

（深圳）

实验报告

开课学期： 2019秋季

课程名称： 高级语言程序设计

实验名称： 学生成绩管理系统V5.0

实验性质： 设计型

实验学时： 4 地点： T2608

学生班级： 1901107

学生学号： 190110728

学生姓名： 徐文浩

评阅教师：

报告成绩：

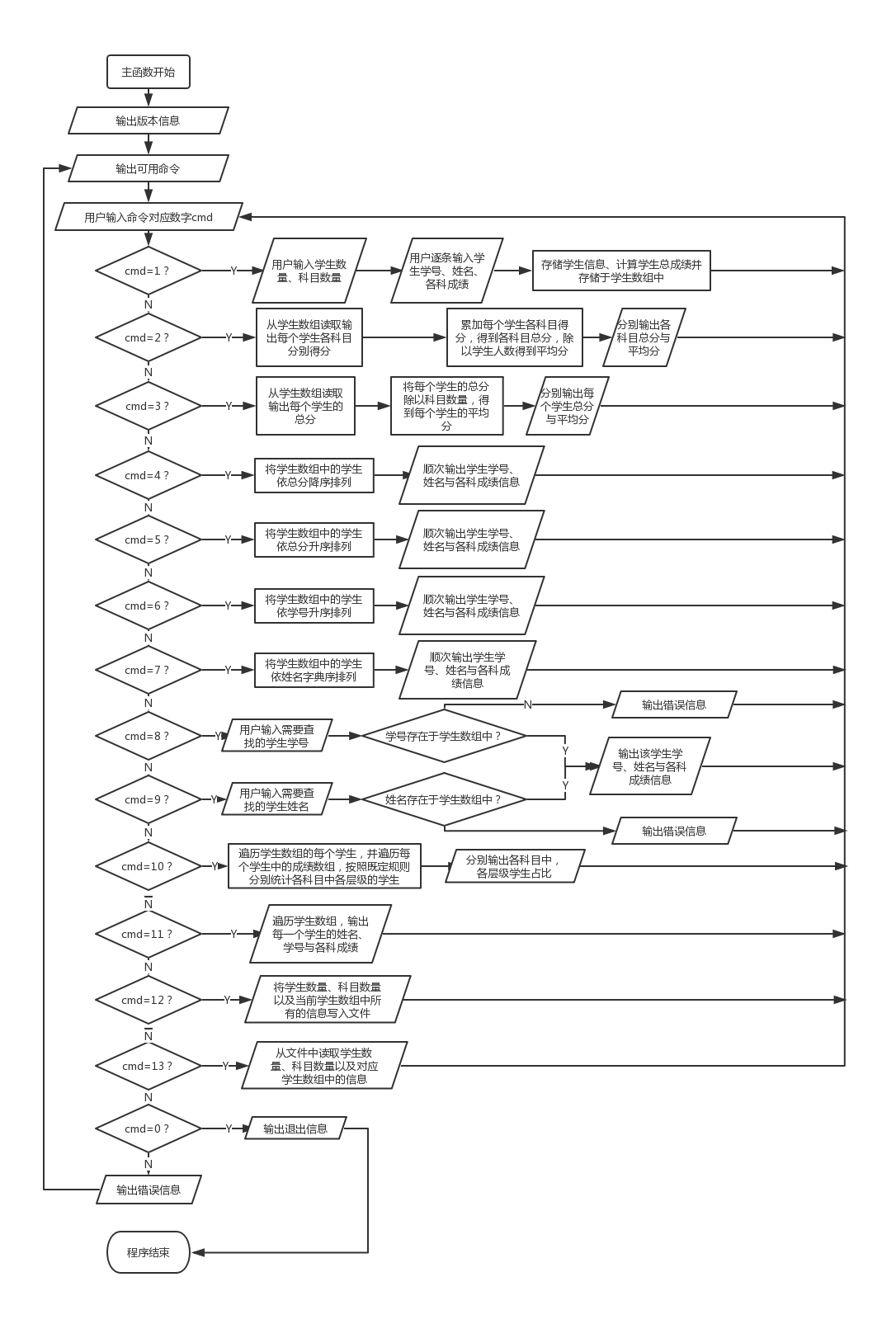
实验与创新实践教育中心制

2019年7月

1. **项目名称**

学生成绩管理系统V5.0

1. **系统设计**
2. **程序总体结构设计：**



注：程序默认读取/写入当前目录下的data.bin文件。若要改变读取/写入文件的路径，可以将文件路径作为主函数参数传入程序。

1. **数据结构：**

宏常量定义：最大学生数量为30（个），姓名数组最大长度为30（字节），最大科目数量为6（门）。

Student结构体：依次存储 {

长整型：ID，存储学生编号；

字符数组（长度为30）：Name，存储学生姓名；

长整型数组（长度为7）：Scores，首位置存储学生总分，后6个位置存储学生分科成绩；

}

1. **函数定义与功能：**
   1. **“显示提示信息”族函数**
      1. **void showinfo()**

函数依赖性：标准库函数。

功能：该函数功能为显示系统版本号以及版权信息，仅在程序启动时被调用一次。

* + 1. **void showMenu()**

函数依赖性：标准库函数。

功能：该函数功能为显示所有可用的命令。程序启动时被调用，输入不存在的命令时亦会被调用。

* 1. **“基本输入”族函数**
     1. **void inputNumber(long \*p, const char \*prompt)**

函数依赖性：标准库函数。

功能：输出提示符，并要求输入一个长整型变量。注意，若输入的数据类型不符合scanf()函数的规范，则需要重新输入直到输入成功为止。

* + 1. **void input\_c\_string(char \*sp, const char \*prompt, long len)**

函数依赖性：标准库函数。

功能：输出提示符，并要求输入一个字符串。注意，若输入的字符串大于或等于给定的字符串长度上限，则多余部分会被舍弃。

* 1. **“学生信息输入输出”族函数**
     1. **void showOneStudent\_IDScoreName(long subjectQuantity, Student Stus[], long addr)**

函数依赖性：标准库函数。

参数表：long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组；long addr：指定学生在数组中的下标。

功能：输出一名学生的姓名、学号、各科成绩、总分信息。

* + 1. **void showAll\_IDScoreName(long studentQuantity, long subjectQuantity, Student Stus[])**

函数依赖性：标准库函数；3.3.1: void showOneStudent\_IDScoreName(…)

参数表：long studentQuantity ：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组。

功能：遍历学生数组，输出所有学生的姓名、学号、各科成绩、总分信息。

* + 1. **void inputNumScoreName(long \*studentQuantity, long \*subjectQuantity, Student Stus[])**

函数依赖性：标准库函数；3.2.1: void inputNumber(…)；3.2.2:void input\_c\_string (…)

参数表：long \*studentQuantity：传入指针，便于更新外部变量“学生数量”。long \*subjectQuantity：传入指针，便于更新外部变量“科目数量”。Student Stus[]：传入学生信息数组。

功能：让用户输入学生数目、科目数目与各学生姓名、学号、各科成绩、总分，并存储在内存中。

* + 1. **void totalAverage\_Subject(long studentQuantity, long subjectQuantity, Student Stus[])**

函数依赖性：标准库函数。

参数表：long studentQuantity ：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组。

功能：通过遍历学生信息数组并分别累加相应科目的分数，再根据累加结果与学生数目输出每个科目的总分与平均分。

* + 1. **void totalAverage\_Student(int studentQuantity, int subjectQuantity, Student Stus[])**

函数依赖性：标准库函数。

参数表：long studentQuantity ：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组。

功能：遍历学生数组，读取科目数量与每个学生的总分，分别输出每个学生的总分与平均分。

* 1. **“交换与排序”族函数**
     1. **void swap(Student Stu[], int i)**

函数依赖性：标准库函数。

参数表：Student Stus[]：传入学生信息数组；int i：指定在学生信息数组需要交换的相邻两个学生中，下标较小个学生的下标。

功能：交换学生信息数组中相邻两个学生的位置，便于冒泡排序。

* + 1. **bool Score\_Descend(long \_Main[], long \_\_Main[])**

函数依赖性：标准库函数。

参数表：long \_Main[]，long \_\_Main[]：两个学生的成绩数组，总分为这两个数组的第一个元素。

功能：比较两个学生成绩数组的第一个元素（总分）大小。若前者小于后者，则返回真，否则返回假。

* + 1. **bool Score\_Ascend(long \_Main[], long \_\_Main[])**

函数依赖性：标准库函数。

参数表：long \_Main[]，long \_\_Main[]：两个学生的成绩数组，总分为这两个数组的第一个元素。

功能：比较两个学生成绩数组的第一个元素（总分）大小。若前者大于后者，则返回真，否则返回假。

* + 1. **void sortByScore(long studentQuantity, Student Stus[], bool (\*cmp\_type)(long[], long[]))**

函数依赖性：标准库函数；通过函数指针依赖于3.4.2：bool Score\_Descend(…)与3.4.3：bool Score\_Ascend。

参数表：long studentQuantity：指定学生数量； Student Stus[]Student Stus[]：传入学生信息数组；bool (\*cmp\_type)(long[], long[])：指定排序类型（降序或升序）

功能：按照总成绩，降序地或升序地排列学生信息数组。

* + 1. **void sortByID(long studentQuantity, Student Stus[])**

函数依赖性：标准库函数。

参数表：long studentQuantity：指定学生数量； Student Stus[]Student Stus[]：传入学生信息数组。

功能：按照学号，升序地排列学生信息数组。

* + 1. **void sortByName(long studentQuantity, Student Stus[])**

函数依赖性：标准库函数。

参数表：long studentQuantity：指定学生数量； Student Stus[]Student Stus[]：传入学生信息数组。

功能：按照姓名，字典序地排列学生信息数组。

* 1. **“搜索与分析”族函数**
     1. **void searchAndPrint(long studentQuantity, long subjectQuantity, Student Stus[], unsigned key)**

函数依赖性：标准库函数；3.3.1：void showOneStudent\_IDScoreName(…)；3.4.4：void sortByScore(…)。

参数表：long studentQuantity：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组；unsigned key：传入待查找学号。

功能：先将学生数组按总分降序地排列，再在学生信息数组中搜索某个学号，若有匹配，则输出第一个匹配结果的姓名、学号、各科成绩、总分信息与排名。若无匹配，则输出错误信息。

* + 1. **void searchAndPrint\_name(long studentQuantity, long subjectQuantity, Student Stus[], char key\_name[])**

函数依赖性：标准库函数；3.3.1：void showOneStudent\_IDScoreName(…)；3.4.4：void sortByScore(…)。

参数表：long studentQuantity：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组；char key\_name[]：传入待查找姓名。

功能：先将学生数组按总分降序地排列，再在学生信息数组中搜索某个姓名，若有匹配，则输出第一个匹配结果的姓名、学号、各科成绩、总分信息与排名。若无匹配，则输出错误信息。

* + 1. **void statisticAnalysis(long studentQuantity, long subjectQuantity, Student Stus[])**

函数依赖性：标准库函数。

参数表：long studentQuantity：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组。

功能：按优秀(90-100分、良好(80-89分)、中等(70-79分)、及格(60-69分)、不及格(0-59分)5个类别，对每门课程分别统计每个类别的人数以及所占的百分比。

* 1. **“文件操作”族函数**
     1. **void WriteTo (long studentQuantity, long subjectQuantity, Student Stus[], const char \*filepath)**

函数依赖性：标准库函数。

参数表：long studentQuantity：指定学生数量；long subjectQuantity：指定科目数量；Student Stus[]：传入学生信息数组；const char \*filepath：指定需要写入的文件的路径。

功能：将学生数量、科目数量写入指定文件，再将内存中的学生信息数组中所有学生的姓名、学号、各科成绩、总分信息顺次写入文件。

* + 1. **void ReadIn (long \*studentQuantity, long \*subjectQuantity, Student Stus[], const char \*filepath)**

函数依赖性：标准库函数。

参数表：long \*studentQuantity：传入指针，便于更新外部变量“学生数量”。long \*subjectQuantity：传入指针，便于更新外部变量“科目数量”。；Student Stus[]：传入学生信息数组；const char \*filepath：指定需要读取的文件的路径。

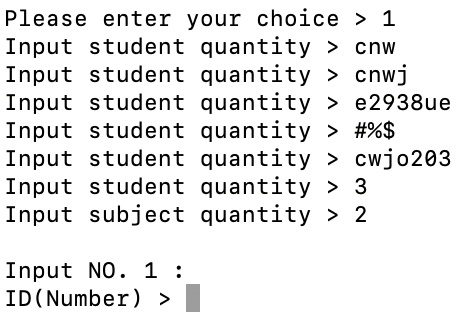
功能：从指定文件中读取学生数量、科目数量，再将文件中存储的所有学生的姓名、学号、各科成绩、总分信息顺次读取入内存中的学生信息数组中。

1. **测试数据列表及测试结果描述**

异常值输入：

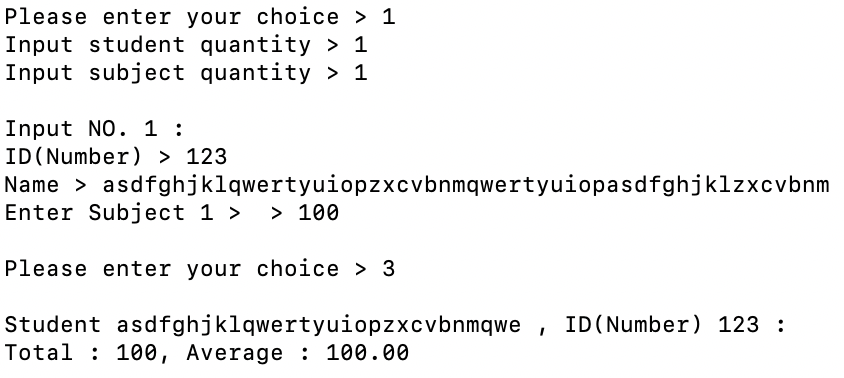
1. 要求输入数字时异常：输入非法字符会要求重新输入。

由于输入数字功能已包装为函数，则只演示一处的数字输入。

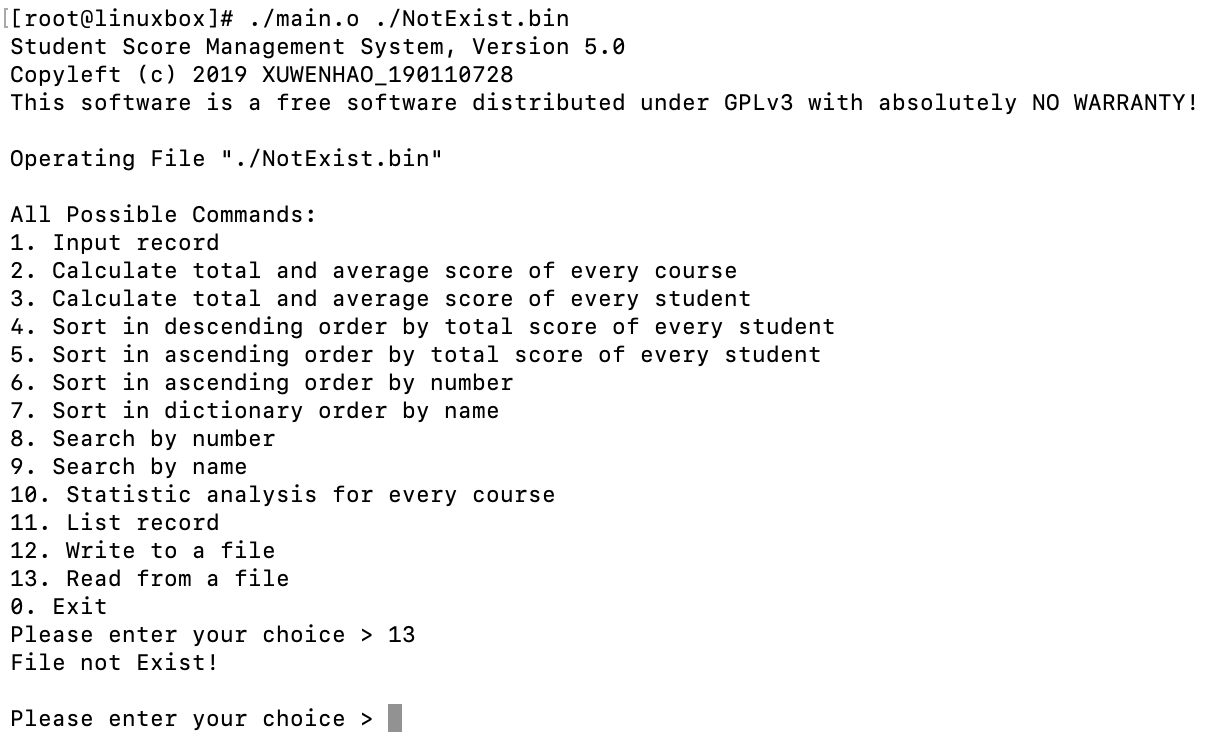


1. 要求输入字符串时越界（名字长度大于30）：自动截取字符串。

由于输入字符串功能已包装为函数，则只演示一处的字符串输入。



1. 读取文件时文件不存在：自动中止读取过程，返回操作界面。

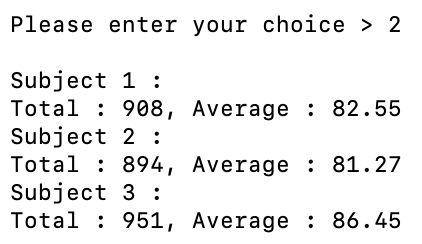


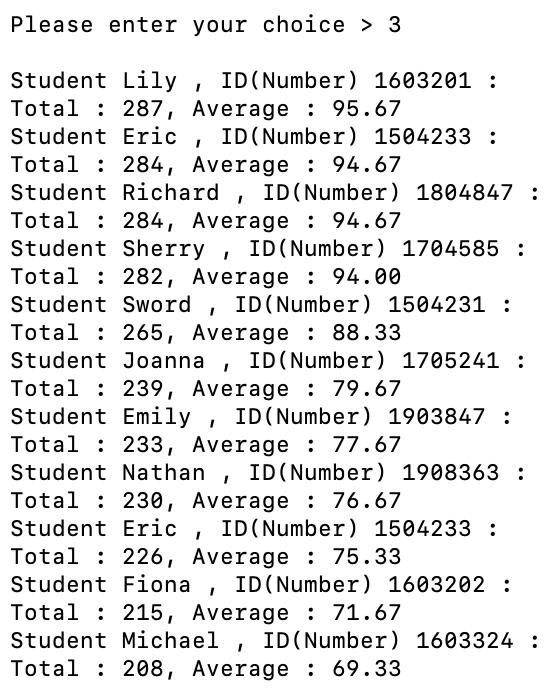
测试数据集1：（正常数据集）

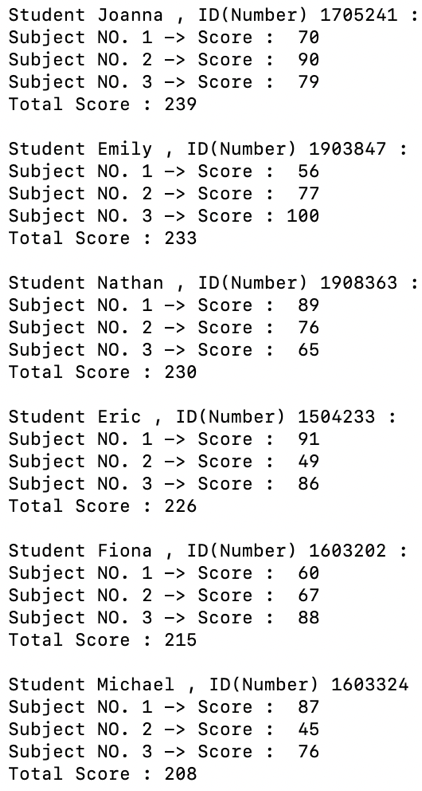
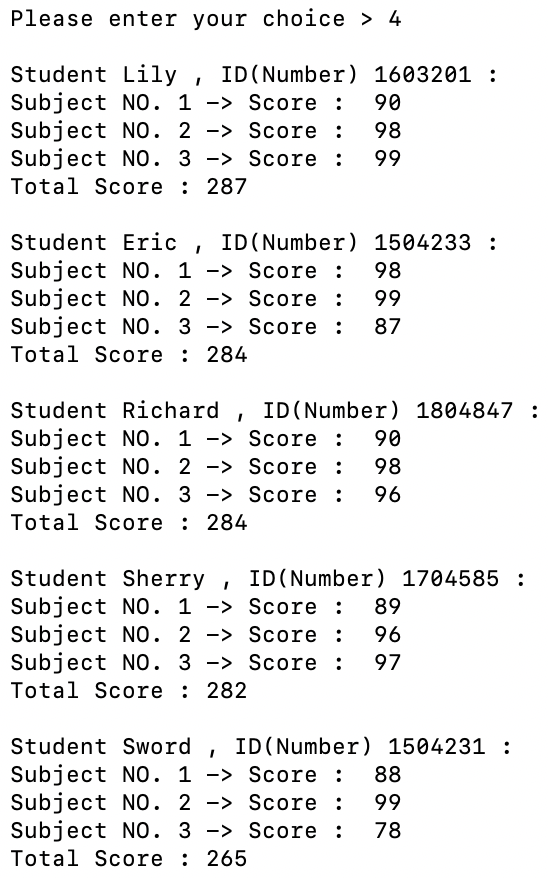
学生数目：10，科目：3。

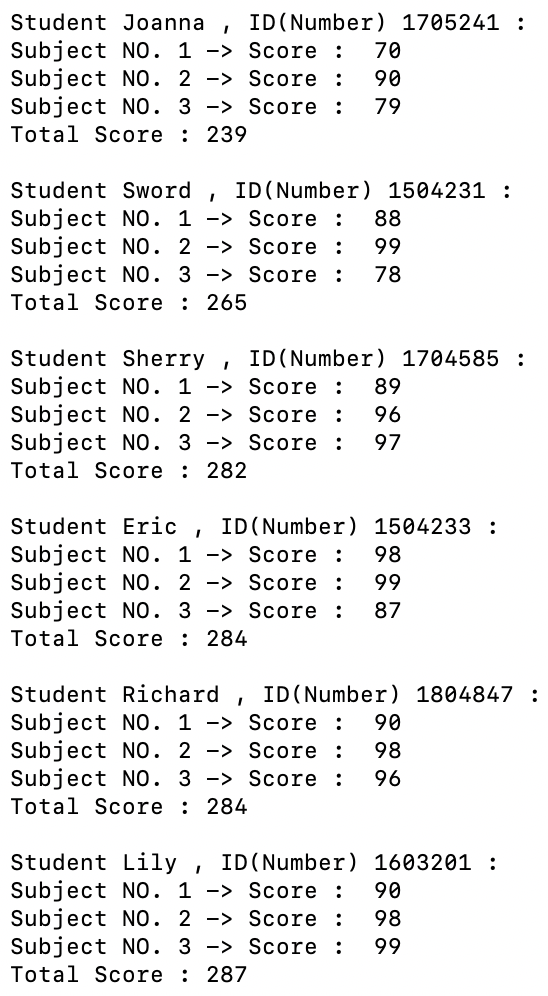
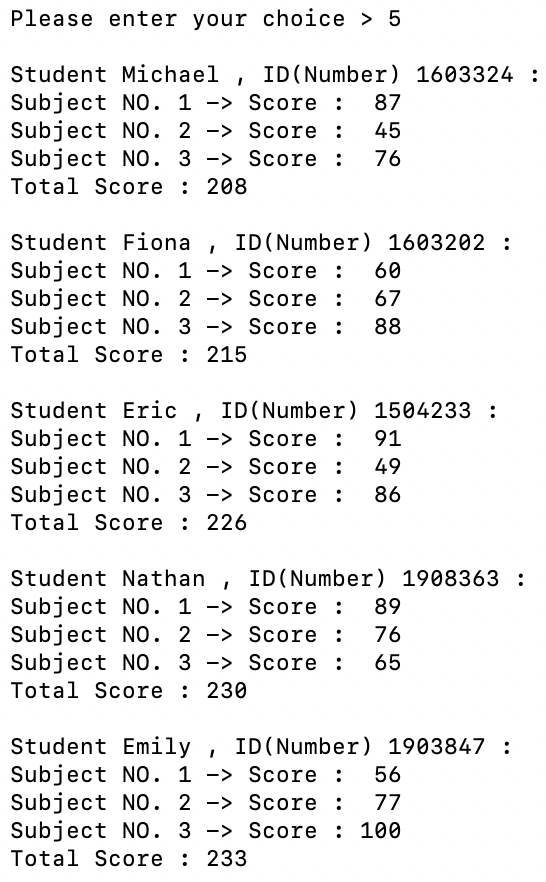
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **课程1成绩** | **课程2成绩** | **课程3成绩** |
| *1603201* | *Lily* | *90* | *98* | *99* |
| *1705241* | *Joanna* | *70* | *90* | *79* |
| *1504233* | *Eric* | *98* | *99* | *87* |
| *1603202* | *Fiona* | *60* | *67* | *88* |
| *1504231* | *Sword* | *88* | *99* | *78* |
| *1903847* | *Emily* | *56* | *77* | *100* |
| *1603324* | *Michael* | *87* | *45* | *76* |
| *1804847* | *Richard* | *90* | *98* | *96* |
| *1704585* | *Sherry* | *89* | *96* | *97* |
| *1908363* | *Nathan* | *89* | *76* | *65* |

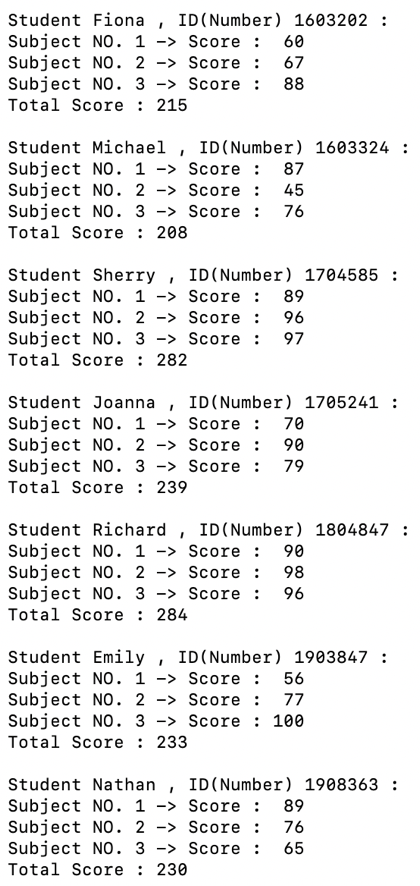
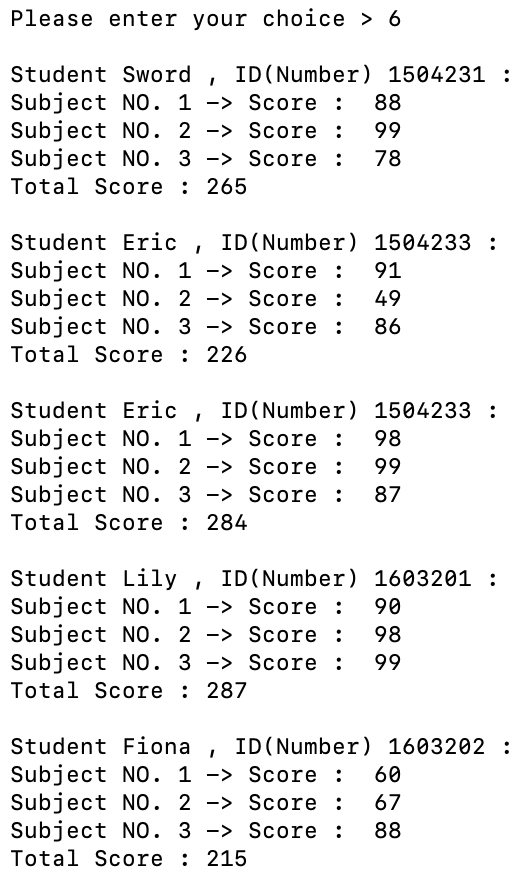
测试结果：所有功能均正常

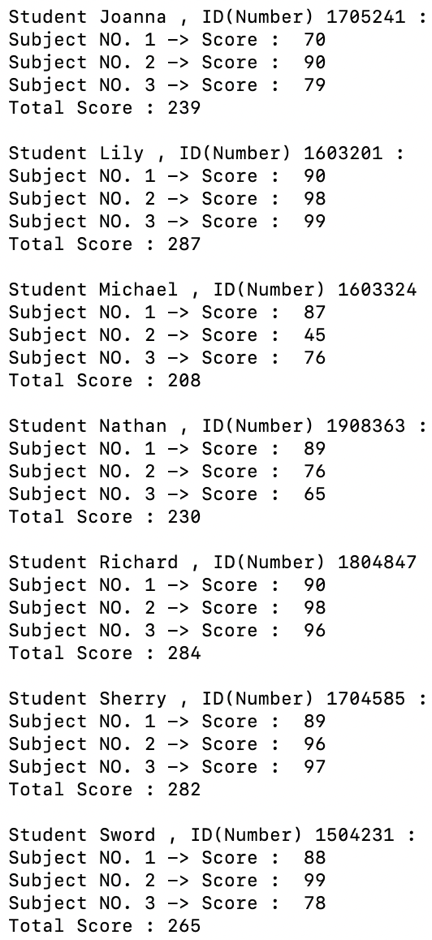
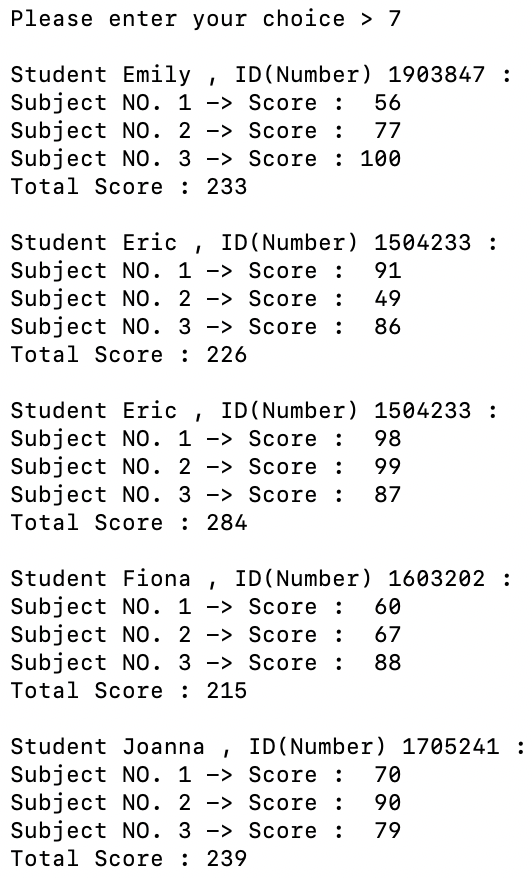


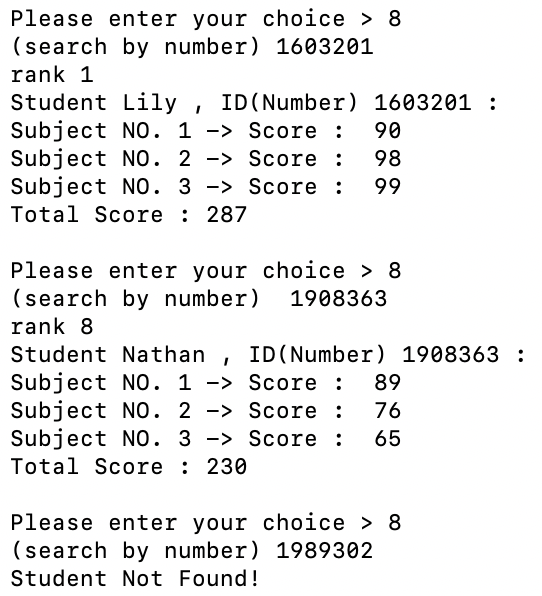


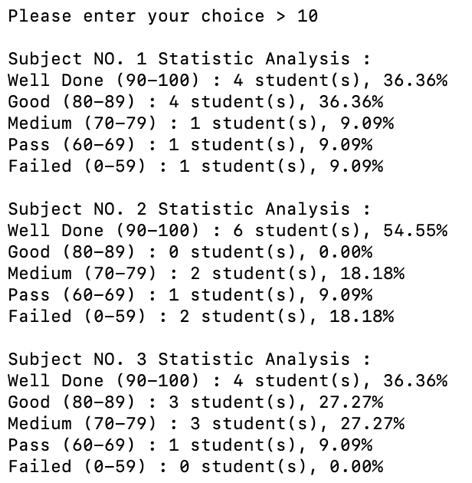
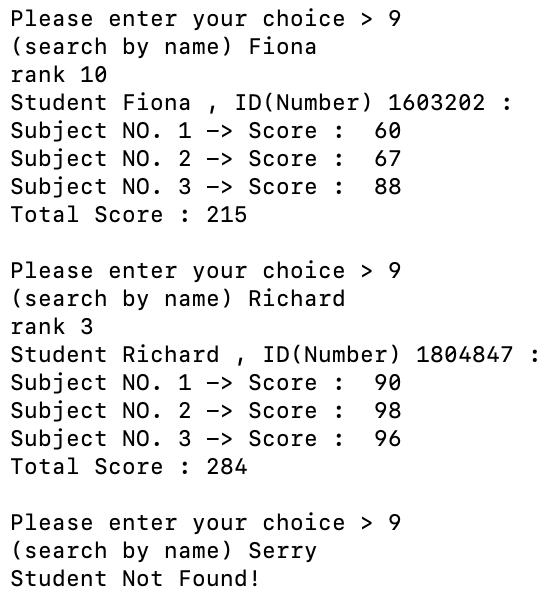


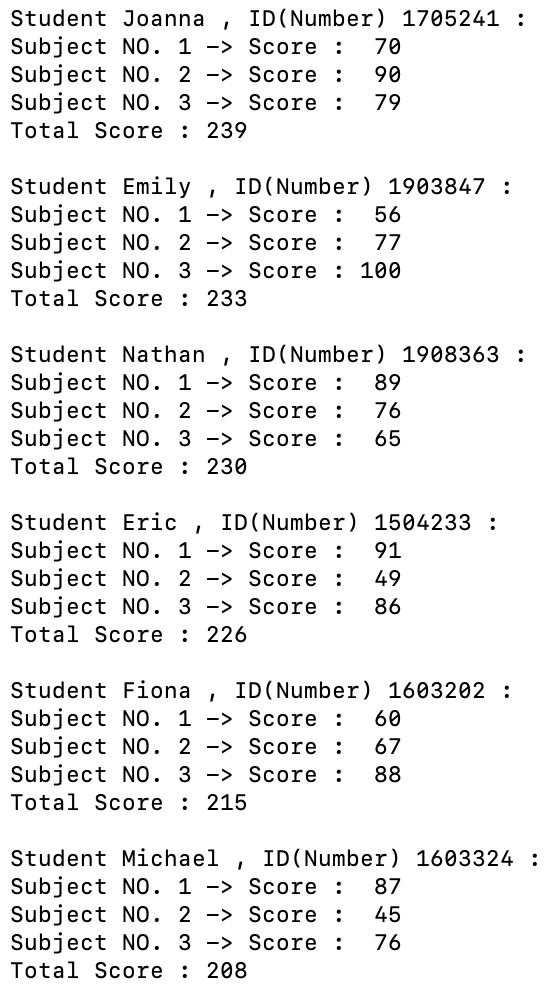
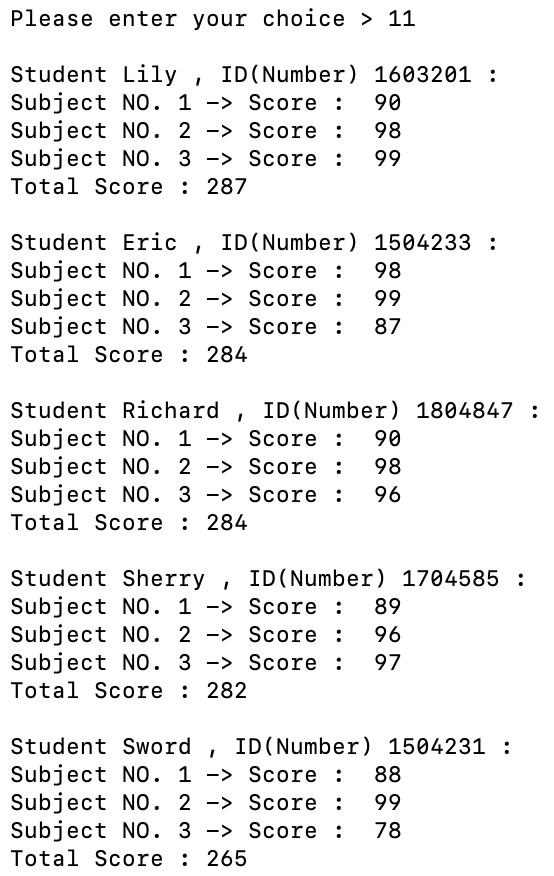


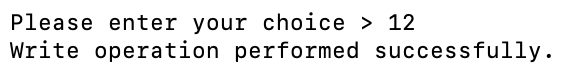


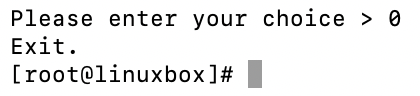


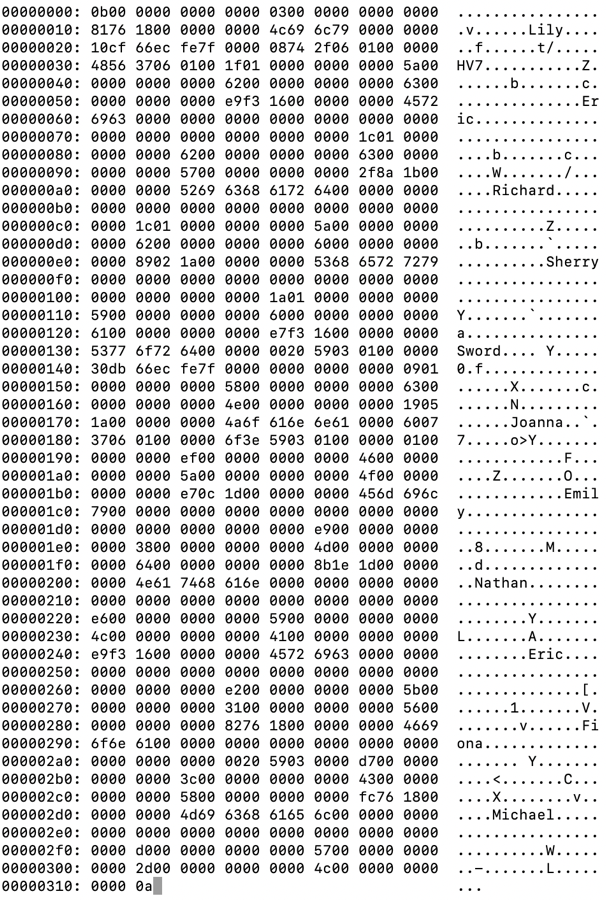
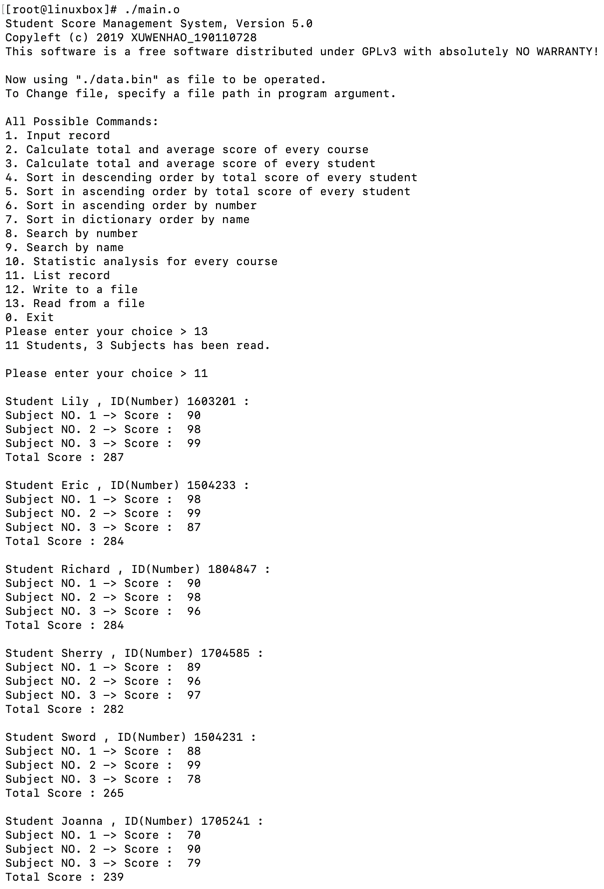










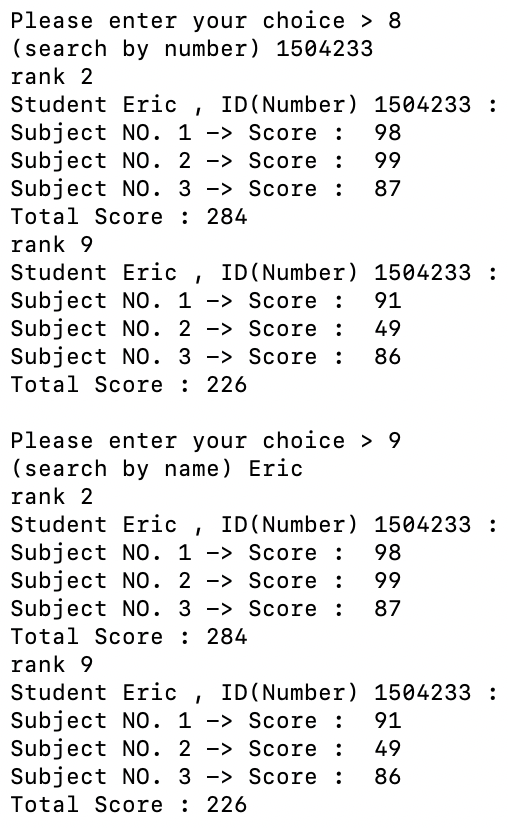
 

测试数据集2：（学号与姓名存在重复）

学生数目：11，科目：3。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **课程1成绩** | **课程2成绩** | **课程3成绩** |
| *1603201* | *Lily* | *90* | *98* | *99* |
| *1705241* | *Joanna* | *70* | *90* | *79* |
| *1504233* | *Eric* | *98* | *99* | *87* |
| *1603202* | *Fiona* | *60* | *67* | *88* |
| *1504231* | *Sword* | *88* | *99* | *78* |
| *1504233* | *Eric* | *91* | *49* | *86* |
| *1903847* | *Emily* | *56* | *77* | *100* |
| *1603324* | *Michael* | *87* | *45* | *76* |
| *1804847* | *Richard* | *90* | *98* | *96* |
| *1704585* | *Sherry* | *89* | *96* | *97* |
| *1908363* | *Nathan* | *89* | *76* | *65* |

根据学号与姓名搜索时能将学号或姓名相同的数据全部输出。

****

1. **实验课收获与总结**

通过学生成绩管理系统从V1.0到V5.0的逐步完善，了解了迭代开发、增量开发的基本原理。

通过设计该系统，掌握了设计简单应用系统的架构与数据结构的能力。

通过为该系统编写各种函数，进一步熟练了函数间调用以及传值的基本方法。

个人签名 ：

2019年 11月 1日