**高级语言程序设计实验报告**

**实验项目名称**  学生成绩管理系统 **成绩评定**

**实验项目类型：综合设计型 实验时间： 2016.12.26**

**学生姓名：王昭为 班级： 1603009 学号： 1160300903**

**学院：计算机科学与技术学院 专业： 计算机类 任课教师： 张羽**

1. **实验目的**

熟悉一维、二维数组做函数参数，排序、查找、统计分析等常用算法，自顶向下、逐步求精的模块化设计以及增量测试方法

1. **实验内容和要求**

**实验内容：**

学生成绩管理系统V7.0。某班有最多不超过30人（具体人数由键盘输入）参加某门课程的考试，用链表编程实现对学生成绩的管理。

**实验要求：**

采用自顶向下、逐步求精的模块化设计思想设计的一个小型数据库管理系统，也可以是一个游戏，或者其他作品。要求按系统的需求分析、系统设计（总体设计、各个模块的设计）、系统实现（各个模块的实现）、系统测试等过程组织报告内容，说明采用了什么数据结构和算法，遇到哪些问题，这些问题是如何解决的，本设计的亮点在哪里。

本系统主要实现下面的功能：

（1）录入每个学生的学号、姓名和各科考试成绩；

（2）计算每门课程的总分和平均分；

（3）计算每个学生的总分和平均分；

（4）按每个学生的总分由高到低排出名次表；

（5）按每个学生的总分由低到高排出名次表；

（6）按学号由小到大排出成绩表；

（7）按姓名的字典顺序排出成绩表；

（8）按学号查询学生排名及其考试成绩；

（9）按姓名查询学生排名及其考试成绩；

（10）按优秀（90~100）、良好（80~89）、中等（70~79）、及格（60~69）、不及格（0~59）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；

（11）输出每个学生的学号、姓名、各科考试成绩，以及每门课程的总分和平均分。

（12）将每个学生的纪录信息写入文件。

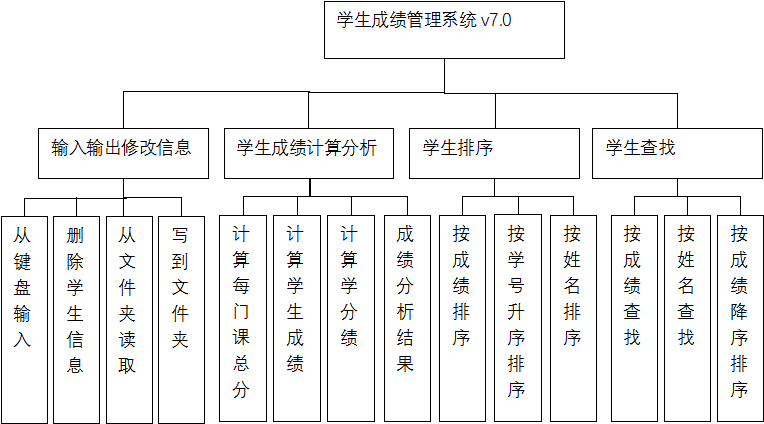
（13）从文件中读出每个学生的纪录信息并显示。

1. **实验环境**

Win10、dev c++

1. **系统设计**

**系统功能模块结构图：**

****

**算法设计：**

**直接选择排序**：第一次从R[0]~R[n-1]中选取最小值，与R[0]交换，第二次从R[1]~R[n-1]中选取最小值，与R[1]交换，....，第i次从R[i-1]~R[n-1]中选取最小值，与R[i-1]交换，.....，第n-1次从R[n-2]~R[n-1]中选取最小值，与R[n-2]交换，总共通过n-1次，得到一个按排序码从小到大排列的有序序列·

**数据结构设计：**

**结构体student（学生数据）**

|  |  |
| --- | --- |
| **变量类型** | **变量名** |
| **long** | **num** |
| **char** | **name[MAX\_LEN]** |
| **float** | **score[COURSE\_NUM]** |
| **float** | **sum** |
| **float** | **aver** |
| **float** | **creditgrade** |
| **struct student \*** | **next** |

**结构体Class**

|  |  |
| --- | --- |
| **变量类型** | **变量名** |
| **float** | **point** |

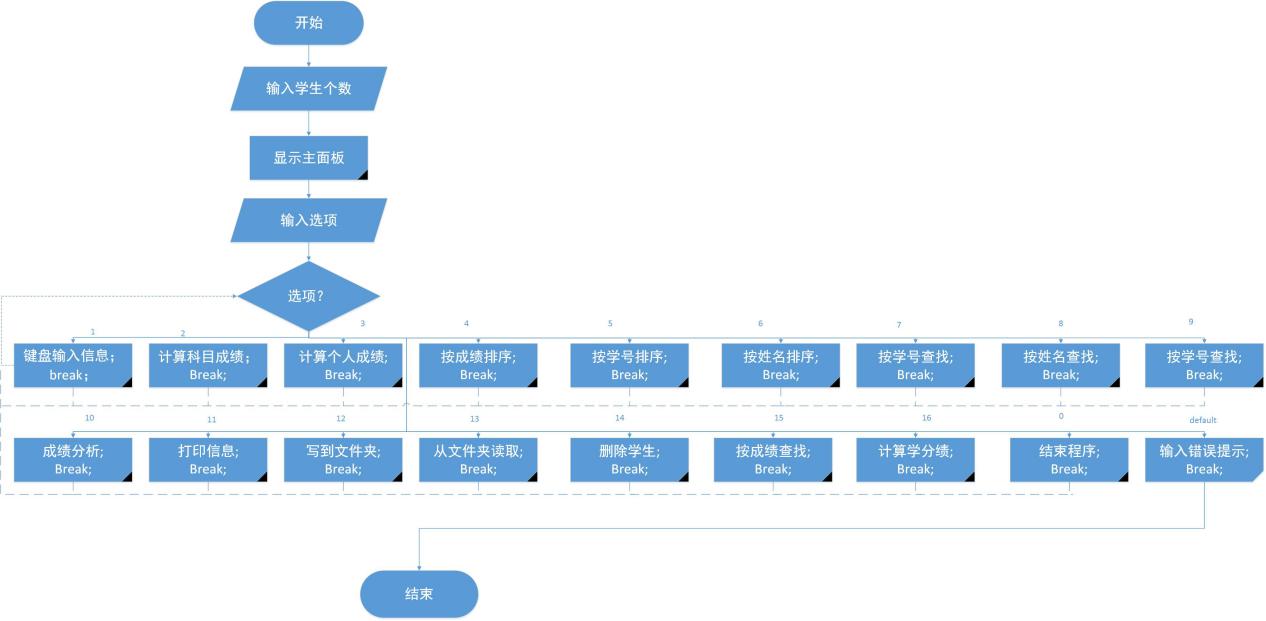
**student结构体组成的链表（储存所有学生信息）**

**各模块（函数）的功能和接口设计：**（以学生成绩管理系统V1.0为例）

**主函数：** 完成学生成绩功能

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 函数名 | 函数功能 | 函数参数 | 函数返回值 |
| 1 | swap | 交换两个节点数据域信息 | (STU \*stu1, STU \*stu2) | void |
| 2 | distribute | 分配节点 | (STU \*\*stu, int n) | void |
| 3 | ReadScore | 用于从键盘读取成绩 | (STU \*stu,int \*m) | void |
| 4 | AverSumofEveryStudent | 计算每名学生的总分 | (STU \*stu, int m) | void |
| 5 | AverSumofEveryCourse | 计算每门课的总分 | (STU \*stu, int m,int n) | void |
| 6 | SortbyScore | 按成绩排序（降序/升序） | (STU \*stu,int (\*compare)(float a,float b)) | void |
| 7 | Ascending | 比两数大小（用于函数6） | (float a, float b) | int |
| 8 | Descending | 比两数大小（用于函数6） | (float a, float b) | int |
| 9 | AsSortbyNum | 按学号升序排序 | (STU \*stu) | void |
| 10 | SortbyName | 按姓名字典顺序排序 | (STU \*stu) | void |
| 11 | SearchbyNum | 按学号查找 | (STU \*stu, int m) | void |
| 12 | SearchbyName | 按姓名查找 | (STU \*stu, int m) | void |
| 13 | StatisticAnalysis | 成绩分析 | (STU \*stu,int m) | void |
| 14 | PrintScore | 打印学生信息 | (STU \*stu,int m) | void |
| 15 | ReadfromFile | 从文件读取信息 | (STU \*stu,int \*n, int \*m) | void |
| 16 | WritetoFile | 将信息写到文件 | (STU \*stu,int n,int m) | void |
| 17 | Delete | 删除学生信息 | (STU \*head) | STU \* |
| 18 | SearchInaInterval | 按成绩查找 | (STU \*stu,int m) | void |
| 19 | CreditGrade | 学分绩计算、排名 | (STU \*stu,int m) | void |

1. **系统实现**

**主要模块的程序流程图：**

**附件有该图**

**源程序的全部代码：**

**/\*本程序特色：算学分绩并按学分绩排序，查找成绩区间上的所有同学，删除学生\*/**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**#define MAX\_LEN 10 /\* 字符串最大长度 \*/**

**#define STU\_NUM 30 /\* 最多的学生人数 \*/**

**#define COURSE\_NUM 6 /\* 最多的考试科目数 \*/**

**typedef struct Class**

**{**

**float point;**

**}CLASS;**

**typedef struct student**

**{**

**long num; /\* 每个学生的学号 \*/**

**char name[MAX\_LEN]; /\* 每个学生的姓名 \*/**

**float score[COURSE\_NUM]; /\* 每个学生COURSE\_NUM门功课的成绩 \*/**

**float sum; /\* 每个学生的总成绩 \*/**

**float aver;**

**float creditgrade;**

**struct student \*next; /\* 每个学生的平均成绩 \*/**

**}STU;**

**int Menu(void);**

**void swap(STU \*stu1, STU \*stu2);**

**void ReadScore(STU \*stu, int \*m);**

**void AverSumofEveryStudent(STU \*stu, int m);**

**void AverSumofEveryCourse(STU \*stu, int m,int n);**

**void SortbyScore(STU \*stu,int (\*compare)(float a,float b));**

**int Ascending(float a, float b);**

**int Descending(float a, float b);**

**void AsSortbyNum(STU \*stu);**

**void SortbyName(STU \*stu);**

**void SearchbyNum(STU \*stu,int m);**

**void SearchbyName(STU \*stu,int m);**

**void StatisticAnalysis(STU \*stu,int m);**

**void PrintScore(STU \*stu, int m);**

**void WritetoFile(STU \*stu,int n,int m);**

**void ReadfromFile(STU \*stu,int \*n,int \*m);**

**void distribute(STU \*\*stu, int n);**

**STU \*Delete(STU \*head);**

**void SearchInaInterval(STU \*stu,int m);**

**void CreditGrade(STU \*stu,int m);**

**/\*主函数\*/**

**int main()**

**{**

**int ch;**

**int n=0,m=0;**

**STU \*stu;**

**do**

**{**

**printf("Input student number(n<30):\n");**

**scanf("%d",&n);**

**fflush(stdin);**

**}while(!(n<30&&n>0));**

**distribute(&stu,n);**

**while(1)**

**{**

**ch=Menu();**

**switch(ch)**

**{**

**case 1:**

**ReadScore(stu,&m);**

**break;**

**case 2:**

**AverSumofEveryCourse(stu,m,n);**

**break;**

**case 3:**

**AverSumofEveryStudent(stu, m);**

**break;**

**case 4:**

**SortbyScore(stu,Descending);**

**printf("Sort in descending order by score:\n");**

**PrintScore(stu,m);**

**break;**

**case 5:**

**SortbyScore(stu,Ascending);**

**printf("Sort in ascending order by score:\n");**

**PrintScore(stu,m);**

**break;**

**case 6:**

**AsSortbyNum(stu);**

**printf("Sort in ascending order by number:\n");**

**PrintScore(stu,m);**

**break;**

**case 7:**

**SortbyName(stu);**

**printf("Sort in dictionary order by name:\n");**

**PrintScore(stu,m);**

**break;**

**case 8:**

**SearchbyNum(stu,m);**

**break;**

**case 9:**

**SearchbyName(stu,m);**

**break;**

**case 10:**

**StatisticAnalysis(stu,m);**

**break;**

**case 11:**

**PrintScore(stu,m);**

**break;**

**case 12:**

**WritetoFile(stu,n,m);**

**break;**

**case 13:**

**ReadfromFile( stu, &n, &m);**

**break;**

**case 14:**

**stu=Delete(stu);**

**PrintScore(stu,m);**

**break;**

**case 15:**

**SearchInaInterval(stu,m);**

**break;**

**case 16:**

**CreditGrade(stu,m);**

**break;**

**case 0:**

**printf("End of program!\n");**

**exit(0);**

**default:**

**printf("Input error!\n");**

**}**

**}**

**return 0;**

**}**

**/\*打印菜单\*/**

**int Menu(void)**

**{**

**int itemSelected;**

**printf("Management for Students' scores\n");**

**printf("1.Input record\n");**

**printf("2.Caculate total and average score and credit grade of every course\n");**

**printf("3.Caculate total and average score of every student\n");**

**printf("4.Sort in descending order by score\n");**

**printf("5.Sort in ascending order by score\n");**

**printf("6.Sort in ascending order by number\n");**

**printf("7.Sort in dictionary order by name\n");**

**printf("8.Search by number\n");**

**printf("9.Search by name\n");**

**printf("10.Statistic analysis\n");**

**printf("11.List record\n");**

**printf("12.Write to a file\n");**

**printf("13.Read from a file\n");**

**printf("14.Delete a student\n");**

**printf("15.Search student in a interval of score\n");**

**printf("16.Calculate the credit grade of every student\n");**

**printf("0.Exit\n");**

**printf("Please Input your choice:\n");**

**scanf("%d",&itemSelected);**

**return itemSelected;**

**}**

**/\*交换数据\*/**

**void swap(STU \*stu1, STU \*stu2)**

**{**

**STU temp;**

**temp = \*stu1;/\*交换节点\*/**

**\*stu1 = \*stu2;**

**\*stu2 = temp;**

**temp.next=stu1->next;/\*把指针域换回来\*/**

**stu1->next=stu2->next;**

**stu2->next=temp.next;**

**}**

**/\*申请内存\*/**

**void distribute(STU \*\*stu, int n)**

**{**

**int i;**

**\*stu=(STU \*)malloc(sizeof(STU));**

**STU \*stutail=\*stu;**

**for(i=1;i<n;i++)/\*根据学生个数申请内存\*/**

**{**

**stutail->next=(STU \*)malloc(sizeof(STU));**

**stutail=stutail->next;**

**}**

**stutail->next=NULL;**

**}**

**/\*读入数据\*/**

**void ReadScore(STU \*stu,int \*m)**

**{**

**int j;**

**do{**

**printf("Input course number(m<=%d):\n",COURSE\_NUM);**

**scanf("%d",m);**

**fflush(stdin);**

**}while(!(\*m<=6&&\*m>0));/\*确保输入数据正确\*/**

**printf("Input student's ID, name and score:\n");**

**for( ;stu!=NULL;stu=stu->next)/\*读入数据\*/**

**{**

**scanf("%ld%s",&stu->num,stu->name);**

**for(j=0;j<\*m;j++)**

**scanf("%f",&stu->score[j]);**

**fflush(stdin);**

**}**

**}**

**/\*计算每个学生得平均成绩\*/**

**void AverSumofEveryStudent(STU \*stu, int m)**

**{**

**int j;**

**for( ;stu!=NULL;stu=stu->next)**

**{**

**stu->sum=0;**

**for(j=0;j<m;j++)/\*对每个学生成绩求和\*/**

**{**

**stu->sum+=stu->score[j];**

**}**

**stu->aver=m>0?stu->sum/m:-1;**

**printf("student %d: sum=%.0f,aver=%.0f\n",stu->num,stu->sum,stu->aver);**

**}**

**}**

**/\*计算每门课得平均成绩\*/**

**void AverSumofEveryCourse(STU \*stu, int m,int n)**

**{**

**int j;**

**STU \*blank;**

**float sum[COURSE\_NUM],aver[COURSE\_NUM];**

**for(j=0;j<m;j++)/\*对每门课成绩求和\*/**

**{**

**blank=stu;**

**sum[j]=0;**

**for(;blank!=NULL;blank=blank->next)**

**{**

**sum[j]=sum[j]+blank->score[j];**

**}**

**aver[j]= n>0 ? sum[j]/n: -1;**

**printf("course %d:sum=%.0f,aver=%.0f\n",j+1,sum[j],aver[j]);**

**}**

**}**

**/\*按成绩排序\*/**

**void SortbyScore(STU \*stu,int (\*compare)(float a,float b))**

**{**

**STU \*i,\*j,\*temp;**

**for(i=stu;i->next!=NULL;i=i->next)**

**{**

**temp=i;**

**for(j=i->next;j!=NULL;j=j->next)/\*寻找最大值\*/**

**{**

**if(compare(j->sum,temp->sum))**

**temp=j;**

**}**

**if(temp!=i)/\*判断是否需要交换\*/**

**{**

**swap(i,temp);**

**}**

**}**

**}**

**/\*升序\*/**

**int Ascending(float a, float b)**

**{**

**return a<b;**

**}**

**/\*降序\*/**

**int Descending(float a, float b)**

**{**

**return a>b;**

**}**

**/\*按学号升序排序\*/**

**void AsSortbyNum(STU \*stu)**

**{**

**STU \*i,\*j,\*temp;**

**for(i=stu;i->next!=NULL;i=i->next)/\*按学号进行排序\*/**

**{**

**temp=i;**

**for(j=i->next;j!=NULL;j=j->next)/\*寻找最大值\*/**

**if(j->num<temp->num)**

**temp=j;**

**if(i!=temp)/\*判断是否需要交换\*/**

**swap(i,temp);**

**}**

**}**

**/\*按姓名排序\*/**

**void SortbyName(STU \*stu)**

**{**

**STU \*i,\*j,\*temp;**

**for(i=stu;i->next!=NULL;i=i->next)/\*按姓名排序\*/**

**{**

**temp=i;**

**for(j=i->next;j!=NULL;j=j->next)/\*寻找最大值\*/**

**if(strcmp(j->name,i->name)<0)**

**temp=j;**

**if(temp!=i)/\*判断是否需要交换\*/**

**swap(i,temp);**

**}**

**}**

**/\*按学号查找\*/**

**void SearchbyNum(STU \*stu, int m)**

**{**

**long number;**

**STU \*i;**

**int j;**

**printf("INput the number you want to search:");**

**scanf("%ld",&number);**

**for(i=stu;i!=NULL;i=i->next)**

**{**

**if(i->num==number)/\*判断是否为目标\*/**

**{**

**printf("%ld\t%s\t",i->num,i->name);**

**for(j=0;j<m;j++)**

**{**

**printf("%0.f\t",i->score[j]);**

**}**

**printf("%0.f\t%0.f\n",i->sum,i->aver);**

**return;**

**}**

**}**

**printf("Not found!\n");**

**}**

**/\*按姓名查找\*/**

**void SearchbyName(STU \*stu, int m)**

**{**

**char x[MAX\_LEN];**

**int j;**

**printf("Input the name you want to search:\n");**

**scanf("%s",x);**

**for( ;stu!=NULL;stu=stu->next)**

**{**

**if(strcmp(stu->name,x)==0)/\*判断是否是目标\*/**

**{**

**printf("%ld\t%s\t",stu->num,stu->name);**

**for(j=0;j<m;j++)**

**{**

**printf("%0.f\t",stu->score[j]);**

**}**

**printf("%0.f\t%0.f\n",stu->sum,stu->aver);**

**return;**

**}**

**}**

**printf("Not found!\n");**

**}**

**/\*数据分析\*/**

**void StatisticAnalysis(STU \*stu,int m)**

**{**

**int j,total,k,t[6];**

**STU \*i;**

**for(j=0;j<m;j++)**

**{**

**printf("For course %d:\n",j+1);**

**memset(t,0,sizeof(t));**

**for(i=stu;i!=NULL;i=i->next)**

**{**

**if(i->score[j]>=0&&i->score[j]<=60) t[0]++;**

**else if(i->score[j]<70) t[1]++;**

**else if(i->score[j]<80) t[2]++;**

**else if(i->score[j]<90) t[3]++;**

**else if(i->score[j]<100) t[4]++;**

**else if(i->score[j]==100) t[5]++;**

**}**

**for(total=0,k=0;k<6;k++)**

**{**

**total+=t[k];**

**}**

**for(k=0;k<=5;k++)**

**{**

**if(k==0)**

**printf("<60\t%d\t%.2f%%\n",t[0],t[0]\*1.0/total\*100);**

**else if(k==5)**

**printf("%d\t%d\t%.2f%%\n",100,t[4],t[4]\*1.0/total\*100);**

**else**

**printf("%d-%d\t%d\t%.2f%%\n",50+k\*10,60+k\*10,t[k],t[k]\*1.0/total\*100);**

**}**

**}**

**}**

**/\*打印函数\*/**

**void PrintScore(STU \*stu,int m)**

**{**

**int j;**

**STU \*i;**

**for(i=stu;i!=NULL;i=i->next)**

**{**

**printf("%ld\t%s\t",i->num,i->name);**

**for(j=0;j<m;j++)**

**{**

**printf("%.0f\t",i->score[j]);**

**}**

**printf("%.0f\t%.0f\n",i->sum,i->aver);**

**}**

**}**

**/\*从文件中输入\*/**

**void ReadfromFile(STU \*stu,int \*n, int \*m)**

**{**

**FILE \*fp;**

**int i,j;**

**int temp = \*n;**

**if((fp = fopen("student.txt", "r")) == NULL)**

**{**

**printf(" Fail to open student.txt \n");**

**return ;**

**}**

**else**

**{**

**fscanf(fp, "%d\t%d\n",n, m);**

**if(temp<\*n)**

**{**

**free(stu);**

**distribute(&stu, \*n);**

**}**

**for(i=0;i<\*n;i++)**

**{**

**fscanf(fp, "%ld%s\t", &stu->num, stu->name);**

**for(j=0;j<\*m;j++)**

**{**

**fscanf(fp,"%f\t", &stu->score[j]);**

**}**

**fscanf(fp, "%f\t%f\n", &stu->sum, &stu->aver);**

**stu=stu->next;**

**}**

**printf("Import Successfully!\n");**

**}**

**}**

**/\*输出到文件\*/**

**void WritetoFile(STU \*stu,int n,int m)**

**{**

**FILE \*fp;**

**int i,j;**

**if((fp = fopen("student.txt", "w"))==NULL)**

**{**

**printf("Fail to open student.txt ");**

**return;**

**}**

**else**

**{**

**fprintf(fp, "%d\t%d\n", n, m);**

**for(i=0;i<n;i++)**

**{**

**fprintf(fp,"%12ld%12s", stu->num, stu->name);**

**for(j=0;j<m;j++)**

**{**

**fprintf(fp,"%12.0f", stu->score[j]);**

**}**

**fprintf(fp,"%12.0f%12.0f\n", stu->sum, stu->aver );**

**stu=stu->next;**

**}**

**printf("Export Successfully!\n");**

**}**

**fclose(fp);**

**}**

**/\*删除学生信息\*/**

**STU \*Delete(STU \*head)**

**{**

**long num;**

**char a[5],name[10],c;**

**STU \*node1=head;**

**STU \*node2=NULL;/\*用于存储目标节点上一个节点\*/**

**scanf("%c",&c);**

**while(c!='\n')/\*清空缓存区\*/**

**{**

**scanf("%c",&c);**

**}**

**printf("Do you want to enter name or num\n");**

**gets(a);**

**if(strcmp(a,"name")==0)**

**{**

**printf("Please enter the name of the student you want to delete.\n");**

**gets(name);/\*获得删除信息\*/**

**if (head==NULL)**

**{**

**return NULL; /\*如果链表为空，返回NULL\*/**

**}**

**else/\*如果链表不为空\*/**

**{**

**if (strcmp(node1->name,name)==0) /\*如果头节点为目标\*/**

**{**

**head=head->next;**

**free(node1);**

**return head;**

**}**

**else /\*如果头结点不是目标\*/**

**{**

**while (node1!=NULL)**

**{**

**node2=node1;**

**node2=node2->next;**

**if (strcmp(node2->name,name)==0)**

**{**

**node1->next=node2->next;**

**free(node2);**

**break;**

**}**

**node1=node1->next;**

**}**

**return head;**

**}**

**}**

**}**

**else if(strcmp(a,"num")==0)/\*如果为学号删除\*/**

**{**

**printf("Please enter the num of the student you want to delete.");**

**scanf("%ld",&num);**

**if (head==NULL)**

**{**

**return NULL;**

**}**

**else**

**{**

**if (node1->num==num)**

**{**

**head=head->next;**

**free(node1);**

**return head;**

**}**

**else**

**{**

**while (node1!=NULL)**

**{**

**node2=node1;**

**node2=node2->next;**

**if (node2->num==num)**

**{**

**node1->next=node2->next;**

**free(node2);**

**break;**

**}**

**node1=node1->next;**

**}**

**return head;**

**}**

**}**

**}**

**}**

**/\*根据总成绩进行区间查找\*/**

**void SearchInaInterval(STU \*stu,int m)**

**{**

**STU \*head=stu;**

**int lower=0,upper=-1,j,i=0;/\*lower,upper分别为删除成绩的上下线\*/**

**SortbyScore(stu,Descending);**

**for( ;head!=NULL;head=head->next)/\*先求的总成绩\*/**

**{**

**head->sum=0;**

**for(j=0;j<m;j++)**

**{**

**head->sum+=head->score[j];**

**}**

**stu->aver=m>0?stu->sum/m:-1;**

**}**

**while(upper<lower)/\*保证上限比下限大，数据输入正确\*/**

**{**

**printf("Please enter the Interval upper limit and Interval lower limit(lower can be equal to upper)(sum score is %d)\n",100\*m);**

**scanf("%d%d",&upper,&lower);**

**}**

**for( ;stu!=NULL;stu=stu->next)**

**{**

**if(stu->sum<=upper&&stu->sum>=lower)**

**{**

**i++;**

**printf("%10ld%10s\t", stu->num, stu->name);**

**for(j=0;j<m;j++)**

**printf("%.0f\t", stu->score[j]);**

**printf("%.0f\t%.0f\n", stu->sum, stu->aver);**

**}**

**}**

**if(i==0)**

**printf("Not Found\n");**

**else**

**printf("%d students' score are between %d and %d",i,lower,upper);**

**}**

**/\*用于计算学分绩\*/**

**void CreditGrade(STU \*stu,int m)**

**{**

**int i,count=1,countc=0;/\*countc用于记录考试课门数\*/**

**float sumcredit=0;**

**STU \*pi,\*j,\*k,\*b;**

**pi=j=b=stu;**

**char a[MAX\_LEN],c;**

**CLASS cla[COURSE\_NUM];**

**for(i=0;i<m;i++)**

**{**

**strcpy(a,"a");/\*用于判断是否为考试课\*/**

**while(!(strcmp(a,"y")==0||strcmp(a,"Y")==0||strcmp(a,"n")==0||strcmp(a,"N")==0))**

**{**

**printf("Is the %d course a compulsory course?(input 'y'or'n')",i+1);**

**scanf(" %s",a);**

**scanf("%c",&c);**

**while(c!='\n')**

**{**

**scanf("%c",&c);**

**}**

**}**

**if(strcmp(a,"y")==0||strcmp(a,"Y")==0)/\*读入考试课对应学分\*/**

**{**

**countc++;**

**printf("Please input the credit");**

**scanf("%d",&cla[i].point);**

**sumcredit+=cla[i].point;**

**}**

**else**

**cla[i].point=0;**

**}**

**for( ;stu!=NULL;stu=stu->next)/\*计算每个学生的学分绩\*/**

**{**

**stu->creditgrade=0;**

**for(i=0;i<m;i++)**

**{**

**stu->creditgrade+=cla[i].point\*stu->score[i];**

**}**

**stu->creditgrade/=sumcredit;**

**}**

**for( ;pi!=NULL;pi=pi->next)/\*对学生学分绩排名\*/**

**{**

**k=pi;**

**for(j=pi->next;j!=NULL;j=j->next)**

**if(j->creditgrade>pi->creditgrade)**

**k=j;**

**if(k!=pi)**

**swap(pi,k);**

**}**

**if(countc!=0)**

**for( ;b!=NULL;b=b->next)/\*打印排名成绩等信息\*/**

**{**

**printf("num=%ld name=%s creditgrade=%.2f rank=%d\n",b->num,b->name,b->creditgrade,count);**

**count++;**

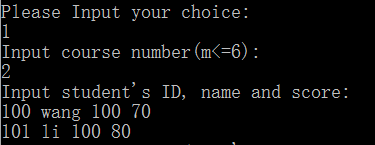
**}**

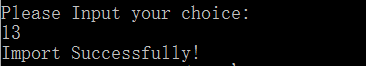
**else if(countc==0)/\*没有考试课的情况\*/**

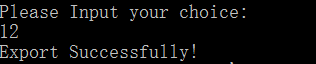
**printf("There is no compulsory course\n");**

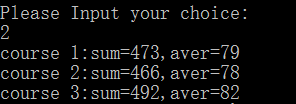
**}**

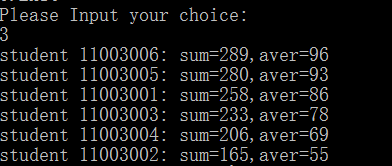
1. **实验步骤及调试过程**
2. 输入错误系统会崩溃，用while语句清空缓存区。
3. 链表申请数量错误。
4. 从文件输入会将学生名字读错，修改了读取方式。
5. **系统测试结果**

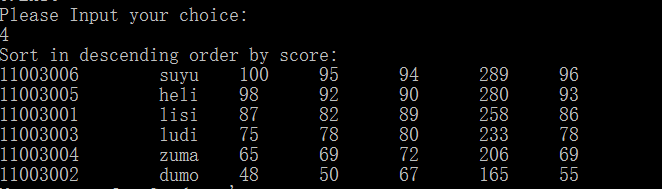
键盘输入学生信息

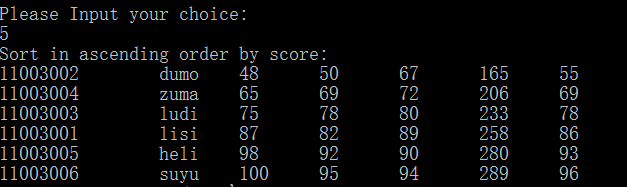
从文件输入 

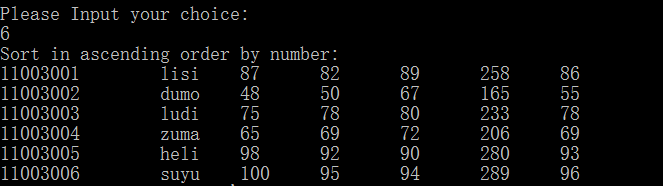
输出到文件 

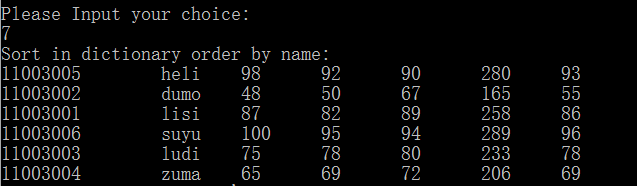
计算学科总分 

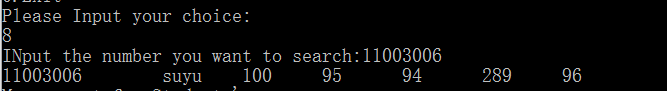
计算个人总分 

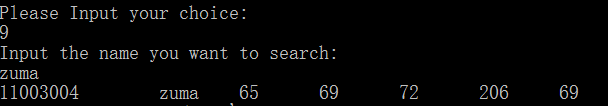
按成绩排名（降）

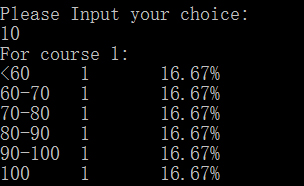
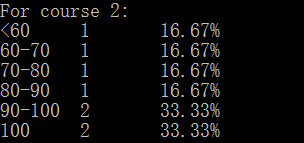
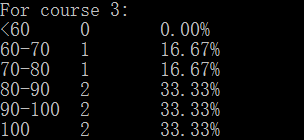
按成绩排名（升）

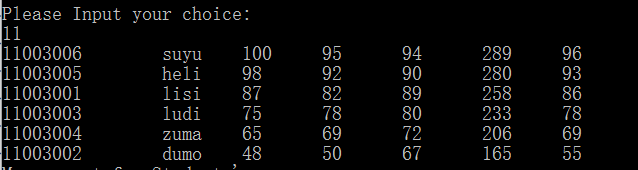
按学号排序 

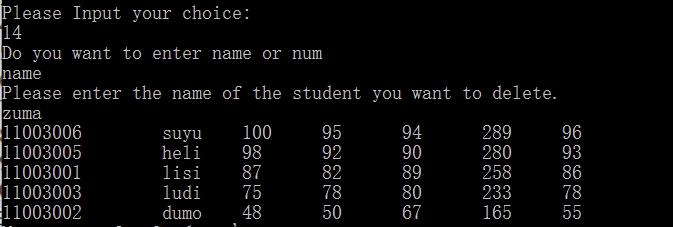
按姓名排序 

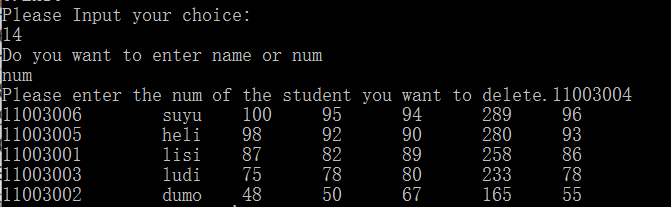
按学号查找 

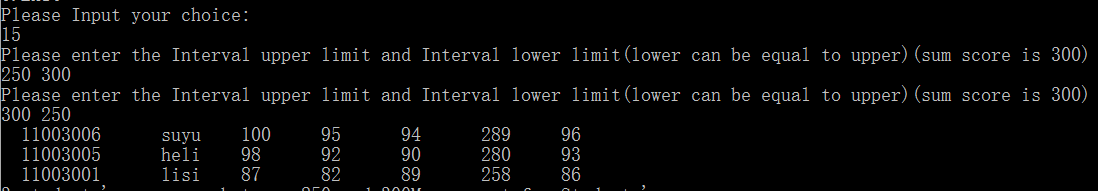
按姓名查找 

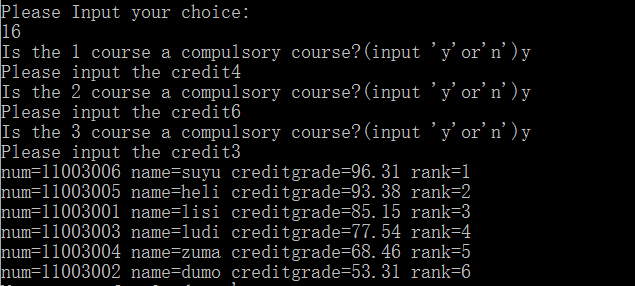
成绩分析 ****

显示成绩 

删除学生 

（姓名/学号） 

成绩查找 

学分绩计算 

1. **分析总结**

题目理解：学生管理系统的作用就是更加方便的管理学生成绩，所以应该有更详细的注释，方便使用的功能，从文件夹输入输出的功能尤为重要，该功能可以方便的进行成绩管理。

感受：较大的程序首先要分块构思，将主程序，函数的功能想好，分好，然后一个个写

逐个击破。最好将尽可能多的功能写到函数中，而不是主函数。在写代码的时候要用好 define，写完一个函数debug一次。如果在写完程序后一起debug，容易因为程序过多 不好找到。

**参考文献**

[1] 苏小红，王宇颖，孙志岗.C语言程序设计（第3版）[M].北京：高等教育出版社.2015

[2] 苏小红，王宇颖，孙志岗.C语言程序设计学习指导（第3版）[M].北京：高等教育出版社.2015