课程设计报告

1.题目要求

学生成绩管理系统

（1）增加记录：要求可以连续增加多条记录。
（2）查找：可以根据姓名（或学号）查找某个学生的课程成绩，查找某门课程成绩处于指定分数段内的学生名单等等。可以实现模糊查询，即输入名字的一部
分，可以列出满足条件的所有记录。再从这个记录中进行二次选择。
（3）删除一个学生的记录：要求可以先查找，再删除。删除前，要求用户确认。
（4）成绩修改：若输入错误可进行修改；要求可以先查找，再修改。
（5）统计分析：对某个班级学生的单科成绩进行统计，求出平均成绩；求平均成绩要求实现函数的重载，既能求单科的平均成绩，又能求三科总分的平均成绩。
求出一门课程标准差和合格率；
（6）排序功能：要求按总分进行排序（从高到低），若总分相同，则按数学排序；
若总分和数学相同，则按物理排序；若总分和各科成绩都相同，则按学号排
序；
（7）文件操作：可以打开文件，显示班级的所有学生信息；可以将增加或修改后的成绩重新写入文件；可以将排序好的信息写入新的文件；

2.需求分析

基本思路

设计一个学生的类。类的设计测试数据的记录和将要实现的功能。学生成绩

管理系统有若干学生成绩记录组成，学生一条记录包括：学号、姓名、数学

成绩、物理成绩和英语成绩。主函数显示功能菜单，供用户选择操作。每步操

作之前，都要显示菜单。在主函数中调用类的方法。

编译环境

在VS2019中运行

3.整体设计

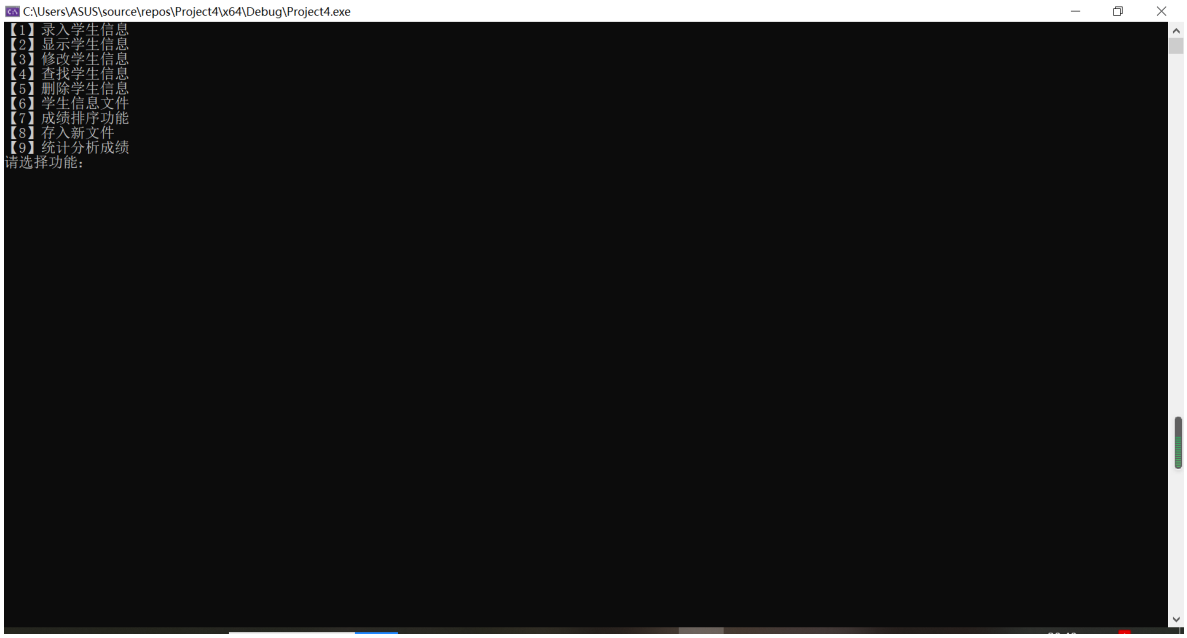
> 【1】录入学生信息
> 【2】显示学生信息
> 【3】修改学生信息
> 【4】查找学生信息
> 【5】删除学生信息
> 【6】学生信息文件
> 【7】成绩排序功能
> 【8】存入新文件
> 【9】统计分析成绩
> 请选择功能：

4.详细设计

主函数设计

```
int main()
{
    int n,i=1;
    student_system s;
    do
    {
        cout << "【1】录入学生信息" << endl;
        cout << "【2】显示学生信息" << endl;
        cout << "【3】修改学生信息" << endl;
        cout << "【4】查找学生信息" << endl;
        cout << "【5】删除学生信息" << endl;
        cout << "【6】学生信息文件" << endl;
        cout << "【7】成绩排序功能" << endl;
        cout << "【8】存入新文件" << endl;
        cout << "【9】统计分析成绩" << endl;
        cout << "请选择功能：";
        cin >> n;
        switch (n)
        {
        case 1:
            s.inputfile(); break;
        case 2:
            s.showfile(); break;
        case 3:
            s.amendfile(); break;
        case 4:
            s.findfile(); break;
        case 5:
            s.deletefile(); break;
        case 6:
            s.display(); break;
        case 7:
            s.rankfile(); break;
        case 8:
            s.storefile(); break;
        case 9:
            s.statistics(); break;
        }
        cout << "继续使用该系统请输入1，退出请输入0:";
        cin >> i;
        system ("cls");//清屏
    } while (i != 0);
    return 0;
}
```

效果如下



学生类与管理系统

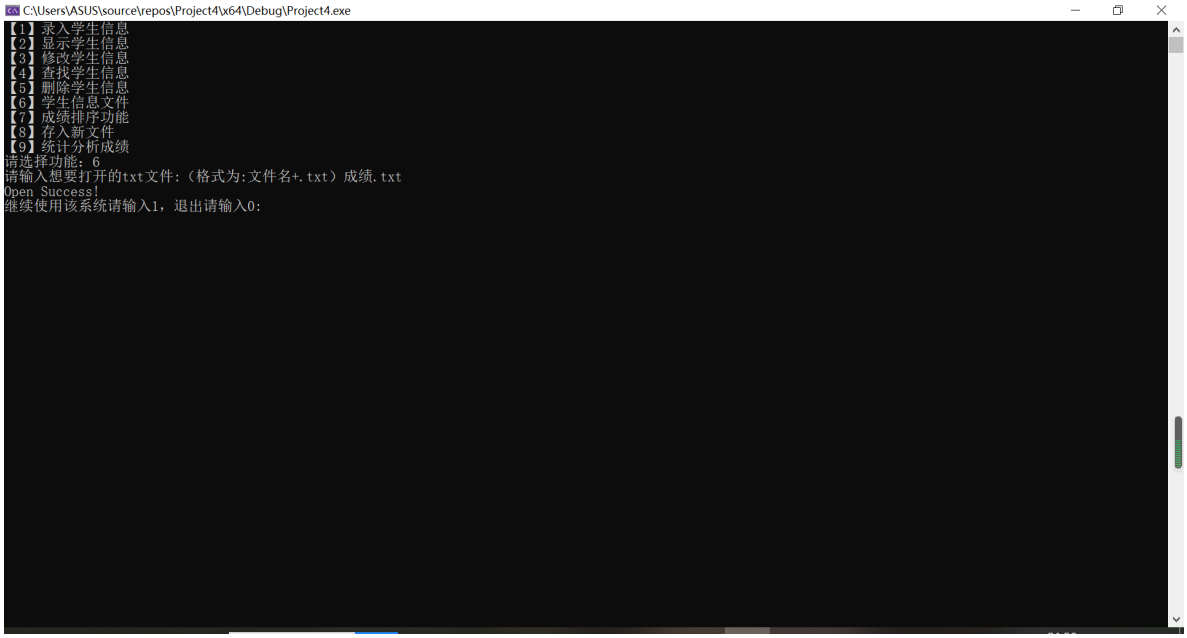
```
//学生成绩管理系统
struct student
{
    string name;
    string ID;
    float math;
    float english;
    float physics;
    float sum;
};
class student_system
{
private:
    float average=0;
    float standard=0;
    float physics_average;
    float math_average;
    float english_average;
    float physics_standard = 0;
    float math_standard = 0;
    float english_standard = 0;
    float math_num;
    float physics_num;
    float english_num;
    float num;
    student h;
    vector <student>a = {50,h};
    int j = 0;
public:
    int g=0;
    void display();//文件输入功能
    void inputfile();//信息录入
    void showfile();//显示功能
    void amendfile();//修改功能
    void findfile();//寻找功能
    void dfindfile();//单个寻找功能
```

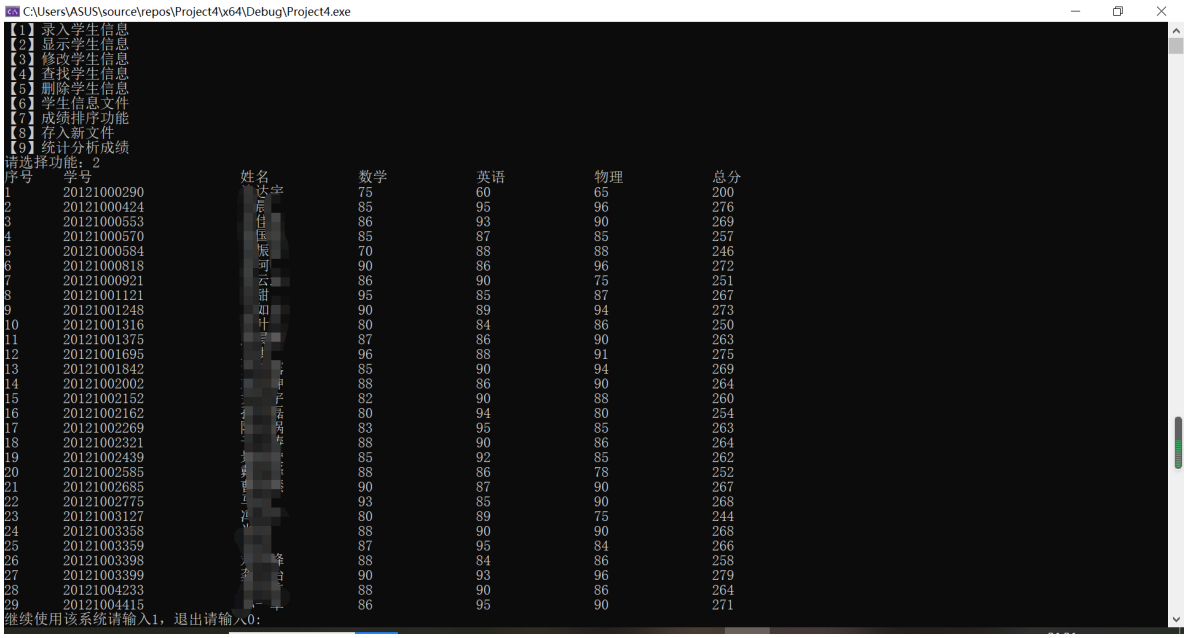
```
    void deletefile();//删除功能
    void Mfindfile();//模糊寻找
    void Ffindfile();//成绩范围寻找
    void rankfile();//排序功能
    void storefile();//文件输出存储
    void statistics();//统计分析功能
};
```

文件输入功能

```
//文件输入
void student_system::display() {
    string b[50][5];
    string filename;
    cout << "请输入想要打开的txt文件:（格式为:文件名+.txt）";
    cin >> filename;
    ifstream infile(filename.c_str(), ios::in);//读入文件信息，注意文件名字是“成绩.txt”，并且该文件要存到程序所在的文件夹中
    if (!infile) {
        cerr << "Open Error!" << endl;//只是判断是否读入成功
        return;
    }
    else
        cout << "Open Success!" << endl;
    while (infile >> a[j].ID >> a[j].name >> a[j].math >> a[j].english >> a[j].physics)
    {
        a[j].sum = a[j].math + a[j].english + a[j].physics;
        MAX++;
        j++;
    }
}
```

效果如下

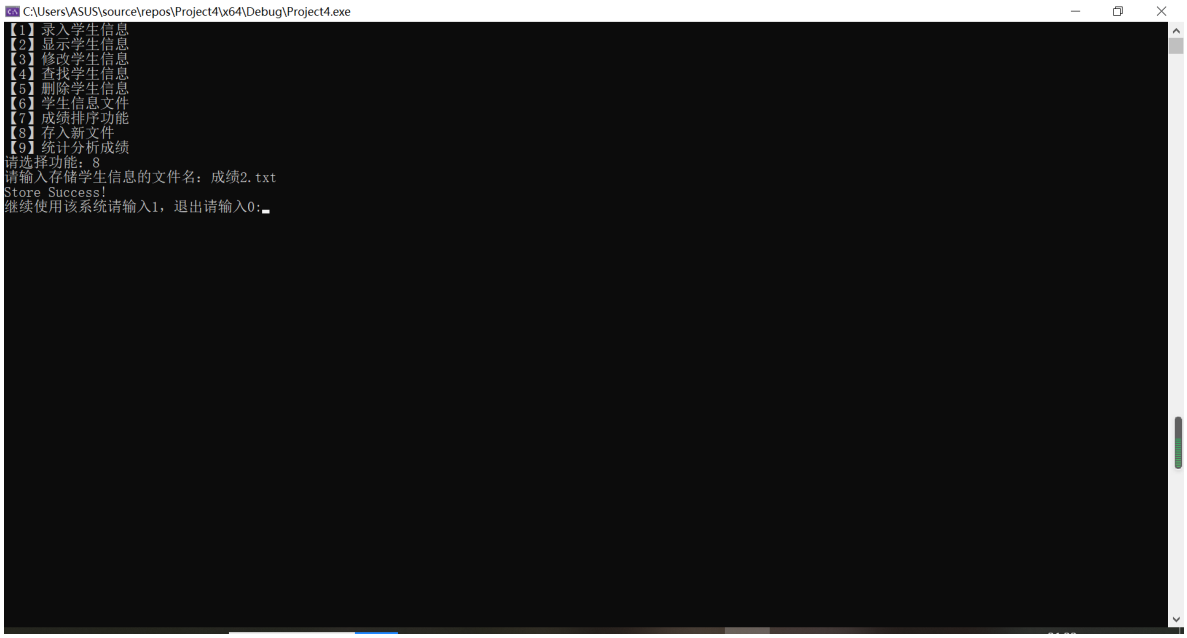


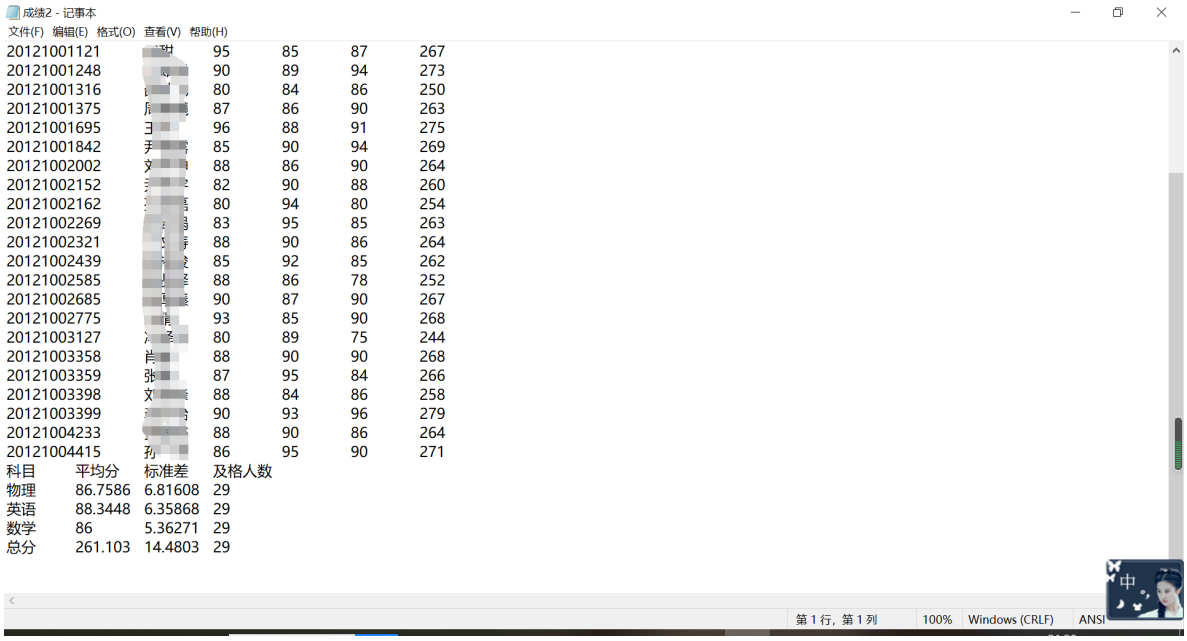


文件输出功能

```
//文件输出
void student_system::storefile() {
    string filename;
    cout << "请输入存储学生信息的文件名：";
    cin >> filename;
    ofstream outfile(filename.c_str(), ios::out);
    if (!outfile) {
        cerr << "open error!" << endl;
        exit(1);
    }
    else
        cout << "Store Success!"<<endl;
        outfile<< "学号" << "\t" << "\t" << "姓名" << "\t"  << "数学" << "\t"  <<
"英语" << "\t" << "物理" << "\t"<< "总分" << endl;
        for (int j = 0; j < MAX + g; j++) {
            outfile << a[j].ID << "\t" << a[j].name << "\t" << a[j].math << "\t"
<< a[j].english << "\t" << a[j].physics << "\t" << a[j].sum << endl;
        }
    outfile << "科目" << "\t"<< "平均分" << "\t" << "标准差" << "\t" << "及格人数"
<< endl;
    outfile << "物理" << "\t" << physics_average<<"\t" << physics_standard <<
"\t" <<physics_num<< endl;
    outfile << "英语" << "\t" << english_average << "\t" << english_standard <<
"\t" << english_num << endl;
    outfile << "数学" << "\t" << math_average << "\t" << math_standard << "\t" <<
math_num << endl;
    outfile << "总分" << "\t" << average << "\t" << standard << "\t" << num <<
endl;
    outfile.close();
}
```

效果如下



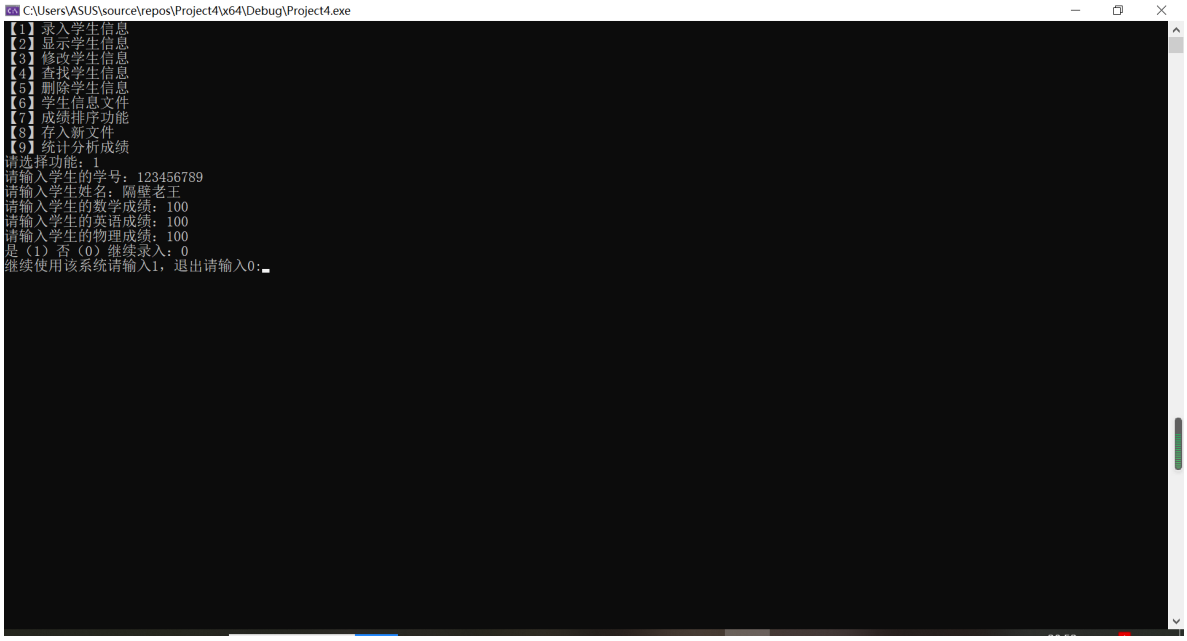


录入学生信息

```
void student_system::inputfile()
{
    int o=1;
    j = MAX+g;
    while (o!=0)
    {
        cout << "请输入学生的学号：";
        cin >> a[j].ID;
        cout << "请输入学生姓名：";
        cin >> a[j].name;
        cout << "请输入学生的数学成绩：";
        cin >> a[j].math;
        cout << "请输入学生的英语成绩：";
        cin >> a[j].english;
        cout << "请输入学生的物理成绩：";
        cin >> a[j].physics;
        a[j].sum = a[j].math + a[j].english + a[j].physics;
        cout << "是（1）否（0）继续录入：";
```

```
        cin >> o;
        j++; g++;
    }
}
```

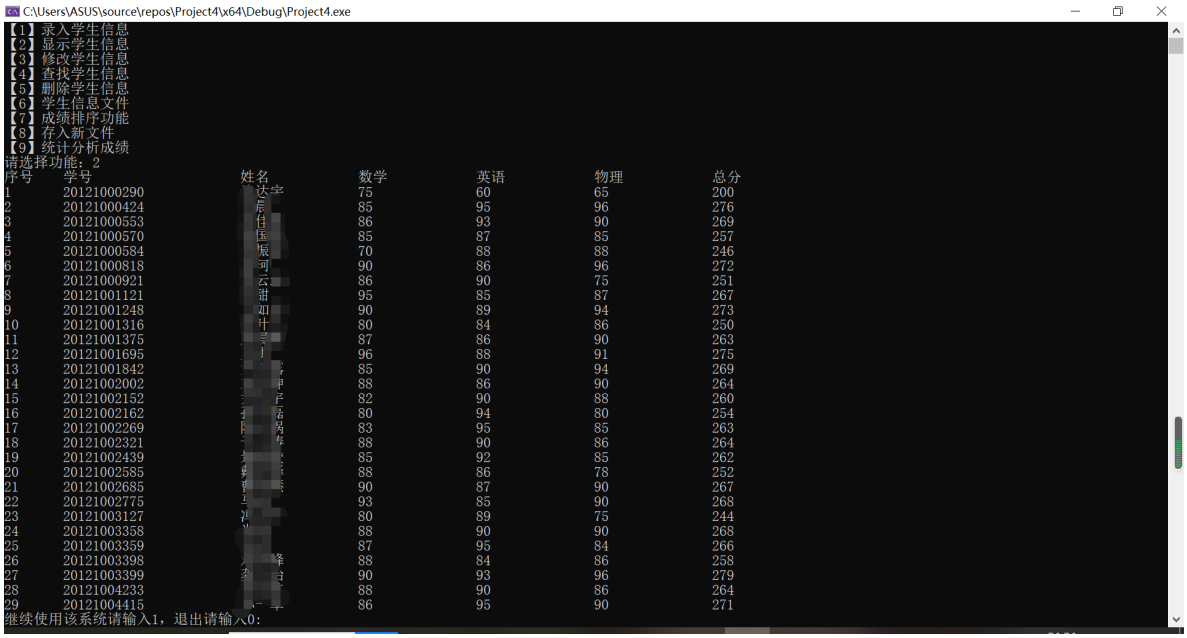
效果如下



显示学生信息

```
//显示学生信息
void student_system::showfile()
{
    cout << "序号" << "\t" << "学号" << "\t" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t" << "\t" << "英语" << "\t" "\t" << "物理" << "\t""\t" << "总分" << endl;
    for (j = 0; j<MAX+g; j++)
    {
        a[j].sum = a[j].math + a[j].english + a[j].physics;
        cout << j + 1 << "\t" << a[j].ID << "\t" << "\t" << a[j].name << "\t" "\t" << a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t" << a[j].sum << endl;
    }
}
```

效果如下

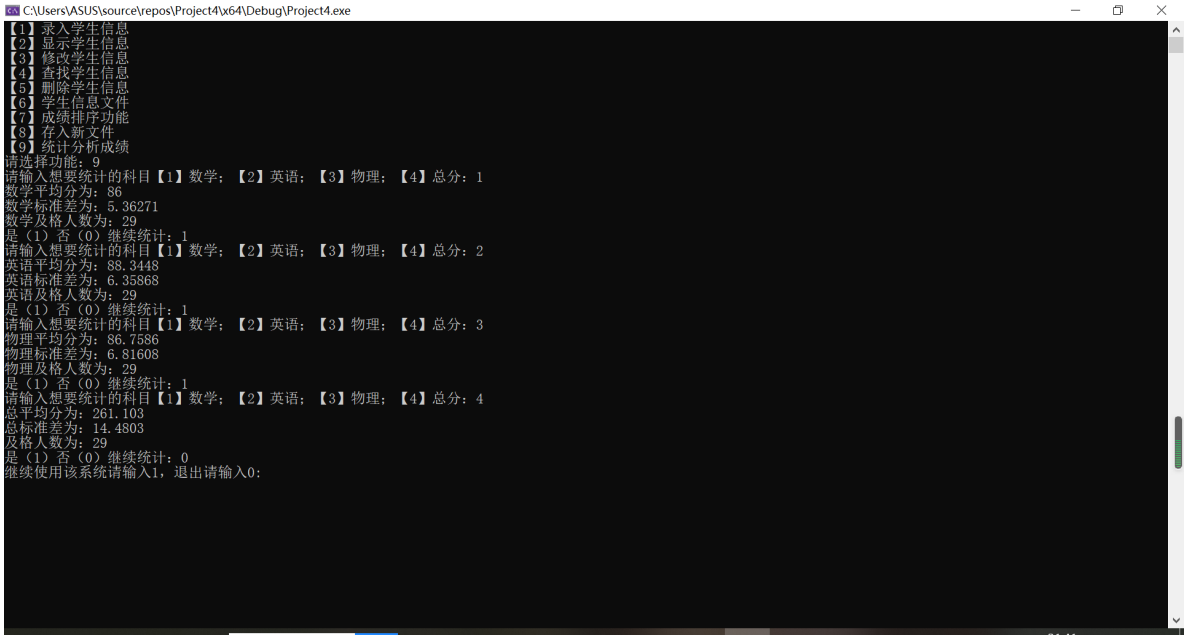


统计功能

```
void student_system::statistics() {
    int n;
    do
    {
        cout << "请输入想要统计的科目【1】数学；【2】英语；【3】物理；【4】总分：";
        cin >> n;
        switch (n) {
        case 1:
            math_num = 0; math_standard = 0; math_average = 0;
            for (int i = 0; i < MAX + g; i++)
                math_average += a[i].math;
            math_average /= (MAX + g);
            for (int i = 0; i < MAX + g; i++)
                math_standard += (a[i].math - math_average)*(a[i].math -
math_average);
            math_standard = sqrt(math_standard / (MAX + g));
            for (int i = 0; i < MAX + g; i++)
            {
                if (a[i].math >= 60)
                    math_num++;
                else
                    continue;
            }
            cout << "数学平均分为：" << math_average << endl;
            cout<< "数学标准差为：" << math_standard << endl;
            cout << "数学及格人数为：" << math_num << endl;
             break;
        case 2:
            english_num = 0; english_standard = 0; english_average = 0;
            for (int i = 0; i < MAX + g; i++)
                english_average += a[i].english;
            english_average /= (MAX + g);
            for (int i = 0; i < MAX + g; i++)
                english_standard += (a[i].english - english_average) *
(a[i].english - english_average);
            english_standard = sqrt(english_standard / (MAX + g));
```

```
            for (int i = 0; i < MAX + g; i++)
            {
                if (a[i].english >= 60)
                    english_num++;
                else
                    continue;
            }
            cout << "英语平均分为：" << english_average << endl;
            cout << "英语标准差为：" << english_standard << endl;
            cout << "英语及格人数为：" << english_num << endl;
        break;
        case 3:
            physics_num = 0; physics_standard = 0; physics_average = 0;
            for (int i = 0; i < MAX + g; i++)
                physics_average += a[i].physics;
            physics_average /= (MAX + g);
            for (int i = 0; i < MAX + g; i++)
                physics_standard += (a[i].physics - physics_average) *
(a[i].physics - physics_average);
            physics_standard = sqrt(physics_standard / (MAX + g));
            for (int i = 0; i < MAX + g; i++)
            {
                if (a[i].physics >= 60)
                    physics_num++;
                else
                    continue;
            }
            cout << "物理平均分为：" << physics_average << endl;
            cout << "物理标准差为：" << physics_standard << endl;
            cout << "物理及格人数为：" << physics_num << endl;
            break;
        case 4:
            num = 0; standard = 0; average = 0;
            for (int i = 0; i < MAX + g; i++)
                average += a[i].sum;
            average /= (MAX + g);
            for (int i = 0; i < MAX + g; i++)
                standard += (a[i].sum - average) * (a[i].sum - average);
            standard = sqrt(standard / (MAX + g));
            for (int i = 0; i < MAX + g; i++)
            {
                if (a[i].sum >= 180)
                    num++;
                else
                    continue;
            }
            cout << "总平均分为：" << average << endl;
            cout << "总标准差为：" << standard << endl;
            cout << "及格人数为：" << num << endl;
         break;
        default:
            break;
        }
        cout << "是（1）否（0）继续统计：";
        cin >> n;
    } while (n);
}
```

效果如下



修改学生信息

```
//修改学生信息
void student_system::amendfile()
{
    int n, k;
    cout << "请输入想要改动的信息对应的学生序号：";
    cin >> j;
    j--;
    cout << j + 1 << "\t" << a[j].ID << "\t" << a[j].name << "\t" "\t" <<
a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics <<
"\t" << "\t" << a[j].sum << endl;
    do
    {
        cout << "请输入改动内容(学号【0】；姓名【1】；数学【2】；英语【3】；物理【4】)：";
        cin >> k;
        switch (k)
        {
        case 0:
            cin >> a[j].ID; break;
        case 1:
            cin >> a[j].name; break;
        case 2:
            cin >> a[j].math; break;
        case 3:
            cin >> a[j].english; break;
        case 4:
            cin >> a[j].physics; break;
        }
        cout << j + 1 << "\t" << a[j].ID << "\t" << a[j].name << "\t" "\t" <<
a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics <<
"\t" << "\t" << a[j].sum << endl;
        cout << "修改成功，是【1】否【0】继续对该学生信息进行改动：";
        cin >> n;
    } while (n != 0);
}
```

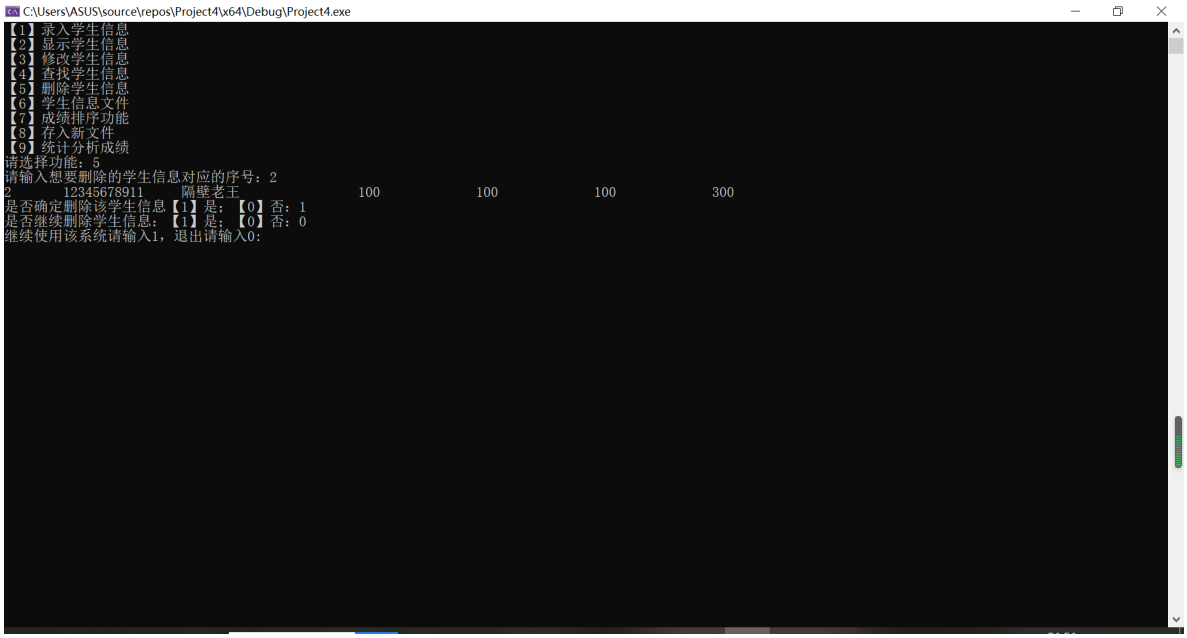
```
void student_system::findfile(){
    int n;
    cout << "请输入查找方式：【1】单个查找；【2】范围查找；【3】模糊查找：";
    cin >> n;
    switch (n) {
    case 1:
        dfindfile(); break;
    case 2:
        Ffindfile(); break;
    case 3:
        Mfindfile(); break;
    default:
        break;
    }
}
```

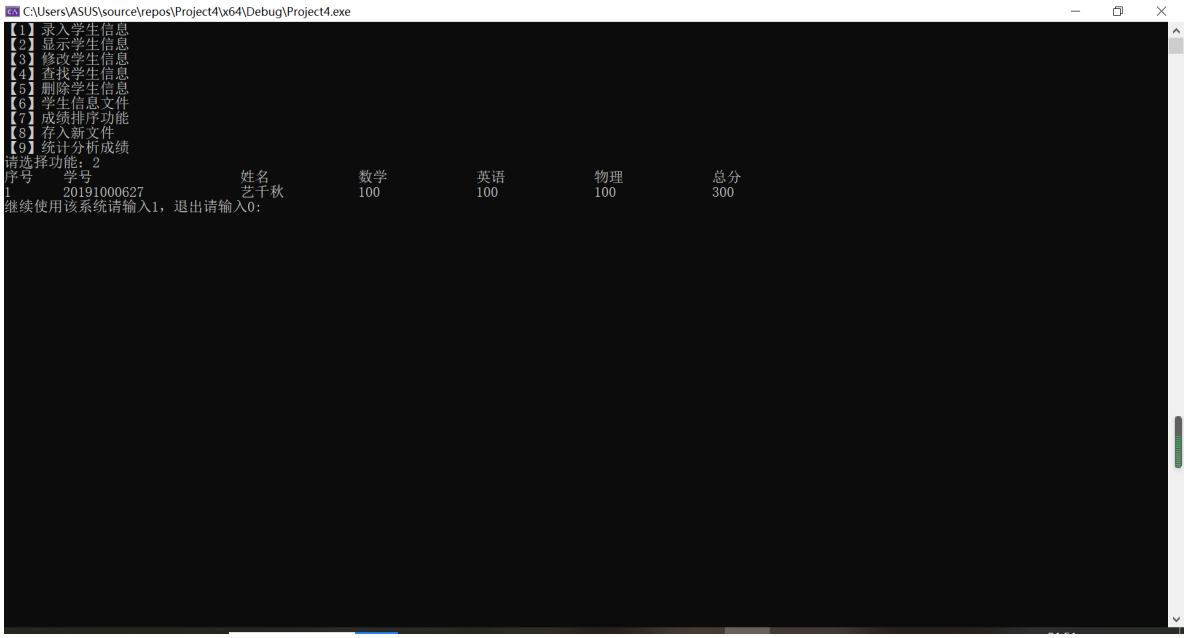
删除学生信息

```
//删除学生信息
void student_system::deletefile()
{
    int find;
    do {
        cout << "请输入想要删除的学生信息对应的序号：";
        cin >> find;
        j = find - 1;
        cout << j + 1 << "\t" << a[j].ID << "\t" << a[j].name << "\t" "\t" <<
a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics <<
"\t" << "\t" << a[j].sum << endl;
        cout << "是否确定删除该学生信息【1】是；【0】否：";
        cin >> find;
        for (int i = j; i < MAX + g - 1; i++)
            a[i] = a[i + 1];
        g = g - 1;
        cout << "是否继续删除学生信息：【1】是；【0】否：";
        cin >> find;
    } while (find != 0);
}
```

```
//成绩排序功能
void change(student&a1,student&a2) {
    student b;
    b = a1;
    a1= a2;
    a2 = b;
}
```

效果如下





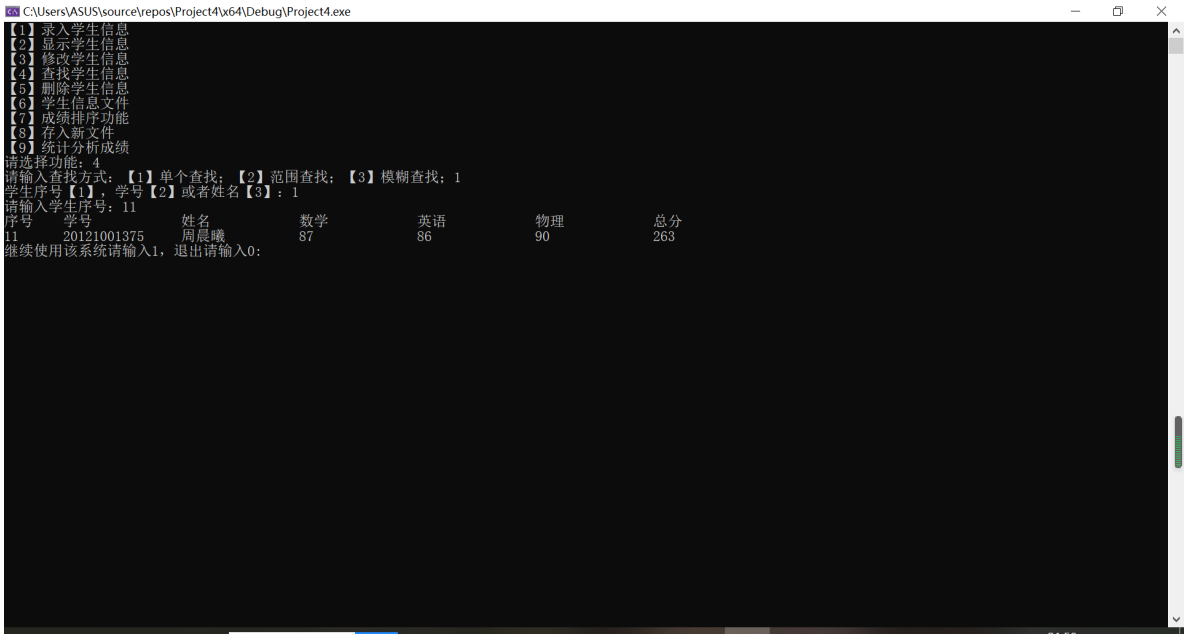
查找功能

```
void student_system::findfile(){
    int n;
    cout << "请输入查找方式：【1】单个查找；【2】范围查找；【3】模糊查找：";
    cin >> n;
    switch (n) {
    case 1:
        dfindfile(); break;
    case 2:
        Ffindfile(); break;
    case 3:
        Mfindfile(); break;
    default:
        break;
    }
}
```

1.单个信息查找

```
//单个查找学生信息
void student_system::dfindfile()
{
    int n=1;
    string name,ID1;
    int find;
    cout << "学生序号【1】，学号【2】或者姓名【3】：";
    cin >> n;
    switch (n)
    {
    case 1:
        cout << "请输入学生序号：";
        cin >> find;
        j = find - 1;
        break;
    case 2:
        cout << "请输入学生学号：";
```

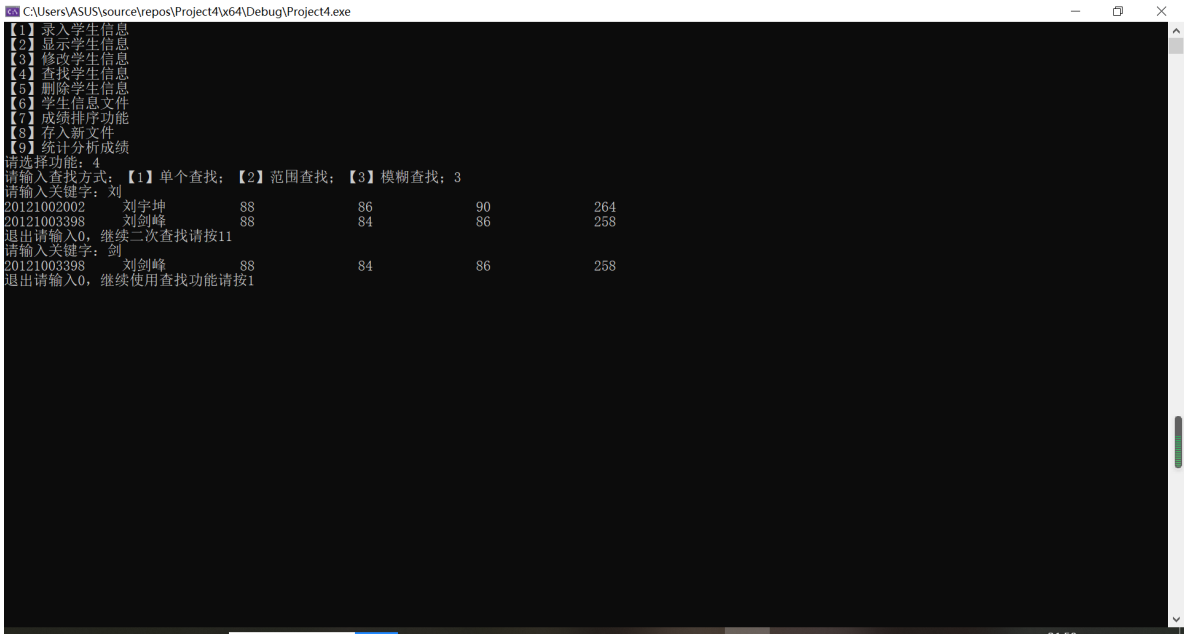
```
        cin >> ID1;
        for (int i = 0; i < MAX + g; i++)
        {
            if (a[i].ID == ID1)
            {
                j = i;
                break;
            }
            else if (a[i].ID != ID1 && i < MAX + g - 1)
                continue;
            else
            {
                cout << "无此学号，请重新输入：";
                cin >> ID1;
                i = -1;
            }
        }break;
    case 3:
        cout << "请输入学生姓名：";
        cin >> name;
        for (int i = 0; i < MAX + g; i++)
        {
            if (a[i].name == name)
            {
                j = i;
                break;
            }
            else if (a[i].name != name && i < MAX + g - 1)
                continue;
            else
            {
                cout << "无此人，请重新输入：";
                cin >> name;
                i = -1;
            }
        }
        break;
    default:
        break;
    }
    cout << "序号" << "\t" << "学号" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t" << "\t" << "英语" << "\t" "\t" << "物理" << "\t" "\t" << "总分" << endl;
    cout << j + 1 << "\t" << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t" << a[j].sum << endl;
}
```



2.模糊查找与二次查找

```
//模糊查找学生信息
void student_system::Mfindfile() {
    char m[20];
    int n;
    int s[50];
    do {
        int z = 0;
        cout << "请输入关键字：";
        cin >> m;
        char* str = m;
        char h[7],H[20];
        for (int j = 0; j < MAX + g; j++)
        {
            string str2= a[j].name;
            string str3 = a[j].ID;
            str2.copy(h, 7, 0);
            *(h + strlen(h)) = '\0';
            str3.copy(H, 20, 0);
            *(H + strlen(H)) = '\0';
            char* str1 = h;
            char* str4 = H;
            if (strstr(str1, str) != NULL|| strstr(str4, str) != NULL) {
                s[z] = j; z++;
                cout << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math
<< "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t"
<< a[j].sum << endl;
            }
            else
                continue;
        }
            cout << "退出请输入0，继续二次查找请按1";
            cin >> n;
            if (n == 1)
            {
                cout << "请输入关键字：";
                cin >> m;
```

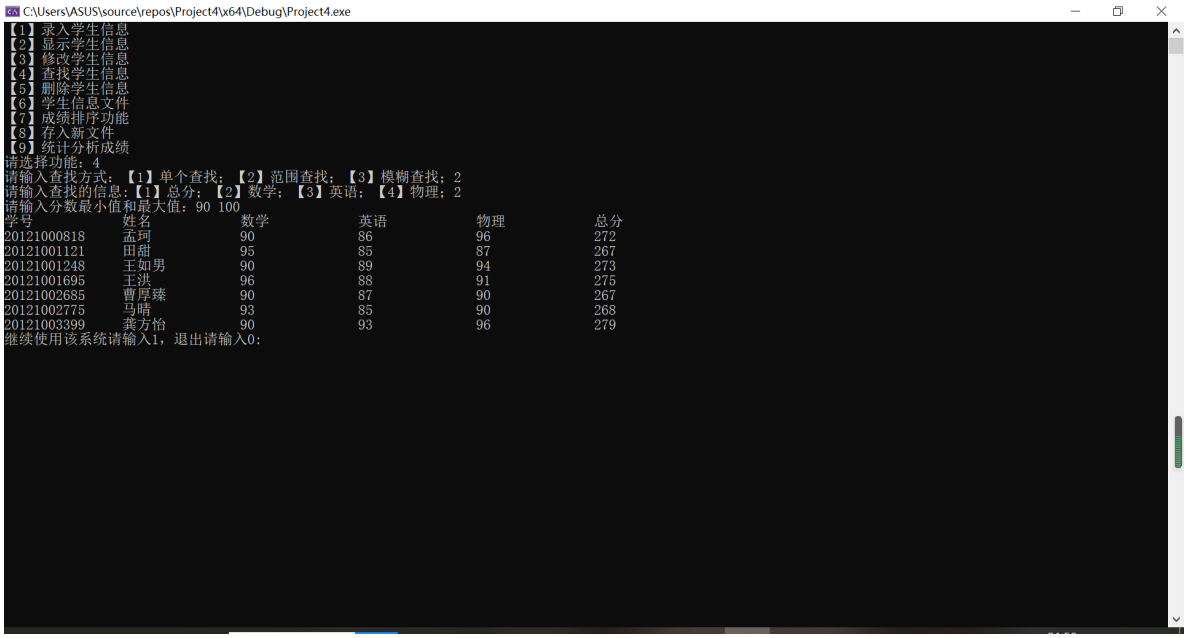
```
                for (int i = 0; i < z; i++)
                {
                    j = s[i];
                    string str2 = a[j].name;
                    string str3 = a[j].ID;
                    str2.copy(h, 7, 0);
                    *(h + strlen(h)) = '\0';
                    str3.copy(H, 20, 0);
                    *(H + strlen(H)) = '\0';
                    char* str1 = h;
                    char* str4 = H;
                    if (strstr(str1, str) != NULL || strstr(str4, str) != NULL)
{
                        cout << a[j].ID << "\t" << a[j].name << "\t" "\t" <<
a[j].math << "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics <<
"\t" << "\t" << a[j].sum << endl;
                    }
                    else
                        continue;
                }
            }
            cout << "退出请输入0，继续使用查找功能请按1";
            cin >> n;
    } while (n);
}
```



3.范围查找

```
//范围查找学生信息
void student_system::Ffindfile() {
    int n;
    cout << "请输入查找的信息:【1】总分；【2】数学；【3】英语；【4】物理；";
    cin >> n;
    int max, min;
    cout << "请输入分数最小值和最大值：";
    cin >> min >> max;
    switch (n)
    {
```

```
    case 1:
        cout << "学号" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t"
<< "\t" << "英语" << "\t" "\t" << "物理" << "\t" "\t" << "总分" << endl;
        for (int j = 0; j < MAX + g; j++)
            if (a[j].sum >= min && a[j].sum <= max)
            {
                cout << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math
<< "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t"
<< a[j].sum << endl;
            }break;
    case 2:
        cout << "学号" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t"
<< "\t" << "英语" << "\t" "\t" << "物理" << "\t" "\t" << "总分" << endl;
        for (int j = 0; j < MAX + g; j++)
            if (a[j].math >= min && a[j].math <= max)
            {
                cout << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math
<< "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t"
<< a[j].sum << endl;
            }break;
    case 3:
        cout << "学号" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t"
<< "\t" << "英语" << "\t" "\t" << "物理" << "\t" "\t" << "总分" << endl;
        for (int j = 0; j < MAX + g; j++)
            if (a[j].english >= min && a[j].english <= max)
            {
                cout << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math
<< "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t"
<< a[j].sum << endl;
            }break;
    case 4:
        cout << "学号" << "\t" << "\t" << "姓名" << "\t" << "\t" << "数学" << "\t"
<< "\t" << "英语" << "\t" "\t" << "物理" << "\t" "\t" << "总分" << endl;
        for (int j = 0; j < MAX + g; j++)
            if (a[j].physics >= min && a[j].physics <= max)
            {
                cout << a[j].ID << "\t" << a[j].name << "\t" "\t" << a[j].math
<< "\t" << "\t" << a[j].english << "\t" << "\t" << a[j].physics << "\t" << "\t"
<< a[j].sum << endl;
            }break;
    default:
        break;
    }
}
```

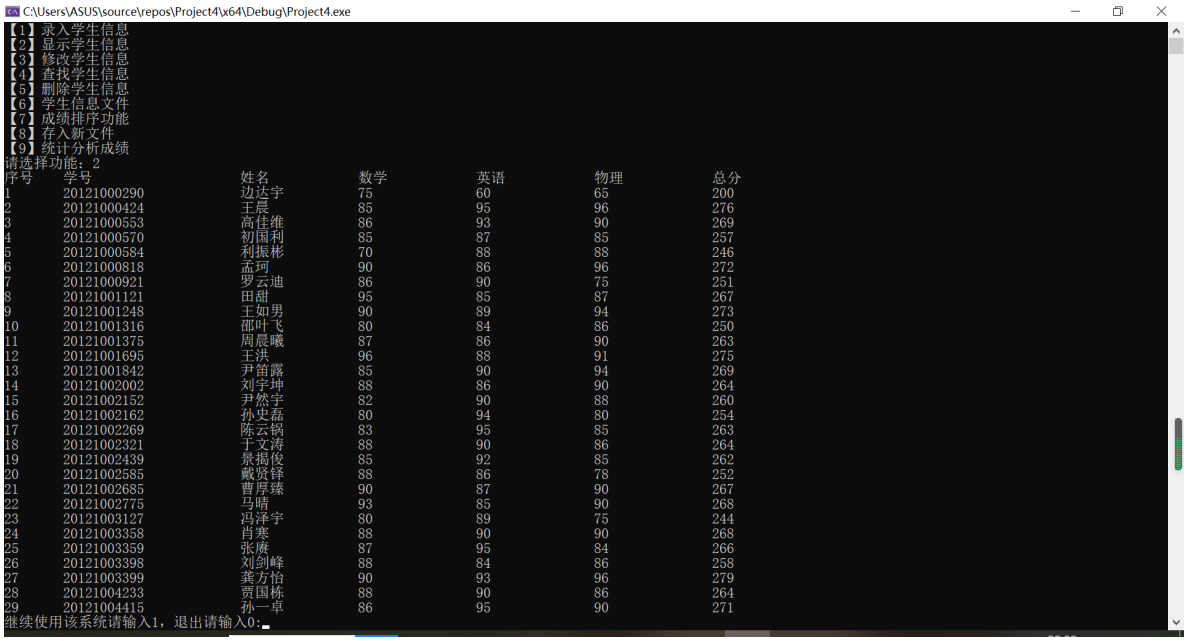


成绩排序功能

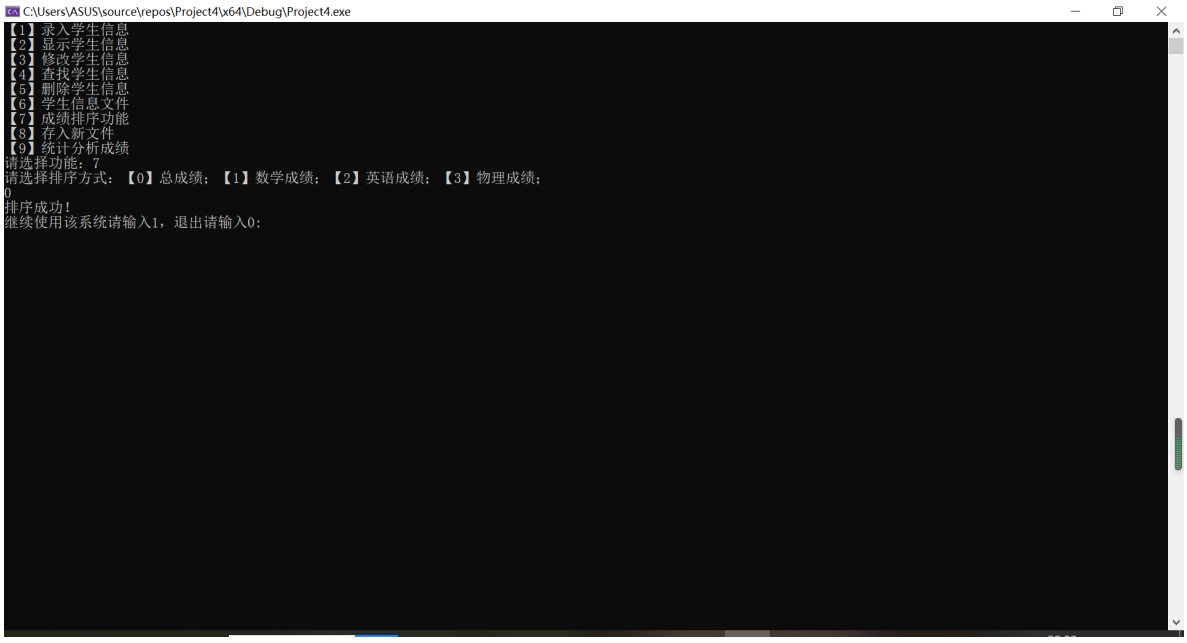
```
//成绩排序功能
void change(student&a1,student&a2) {
    student b;
    b = a1;
    a1= a2;
    a2 = b;
}
void student_system::rankfile()
{
    int num;
    int N = 1;
    cout << "请选择排序方式：【0】总成绩；【1】数学成绩；【2】英语成绩；【3】物理成绩；" <<
endl;
    cin >> num;
    student b;
    switch (num)
    {
    case 0:
        for (int i = 0; i < MAX + g; i++)
            for (int j = MAX + g - 1; j > i; j--)
                if (a[i].sum < a[j].sum)
                    change(a[i], a[j]);
                else if (a[i].sum == a[j].sum)
                    if (a[i].math < a[j].math)
                        change(a[i], a[j]);
                    else if (a[i].math == a[j].math)
                        if (a[i].physics < a[j].physics)
                            change(a[i], a[j]);
        break;
    case 1:
        for (int i = 0; i < MAX + g; i++)
            for (int j = MAX + g - 1; j > i; j--)
                if (a[i].math < a[j].math)
                    change(a[i], a[j]);
        break;
    case 2:
```

```
        for (int i = 0; i < MAX + g; i++)
            for (int j = MAX + g - 1; j > i; j--)
                if (a[i].english < a[j].english)
                    change(a[i], a[j]);
        break;
    case 3:
        for (int i = 1; i < MAX + g; i++)
            for (int j = MAX + g - 1; j > i; j--)
                if (a[i].physics < a[j].physics)
                    change(a[i], a[j]); break;
    default:
        break;
    }
    cout << "排序成功！" << endl;
}
```

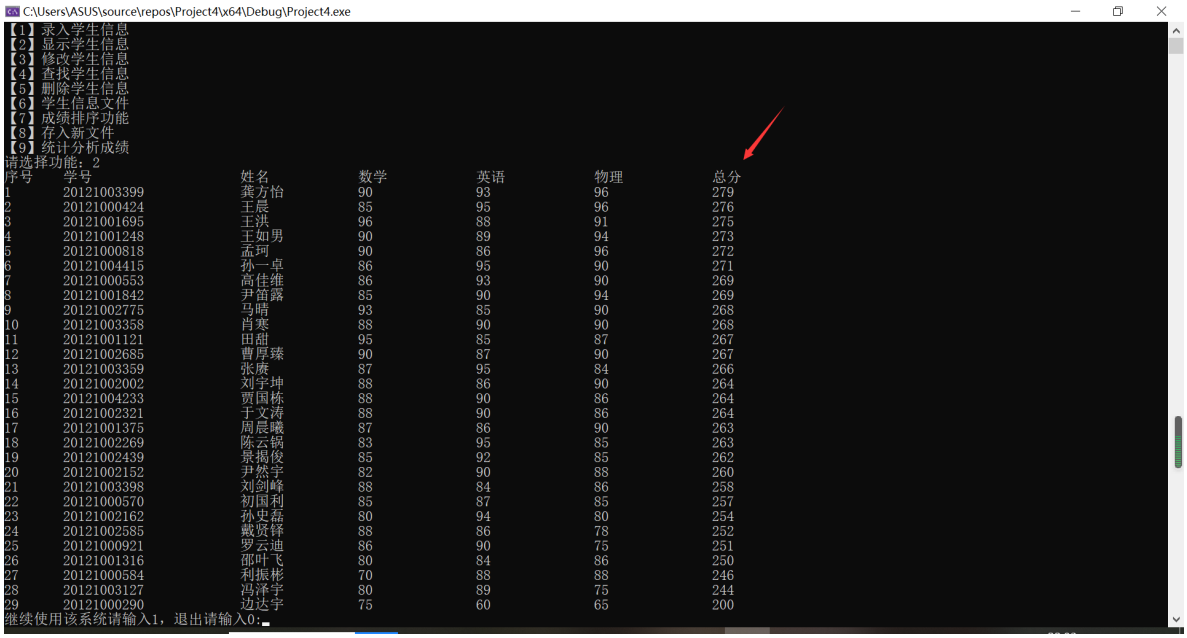
排序前



排序时



排序后



```
C:\Users\ASUS\source\repos\Project4\x64\Debug\Project4.exe
【1】录入学生信息
【2】显示学生信息
【3】修改学生信息
【4】查找学生信息
【5】删除学生信息
【6】学生信息文件
【7】成绩排序功能
【8】存入新文件
【9】统计分析成绩
请选择功能：2
序号    学号            姓名        数学        英语        物理        总分
1       20121003399     龚方怡      90          93          96          279
2       20121000424     王晨        85          95          96          276
3       20121001695     王洪        96          88          91          275
4       20121001248     王如男      90          89          94          273
5       20121000818     孟珂        90          86          96          272
6       20121004415     孙一卓      86          95          90          271
7       20121000553     高佳维      86          93          90          269
8       20121001842     尹笛露      85          90          94          269
9       20121002775     马晴        93          85          90          268
10      20121003358     肖寒        88          90          90          268
11      20121001121     田甜        95          85          87          267
12      20121002685     曹厚臻      90          87          90          267
13      20121003359     张赓        87          95          84          266
14      20121002002     刘宇坤      88          86          90          264
15      20121004233     贾国栋      88          90          86          264
16      20121002321     于文涛      88          90          86          264
17      20121001375     周晨曦      87          86          90          263
18      20121002269     陈云锅      83          95          85          263
19      20121002439     景揭俊      85          92          85          262
20      20121002152     尹然宇      82          90          88          260
21      20121003398     刘剑峰      88          84          86          258
22      20121000570     初国利      85          87          85          257
23      20121002162     孙史磊      80          94          80          254
24      20121002585     戴贤铎      88          86          78          252
25      20121000921     罗云迪      86          90          75          251
26      20121001316     邵叶飞      80          84          86          250
27      20121000584     利振彬      70          88          88          246
28      20121003127     冯泽宇      80          89          75          244
29      20121000290     边达宇      75          60          65          200
继续使用该系统请输入1，退出请输入0:
```

5.总结

实现亮点：

查找功能实现了单个查找、模糊查找、二次查找以及范围查找功能；

界面简洁、每次操作完一个功能可进行一次清屏操作，仅保留功能提示文字；

排序功能实现了多种排序方式：单科排序以及总成绩排序

统计功能实现了平均分、及格人数、标准差等数据的统计

输出新文件时可以新建文件名，保存路径，输出文件同时统计数据：包括总分平均分、及格人数、标准差等；

读入文件允许输入文件路径进行读取；

不足之处：

代码实现中有些变量显得多余，部分变量命名与其功能不太匹配；