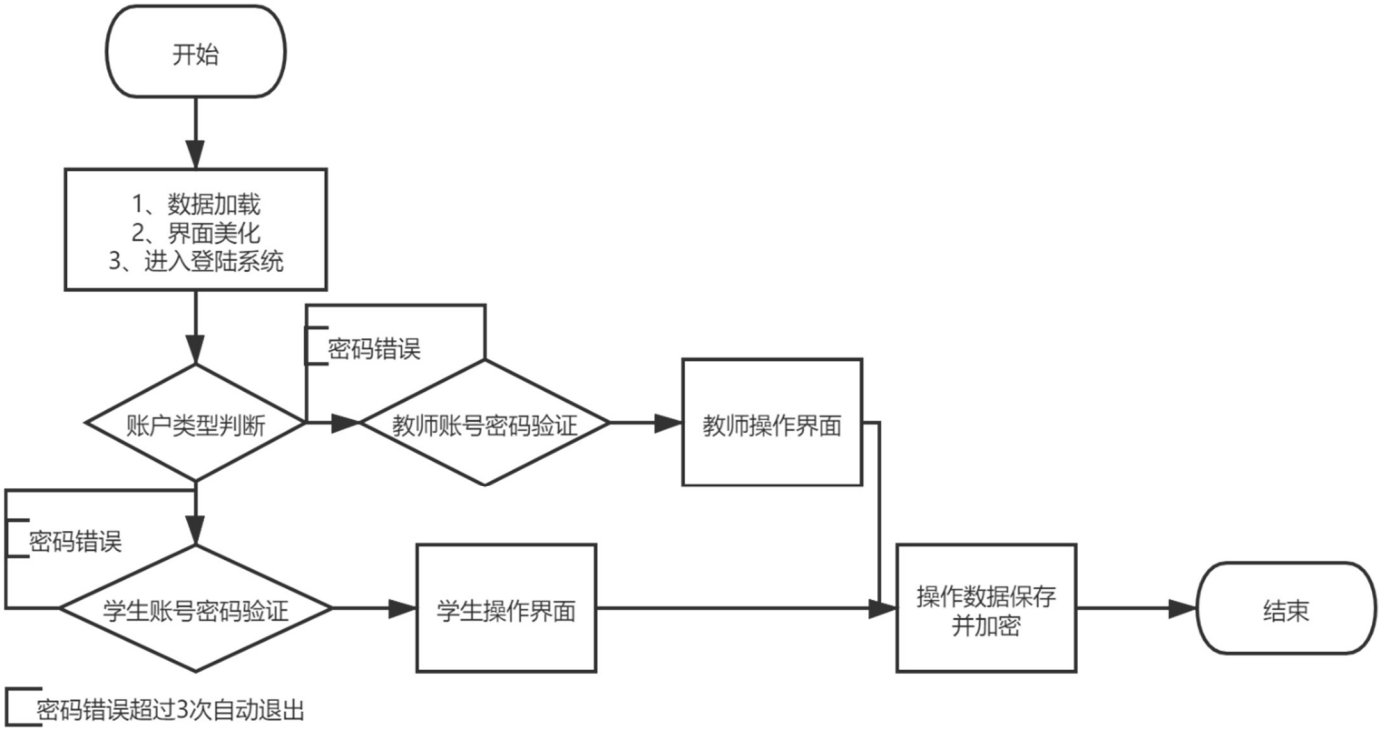
1、整个系统除了主函数外，还有八大模块：数据载入模块、窗台美化及欢迎页模块、登陆系统模块、教师菜单界面、学生菜单界面、链表功能模块、数据分析功能模块、安全加密模块。各个模块的详细设计说明分别如下。

2、 此模块定义变量类型、定义全局数据与变量、函数声明。

（一） main 函数总体实现

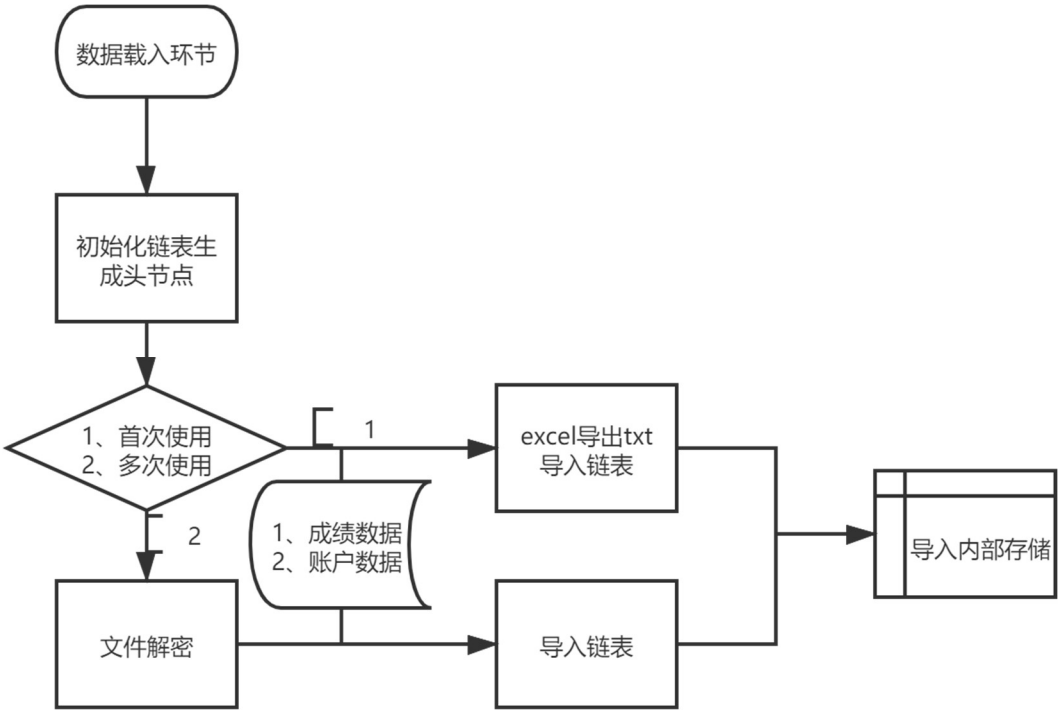
1、 利用无限次循环 while(1) ， ,和 swithch() 实现各函数的调用，系统根据输入的数字选项来调用相应的函数。包含程序初始化、变量定义、窗台美化、用户登录|菜单选择

2、 此模块流程图如下:



（二） 数据载入模块

1、此模块流程图如下:



2、相关函数:

```
1. /**
2.  * 函数名称: init_node
3.  * 函数功能: 双向链表初始化
```

```
 * 输入参数: 无
 * 输出参数: 无
 * 返 回 值: 无
**/
void init_node(void);

/**
 * 函数名称: load
 * 函数功能: 链表数据载入
 * 输入参数: 无
 * 输出参数: 无
 * 返 回 值: 无
**/
void load(void);
```

（三） 窗台美化及欢迎页模块

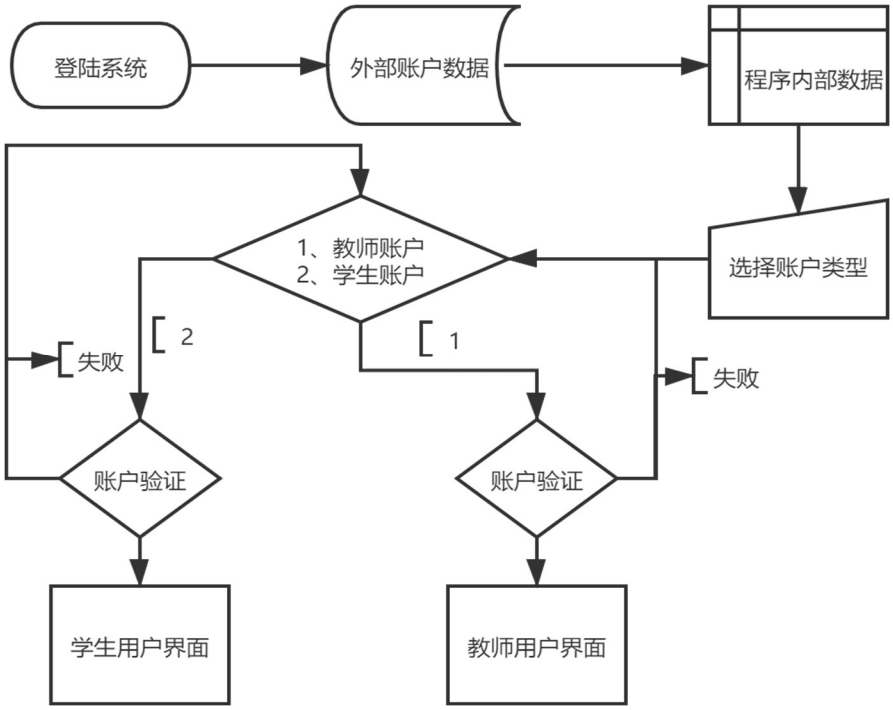
相关函数:

```
/**
 * 函数名称: surface
 * 函数功能: 窗口初始化
 * 输入参数: 无
 * 输出参数: 无
 * 返 回 值: 无
**/
void surface(void);

/**
 * 函数名称: headview
 * 函数功能: UI 美化，顶部图标
 * 输入参数: 无
 * 输出参数: 无
 * 返 回 值: 无
**/
void headview(void);

/**
 * 函数名称: SizeOfWindow
 * 函数功能: 窗口调整
 * 输入参数: HANDLE  hConsoleOutput
```

```
23. * 输出参数: 无
24.**/
25.SMALL_RECT SizeOfWindow(HANDLE hConsoleOutput);
26.
27./**
28. * 函数名称: modeset
29. * 函数功能: 设置窗口大小，为 w*h
30. * 输入参数: int w, int h
31. * 输出参数: 无
32. * 返 回 值: 无
33. * 说明: 可以定义缓冲区大小，隐藏滑动条
34.**/
35.void modeset(int w, int h);
```

（四） 登陆系统模块

1、流程图如下:



2、相关函数:

```
1. /**
2.  * 函数名称: login
3.  * 函数功能: 登录界面
```

```
4.  * 输入参数：无
5.  * 输出参数：无
6.  * 返 回 值：布尔值
7.  * 说明：密码输入错误累计达到一定次数，系统将自动关闭；switch-case 结构
8.  **/
9. bool login(void);
```

（五） 教师菜单界面

相关函数：

```
1. /**
2.  * 函数名称：awelcome
3.  * 函数功能：教师管理员菜单界面
4.  * 输入参数：无
5.  * 输出参数：无
6.  * 返 回 值：choice 选项
7.  * 说明：  switch-case 结构
8.  **/
9. int awelcome(void);
```

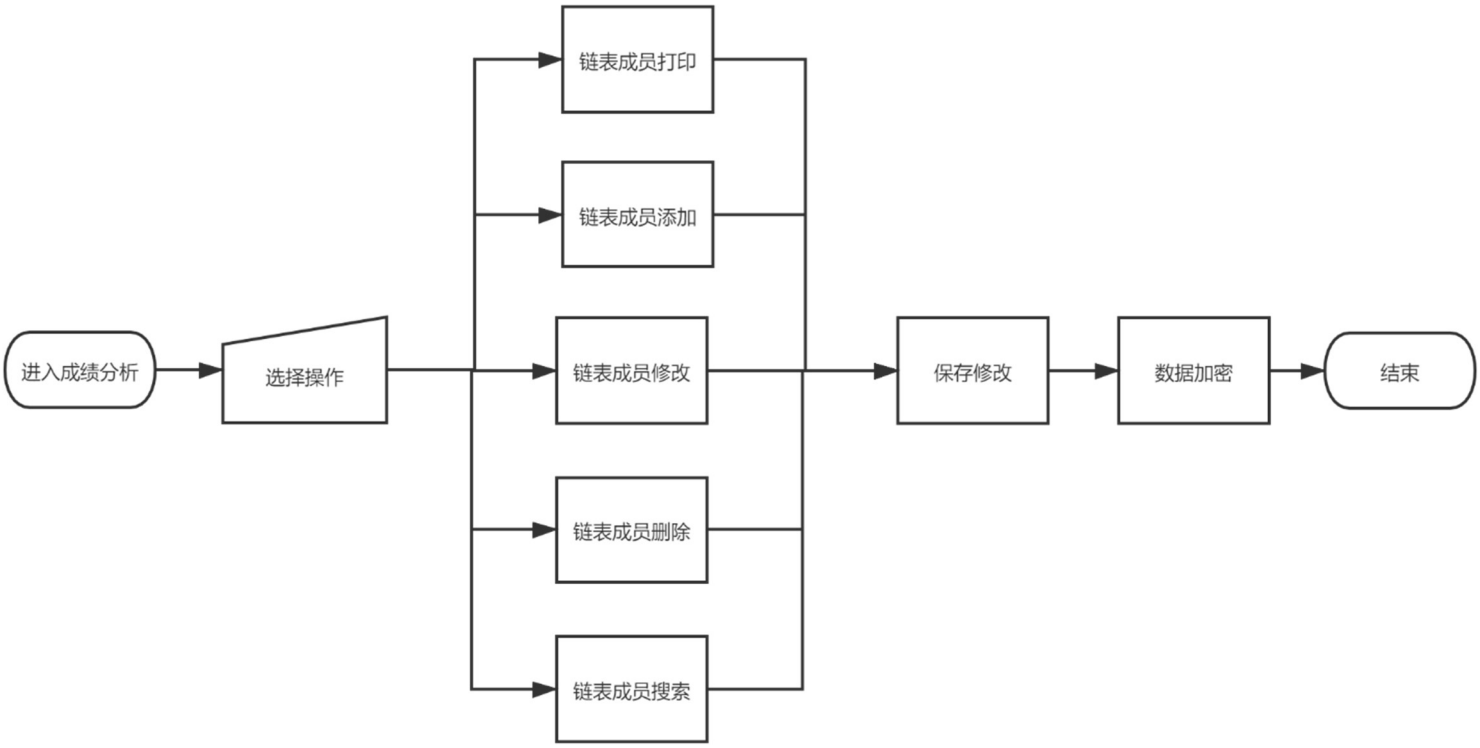
（六） 学生菜单界面

相关函数：

```
1. /**
2.  * 函数名称：swelcome
3.  * 函数功能：学生端菜单界面
4.  * 输入参数：无
5.  * 输出参数：无
6.  * 返 回 值：choice 选项
7.  * 说明：  switch-case 结构
8.  **/
9. int swelcome(void);
```

（七） 链表功能模块

1、流程图如下:



2、相关函数:

```
/**
 * 函数名称: print_one
 * 函数功能:成绩显示模块，打印成绩
 * 输入参数: Node* list
 * 输出参数: 无
 * 返 回 值: 无
 **/
void print_one(Node* list);

/**
 * 函数名称: add_student
 * 函数功能: 成员添加模块，添加成员
 * 输入参数: 无
 * 输出参数: 无
 * 返 回 值: 无
 **/
void add_student(void);

/**
 * 函数名称: write
 * 函数功能: 成绩及信息修改模块，控制台写入信息
 * 输入参数: 无
 * 输出参数: 无
```

```
24. * 返 回 值: 无
25. **/
26.void write(void);
27.
28./**
29. * 函数名称: search
30. * 函数功能: 链表搜索
31. * 输入参数: char id[MAX]
32. * 输出参数: 无
33. * 返 回 值: Node*
34. **/
35.Node* search(char id[MAX]);
36.
37./**
38. * 函数名称: delet
39. * 函数功能: 成员删除模块，删除链表节点
40. * 输入参数: Node** list, char id[MAX]
41. * 输出参数: 无
42. * 返 回 值: 无
43. * 说明: void 函数可以减少赋值一步，避免出错
44. **/
45.void  delet(Node** list, char id[MAX]);
```

3、对链表的优化

1）尽可能地简化链表的操作。例如对链表函数的定义尽量采取 void 型，而非返回指针型，让其他操作者能够简易地利用函数，不会因考虑返回值的问题而出错。

2）双向链表的引入使链表操作变得更加灵活。

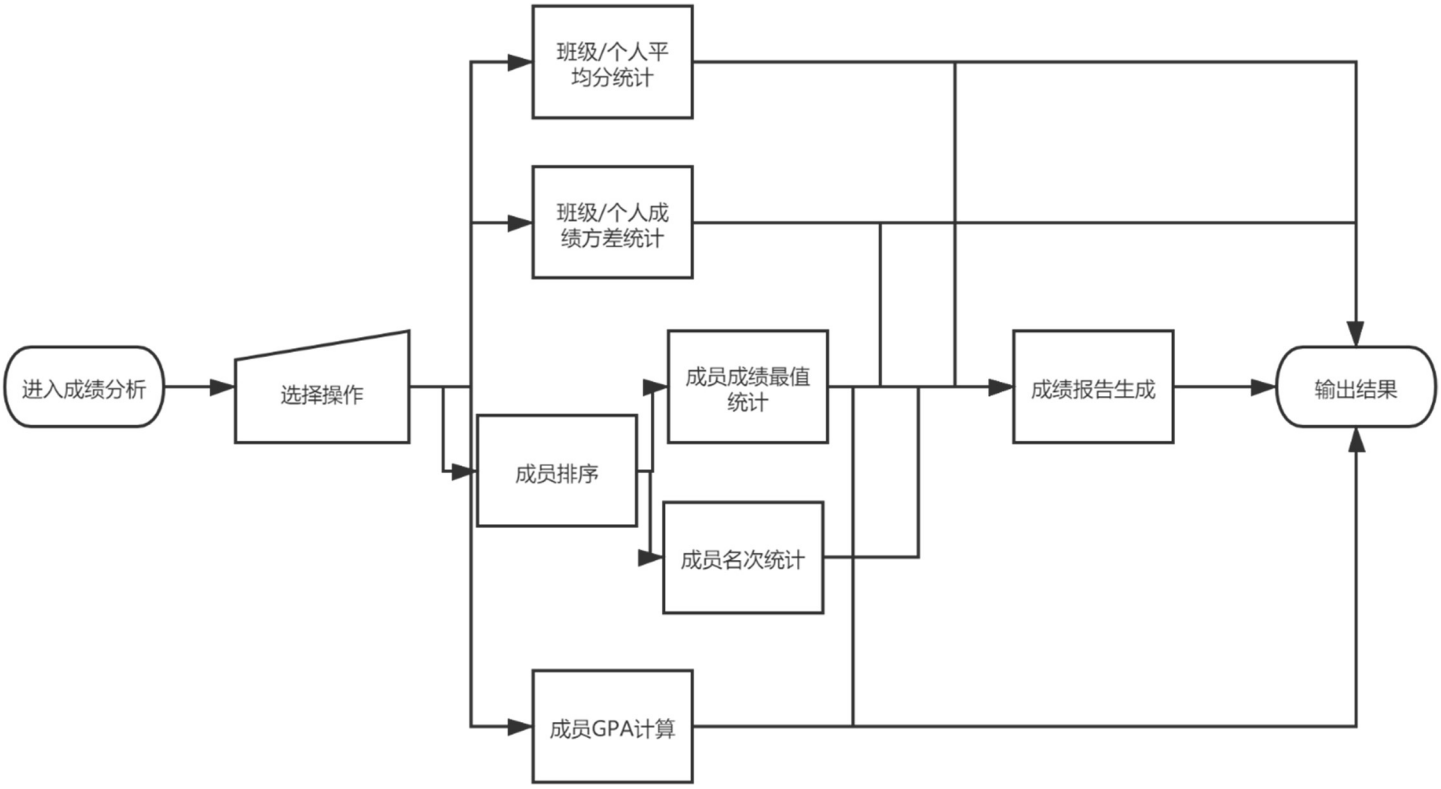
3）对于排序而言，因为可以反向输出，只需一次从高到低的排序便可以解决次序问题，反向输出便是由低到高；双向链表可以实现更多的功能。

4）对于排序而言，单向链表排序一般由冒泡排序实现，而双向链表可以实现插入排序，希尔排序等排序法。对于删除结点等功能，如果是单向链表，需要一个跟随指针记录位置，并且如果删除的是头结点还要特殊处理。更新后，不需要跟随指针，直接搞定；

5）排序，希尔排序（不常用）等排序法。（详见 main.c 源文件）

（八） 数据分析功能模块

1、流程图如下：



2、分析功能：

1）功能完备齐全，多达十二个成绩信息处理功能

A）班级平均分，个人平均分，不同次平均分一键生成。

B）3 门科目，30 位成员，30 次成绩统计完备

C）利用数学公式计算方差，对成绩的波动分析。

D）成绩管理安全，root 权限和用户权限分离，普通用户没有修改成绩的权限。

2）人性化分析，从数据中提取信息

A）GPA 绩点计算器，查询每一次考试的折算绩点。

B）一键生成成绩分析报告，根据排名计算您的成绩档位，提出相应建议。

3、相关函数：

1）链表排序算法模块：

```
//排序函数  默认降序排列
//排序，不输出
//冒泡排序：写了 2 种不同的排列方式，4 种不同排列对象。
//NULL 是为了避免读取最后一个未初始化的尾节点而将其传到链表头节点
void put_in_order(Node* phead, int code_data, int code_num)

//插入排序
//链表插入排序：只写了 1 种排序对象
void insert_sort(Node* phead)

//希尔排序（不常用）等排序法。
//归并排序 对第一次数学成绩由高到低地排序
```

```
13.Node *getMiddleNode(Node *pList)
14.
15.//合并有序链表，合并之后升序排列
16.Node *MergeList(Node *p1, Node *p2)
17..Node *MergeSort(Node *pList)
```

2）纯数据分析模块:

```
1. // 班级人数统计
2. int statistics(void);
3.
4. // 位次输出模块
5. int rank(char id[MAX]);
6.
7. //班级最值统计模块（包含了排序算法）
8. int extreme_value(Node* list, int choice, int code_num);
9.
10.//班级均分统计模块
11.// mode==0 是仅仅用于计算 avr,mode==1 用于完整版本输出 average
12.double average_all(Node* list, int choice, int code_num);
13.
14.//个人均分统计
15.double average_one(Node* list, int choice);
16.
17.//班级波动指数，班级方差
18.double variance_all(Node* phead);
19.
20.//个人波动指数，个人方差
21.double variance_one(Node* p, int choice);
22.
23.//生成成绩报告
24.void report(int code_num);
25.
26.//数据分析界面
27.int analyze(void);
28.
29.//数据分析实现函数
30.void f_analyze(int change);
31.
32.//GPA 计算函数
33.void GPA(Node* phead);
```

（九）安全加密模块

1、相关函数:

```
//下面的函数用于将数组随机化
void f_rand(int a[], int* t);
```

1）此函数可以生成伪随机数，并存储在数组 a 中。
2）同时，也生成一个-200~200 的随机数，并存储在变量 t 中。

```
//下面的函数用于加密。
//输入值：加密前的文本位置，密码位置
void encrypt(char* location_1, char* code_location);
```

1）首先，打开需要加密的文件。
2）接着，将明文中的每个字符的 ASCII 码与随机数进行运算，得到加密后的随机数，并将它保存到密文中。
3）然后，将密码加密，并保存到单独的文件中。
4）最后，为了减少文件改变带来的麻烦，删掉明文，同时，把密文的文件名改为明文。

```
//下面的函数用于解密
//输入值：After_Encrypt 的位置，密码位置
void decode(char* location_2, char* code_location);
```

1）首先打开密码和密文。
2）然后，将密文与密码进行加密时的逆运算，得到明文的 ASCII 码，进而得到明文。
3）由于每次加密使用的随机数都不同，所以，在最后删掉密码和密文，并把明文的文件名改为之前密文的文件名。
4）这样，在完成加密和解密工作后，文件的位置和名称都不会变化。

2、相关说明:
1）首先想到“凯撒加密”，但如果密文较多，就可以利用对字母的频率分析，得到字母与密码的对应关系，这样的密文就容易被破译了。比如，字母“e”平均出现的频率较高。
2）利用随机数列反破译。利用取值于 1~26 之间的整数值随机数列，使每个字母出现在密码中的概率都相等。一种理论上不可破译的密码是（用后即销毁的）一次密码本。在实际应用中，这种密码本是伪随机数列，序列中的每一个数都是 1~26 之间的整数。W 对应于伪随机数 12，就用 W 后面的第 12 个字母 l 表示 W；e 对应于伪随机数 16，就用 e 后面第 16 个字母 u 表示 e。
3）这样，再想通过分析每个字母出现的频率来破译密码就不可能了，因为在密文中每个字母出现的频率几乎相等。C 语言把字符当作小整数进行处理。为简化加密过程，我们直接对文本的 ASCII 码进行加密，并且直接将加密后的 ASCII 码和一次密码本储存在文件中。
4）同时，为了确保加密的安全性，又对该一次密码本进行了一次凯撒加密。

五、功能测试

（零）开始界面

（一） 教师端

0、 教师端界面:

（输入“用户名“ ”密码“进入。初始用户名：admin 初始密码：123456）

1、查看学生成绩:

（输入 1->输入被查询学生学号->输入想要查询第几次考试->查看->输入 h 返回主菜单）

2、新增学生信息:

（输入 2->输入要新增学生：姓名 学号 第几次考试 考试分数->输入 h 返回主菜单）

3、修改学生成绩:

（输入 3->输入要新修改学生的学号->选择”修改单项信息”还是”填入全部信息”->按提示修改->输入 h 返回主菜单）

A）如果输入 1，”填入全部信息”的界面

B）如果输入 0，”修改单项信息”的界面

4、删除学生信息:

（输入 4-> 输入要删除学生的学号->删除成功->输入 h 返回主菜单）

5、教师端成绩数据分析:

（输入 5-> 选择”全班平均成绩“还是“成绩波动指数”还是“最高最低分”->按提示输入查看->输入 h 返回主菜单）

6、教师端账户密码修改

（输入 6-> 输入新密码->修改成功->输入 h 返回主菜单）

7、退出并保存

（二）学生端

0、学生端界面:

（输入“用户名“ ”密码“进入。初始用户名: 201900800110-201900800124 共 15 个 初始密码: 学号后 6 位）

1、查看你的成绩:

（输入 1->输入想要查询第几次考试->查看->输入 h 返回主菜单）

2、成绩数据分析:

（输入 2->选择“你的平均成绩“还是”成绩波动指数“还是”你的成绩报告”还是“你的位次“->按提示键入数字->输入 h 返回主菜单）

A） 如果输入 1->“你的平均成绩“->”数学“ ”英语“ ”C 语言“ ”总分“->按提示键入数字

B） 如果输入 2->”成绩波动指数“->”数学“ ”英语“ ”C 语言“->按提示键入数字

C） 如果输入 3->”你的成绩报告”->输入想要查询第几次考试->按提示键入数字

D） 如果输入 4->“你的位次“->”数学“ ”英语“ ”C 语言“->输入想要查询第几次考试->按提示键入数字

3、账户密码修改:

（输入 3->键入新密码→修改成功>输入 h 返回主菜单）

4、 GPA 查询:

（输入 4->输入想要查询第几次考试“->输入 h 返回主菜单）

5、 退出

（输入 5）

六、总结

（一）学到了什么？

1、 在技术实践上更加熟悉了 C 语言的操作和实际应用。
2、 学会了比较熟练地使用 C 语言链表、文件进行相关操作。
3、 通过查阅资料学习了许多库函数的正确使用方法。
4、 了解了基于控制台的美化方法，学会了调用 Windows 库中的函数来实现对窗台的控制。
5、 学习了几种不同的排序算法的具体实现过程，以及应用在数组和链表上的差异。
6、 学习了基于 Git 的代码托管，便于团队的协作开发。
7、 实践了程序设计的基本流程，以及开发方式。
8、 学会了更加合理利用搜索引擎查找资料，查找手册和文档，加快了解决问题的速度。
9、 通过 debug 的过程也学会了许多代码查错的实用方法技巧。
10、学习了如何通过画流程图来辅助思考与开发。
11、锻炼了测试用例选择的能力，用例选择技巧性很强，需要经验才能考虑周全。
12、拿到一个项目，初步分析需求、划分模块，再写代码，最后完善文档。
13、意识到代码规范重要性，在开发过程中严格遵守，严谨有序简洁易懂，便于后期的维护和升级。
14、尝试对现有算法给出改进方案，比较不同算法之间的优缺点。
15、考虑增加若干基本的容错功能，如用户操作错误时程序出现错误等。

（二）痛点难点/改进之处：

1、 由于知识积累不足，遇到问题花费了大量时间查找文档和资料来解决一个又一个问题。
2、 由于经验欠缺，在设计程序的过程中给自己挖了许多坑。
3、 由于程序最初的整体构架不够优化，导致后面版本迭代时修改起来有不少困难。
4、 由于最初设计欠妥，导致代码量偏大。
5、 由于庞大的体系和众多模块，有时遇到问题时不能很快地定位错误源头。
6、 没有来得及基于开源项目做进一步开发。
7、 没有进行充分的实践测试，提供的帮助信息、失败报错的功能还不够完善，有待补充和改进。
8、 有一些实用功能没来得及实现，如：研究学校的成绩是否符合某些分布规律、哪些科目学生普遍差或者普遍优秀，并使用一些数据模型去拟合分析这个问题等。
9、 没有尝试撰写英文版本的文档。
10、遗憾没有上 SQLite，比从文件层开始实现增删改查效率高还不容易出错，但学习成本太大 QAQ
11、只实现了在 win 架构上的项目构建。

（三）如何与他人合作？

1、 结合使用场景首先讨论项目需求，明确目标，采用自顶向下、逐步求精的模块化设计思想。
2、 根据需求进行模块化设计，划分具体模块。
3、 在开发时先商量好模块间的连接方案，以便预留接口。
4、 各取所长进行任务分配，将模块分配到相应负责人，提高开发效率。
5、 各自模块预留测试单元，以便分模块测试。
6、 完成后独立测试，寻找漏洞，以便及时解决。