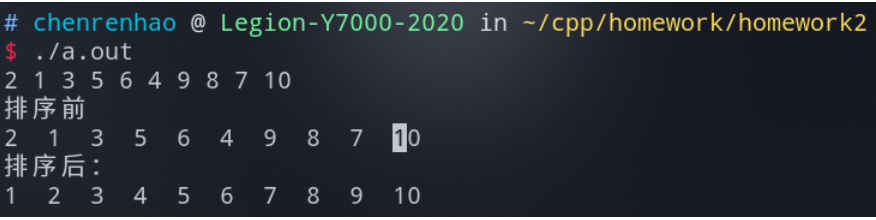
# 题目：

***分别设计排序算法和主函数，完成数组元素自动排序与显示***

[](https://imgtu.com/i/hLwGg1)

## 代码：

#include <iostream>  
using namespace std;  
#define ELEMENT\_COUNT 10  
void arryRank(int a[],int lenth)  
{  
 int i,j,\*target,temp;//target指针用于遍历数组  
 target = &a[0];  
 for (i = 0;i < lenth -1 ;i++)  
 {  
 //用target指针记录最大的元素  
 for(j = 0;j < lenth - i;j++)  
 {  
 if(a[j] > \*target)  
 {  
 target = &a[j];  
 }  
 }  
 j = 0;  
 //将最大的元素与最后一个元素交换位置  
 if(target != &a[lenth -i -1])  
 {  
 temp = \*target;  
 \*target = a[lenth - i - 1];  
 a[lenth - i -1] = temp;  
 }  
 target = &a[0];//target指针重新回到起点准备下一次遍历  
 }  
}  
void showArry(int a[],int lenth)  
{  
 int i;  
 for(i = 0;i < lenth;i ++)  
 {  
 cout<<a[i]<<" ";  
 }  
 cout<<endl;  
}  
int main()  
{  
 int a[ELEMENT\_COUNT];  
 for(int i = 0;i < ELEMENT\_COUNT;i++)  
 {  
 cin>>a[i];  
 }  
 cout<<"排序前"<<endl;  
 showArry(a,ELEMENT\_COUNT);  
 arryRank(a,ELEMENT\_COUNT);  
 cout<<"排序后："<<endl;  
 showArry(a,ELEMENT\_COUNT);  
 return 0;  
}

# 题目：

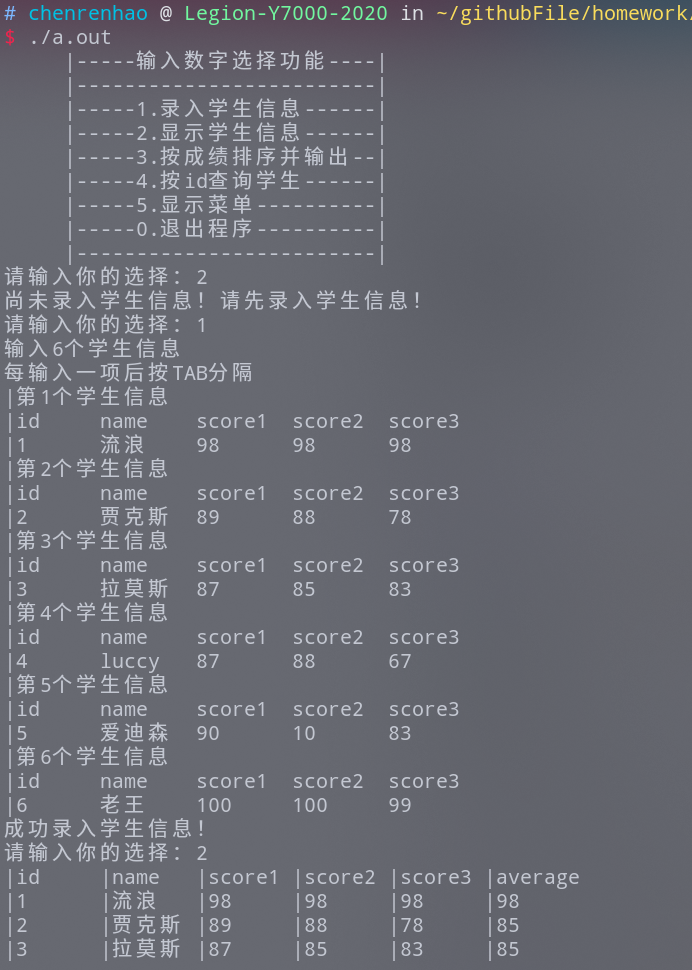
***有六名学生，每位学生的数据包括学号、姓名、三门课程的成绩，设计程序实现：***

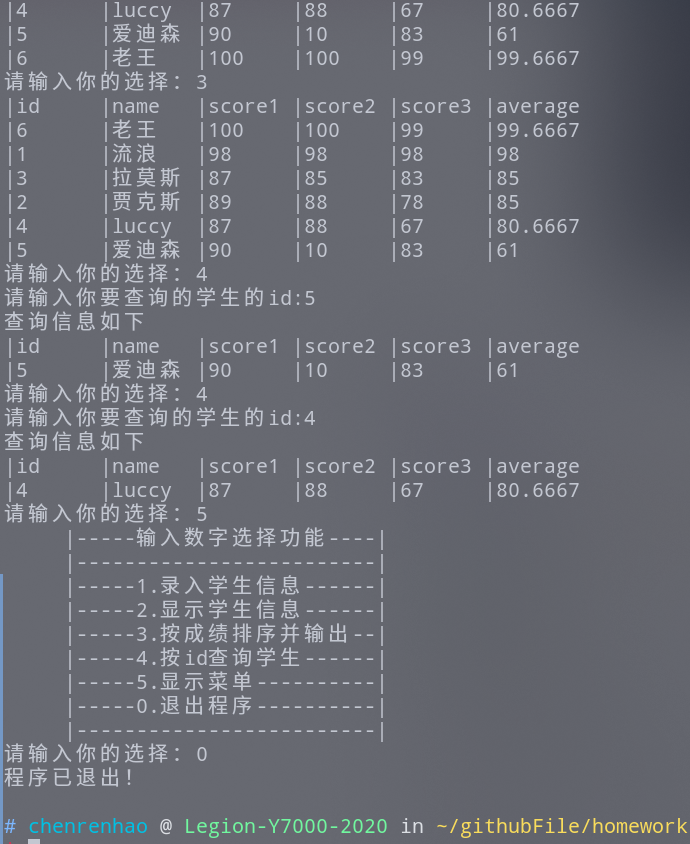
***(1)计算每位学生的平均成绩***

***(2)列表形式显示全部学生信息***

***(3)查找任意位置学生信息***

***(4)按平均成绩由高到低打印全部学生信息***



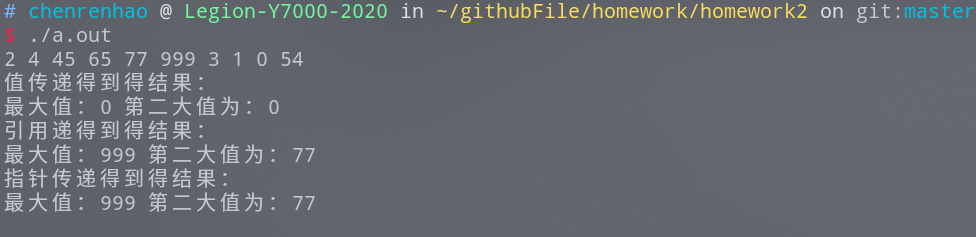


## 代码：

#include <iostream>  
using namespace std;  
struct Student  
{  
 int id = -1;//将id默认标记为-1用于判断数据是否初始化  
 string name;  
 float score1;  
 float score2;  
 float score3;  
 float average;  
};  
//求成绩平均分的函数  
void average(Student students[6])  
{  
 for(int i = 0;i < 6;i ++)  
 {  
 students[i].average = (students[i].score1 + students[i].score2 +students[i].score3)/3;  
 }  
}  
//输入数据的函数  
void inputMessage(Student students[6])  
{  
 if(students[0].id != -1)//第一次录入信息后不能重复录入  
 {  
 cout<<"已录入学生信息，不可重复录入！"<<endl;  
 return;  
 }  
 cout<<"输入6个学生信息"<<endl;  
 cout<<"每输入一项后按TAB分隔"<<endl;  
 for(int i = 0;i < 6;i++)  
 {  
 //默认按id为1,2,3,4,5,6来输入  
 students[i].id = i+1;  
 cout<<"|第"<<i + 1<<"个学生信息"<<endl;  
 cout<<"|id\tname\tscore1\tscore2\tscore3\t"<<endl;  
 cout<<"|"<<i+1<<"\t";  
 cin>>students[i].name>>students[i].score1>>students[i].score2>>students[i].score3;  
 }  
 //录入学生信息后顺便将平均分求出来  
 average(students);  
 cout<<"成功录入学生信息！"<<endl;  
}  
//输出学生信息的函数  
void showMessage(Student students[6])  
{  
 //用id是否为-1 来判断数据是否初始化  
 if(students[0].id == -1)  
 {  
 cout<<"尚未录入学生信息！请先录入学生信息！"<<endl;  
 return;  
 }  
 cout<<"|id\t|name\t|score1\t|score2\t|score3\t|average"<<endl;  
 for(int i = 0;i < 6;i++)  
 {  
 cout<<"|"<<students[i].id<<"\t|"<<students[i].name<<"\t|"<<students[i].score1<<"\t|"<<  
 students[i].score2<<"\t|"<<students[i].score3<<"\t|"<<students[i].average<<endl;  
 }  
  
}  
//按id查找学生的函数  
void findStudent(Student students[6])  
{  
 //先判断数据是否初始化  
 if(students[0].id == -1)  
 {  
 cout<<"尚未录入学生信息！请先录入学生信息！"<<endl;  
 return;  
 }  
 int id;  
 cout<<"请输入你要查询的学生的id:";  
 cin>>id;  
 //根据用户输入的id输出对应的信息  
 if(id > 6 || id < 1)  
 {  
 cout<<"输入错误！请重新输入！"<<endl;  
 return;  
 }  
 Student \*p = students;//通过指针p遍历数组进行查找  
 while(p->id != id) p++;  
 cout<<"查询信息如下"<<endl;  
 cout<<"|id\t|name\t|score1\t|score2\t|score3\t|average"<<endl;  
 cout<<"|"<<p->id<<"\t|"<<p->name<<"\t|"<<p->score1<<"\t|"<<p->score2<<"\t|"<<  
 p->score3<<"\t|"<<p->average<<endl;  
}  
//按平均数用选择法进行排序  
void rankStudent(Student students[6])  
{  
 Student \*p,temp;//p 用来记录分数最小的学生  
 int i,j;  
 p = &students[0];  
 for(i = 0,j = 0; i < 5;i ++)  
 {  
 //遍历出分数最小的学生  
 while(j < 6 - i)  
 {  
 if(p->average > students[j].average)  
 p = &students[j];  
 j++;  
 }  
 j = 0;  
 //将分数最小的学生交换到数组的末尾  
 if(p != &students[6 - i -1])  
 {  
 temp = \*p;  
 \*p = students[6 - i -1];  
 students[6 - i -1] = temp;  
 }  
 //从头开始遍历  
 p = &students[0];  
 }  
  
}  
//展示菜单的函数  
void showMenu()  
{  
 cout<<" |-----输入数字选择功能----| "<<endl;  
 cout<<" |-------------------------| "<<endl;  
 cout<<" |-----1.录入学生信息------| "<<endl;  
 cout<<" |-----2.显示学生信息------| "<<endl;  
 cout<<" |-----3.按成绩排序并输出--| "<<endl;  
 cout<<" |-----4.按id查询学生------| "<<endl;  
 cout<<" |-----5.显示菜单----------| "<<endl;  
 cout<<" |-----0.退出程序----------| "<<endl;  
 cout<<" |-------------------------| "<<endl;  
}  
//实现菜单的逻辑  
void menu(Student students[6])  
{   
 bool isRun = true;//isRun作为程序运行的标志  
 int ch;//记录用户的输入  
 showMenu();  
 while(isRun)  
 {  
 cout<<"请输入你的选择：";  
 cin>>ch;  
 switch(ch)  
 {  
 case 1:{  
 inputMessage(students);  
 break;  
 }  
 case 2:{  
 showMessage(students);  
 break;  
 }  
 case 3:{  
 rankStudent(students);  
 showMessage(students);  
 break;  
 }  
 case 4:{  
 findStudent(students);  
 break;  
 }  
 case 5:{  
 showMenu();  
 break;  
 }  
 case 0:{  
 isRun = false;  
 cout<<"程序已退出！"<<endl;  
 break;  
 }  
 default:  
 cout<<"输入错误！请重新输入！"<<endl;  
 }  
 }  
}  
int main()  
{  
 Student students[6];  
 menu(students);//主函数只负责调用菜单逻辑函数  
 return 0;  
}

# 题目：

***分别使用整型变量、指针变量和引用变量三种参数形式设计函数，实现查找某整型数组中的最大值和次大值，并在主函数中进行调用，输出结果***

[](https://imgtu.com/i/hL6Nz4)

## 代码：

#include <iostream>  
using namespace std;  
//值传递  
void solution1(int arr[],int max1,int max2)  
{  
 int i;  
 max1 = arr[0];  
 for(i = 0;i < 10;i++)  
 {  
 if(arr[i] > max1)  
 {  
 max2 = max1;  
 max1 = arr[i];  
 }  
 if(arr[i] > max2 && arr[i] != max1)  
 max2 = arr[i];  
 }  
}  
//引用传递  
void solution2(int arr[],int &max1,int &max2)  
{   
 int i;  
 max1 = arr[0];  
 for(i = 0;i < 10;i++)  
 {  
 if(arr[i] > max1)  
 {  
 max2 = max1;  
 max1 = arr[i];  
 }  
 if(arr[i] > max2 && arr[i] != max1)  
 max2 = arr[i];  
 }  
}  
//指针传递  
void solution3(int arr[],int \*max1,int \*max2)  
{  
 int i;  
 \*max1 = arr[0];  
 for(i = 0;i < 10;i++)  
 {  
 if(arr[i] > \*max1)  
 {  
 \*max2 = \*max1;  
 \*max1 = arr[i];  
 }  
 if(arr[i] > \*max2 && arr[i] != \*max1)  
 \*max2 = arr[i];  
 }  
}  
int main()  
{  
 int max1 = 0,max2 = 0;  
 int arr[10];  
 for(int i = 0; i < 10 ;i ++)//输入整型数组的数据  
 {  
 cin>>arr[i];  
 }  
 solution1(arr,max1,max2);  
 cout<<"值传递得到得结果："<<endl;  
 cout<<"最大值："<<max1<<" 第二大值为："<<max2<<endl;  
 max1 = 0;  
 max2 = 0;  
 solution2(arr,max1,max2);  
 cout<<"引用递得到得结果："<<endl;  
 cout<<"最大值："<<max1<<" 第二大值为："<<max2<<endl;  
 max1 = 0;  
 max2 = 0;  
 solution3(arr,&max1,&max2);  
 cout<<"指针传递得到得结果："<<endl;  
 cout<<"最大值："<<max1<<" 第二大值为："<<max2<<endl;  
 return 0;  
}