**一、课程设计题目与要求**

**1.1题目**

分数计算器的设计与实现

**1.2系统功能要求**

（1）按照分数类Fraction的声明，实现其所有成员函数、友元及运算符重载

（2）设计一个主菜单，让用户在程序开始时能根据自己的需求选择对应的功能：分数计算、分数排序、退出

（3）设计分数计算的界面，并在本界面实现两个分式的加减乘除运算。如果用户输入正确的分数计算式，则给出计算结果；如果用户输入了错误的分数计算式，就提示输入错误并等待用户重新输入

（4）设计分数排序的界面，并在此界面实现分数的排序。根据用户输入的一组正确分数和排序要求（“>”或“<”），输出排序结果；如果用户输入的分数不正确，就提示输入错误 ，并等待用户重新输入

（5）在分数计算和分数排序的界面增加返回主菜单的功能，让用户能够返回主菜单重新选择功能或退出程序

**二．系统需求分析**

**2.1需求分析**

**（1）主菜单界面**

①能根据用户选择的功能进入对应功能的界面

②能让用户退出程序

③当用户输入错误时，能提示用户输入错误，并等待用户重新输入

**（2）分数计算界面**

①能识别并存储分数计算式中的分数

②能识别并储存分数计算式中的计算符号

③能进行分数的加法运算

④能进行分数的减法运算

⑤能进行分数的乘法运算

⑥能进行分数的除法运算

⑦能让用户返回主菜单

⑧在用户输入了错误的分数计算式时，能提示用户输入错误并等待用户重新输入

**（3）分数排序界面**

①能识别并储存一组正确分数中的分数

②能识别并储存用户的排序要求（“>”或“<”）

③能对一组分数进行排序由大到小的排序

④能让用户返回主菜单

⑤在用户输入错误时，能提示输入错误并等待用户重新输入

**2.2系统性能要求**

①系统安全、可靠

②功能齐全

③操作简单、界面友好

④易于维护和扩充

**2.3系统的功能分析**

分数计算：可以进行分数的加减乘除运算

分数排序：可以根据用户的要求由小到大（或由大到小）排序

退出：退出程序

**三、总体设计**

**3.1系统功能模块**

分数计算器由分数计算、分数排序和退出三个功能构成

**3.2模块框架图**

分数计算器

分数计算

分数排序

退出程序

加法运算

减法运算

乘法运算

除法运算

由小到大

由大到小

**四、详细设计**

**4.1菜单的用户输入模块设计**

首先，在用户的每一次输入前，都使用system（“cls”）清除屏幕，保持界面的整洁。

然后，提示用户选择功能，将用户输入的内容储存在一个字符型变量num中。如果用户输入的是“1”，则进入分数计算界面；如果用户输入的是“2”，则进入分数排序界面；如果用户输入的是“0”，则退出程序；如果用户输入的是其他内容，则提示用户输入错误，并等待用户重新输入。

用字符串变量储存用户的输入，可以快速判断出用户错误的输入。因为用户一旦输入除了“0”“1”“2”以为的字符串，程序会直接判断用户输入错误

**4.2分数计算界面的用户输入模块设计**

首先，在用户的每一次输入前，都使用system（“cls”）清除屏幕，保持界面整洁、友好。

然后，提示用户输入分数计算式，把用户输入的分数计算式储存到一个字符串变量input中，并判断该分数计算式是否正确。如果正确，则返回分数计算式的结果；否则，提示分数计算式输入错误，并等待用户重新输入。

**4.3分数计算式的判断模块设计**

①判断用户输入的是不是“#”，如果是，则返回菜单

②判断input的第一个字符是不是“+”或者“-”，如果是，则将这个字符储存到字符串变量sign1中。这样做可以避免用户在输入正数带“+”或输入负数带“-”时被程序判断为输入错误

③依次对input中的每个字符进行判断，如果是数字字符，则将他储存到数组num中，直到遇到非数字字符。如果没有一个数字字符，就判断用户输入错误，避免用户输入无效分数

④通过数组num，把之前判断到的数字储存到整数型x1中，并判断sign1是不是“-”，如果是，把x1\*（-1），实现负数的输入

⑤判断遇到的字符是不是“/”，如果是，重复②③④的操作，得到x2，如果x2=0，判断用户输入错误，防止用户输入无效分数，否则，创建一个分子为x1，分母为x2的分数frac1；如果不是“/”，则判断用户输入错误，避免用户输入其他字符

⑥判断上述操作后遇到的字符是不是四种计算符号中的一种，如果是，把它储存到字符串变量symbol中，否则，判断用户输入错误，避免用户输入了非运算符的字符

⑦最后，重复上述操作，得到另一个分数frac2

**4.4分数计算的加法和减法计算模块设计**

有两个分数frac1和frac2，先把他们的分母相乘，得到整型变量n1，再把他们的分子分别乘上对方的分母，然后把他们的分子相加或相减，得到整型变量n2。进而得到一个以n1为分母，n2为分子的新分数，再把这个新分数化简

**4.5分数计算的乘法和除法计算模块设计**

有两个分数frac1和frac2，如果是乘法，就先把它们的分子分母相乘，得到整型变量n1,n2 ，进而得到一个以n1为分母，n2为分子的新分数，再把这个新分数化简

如果是除法，就把它们的分子分别与对方的分母相乘，得到整型变量n1,n2 ，进而得到一个以n1为分母，n2为分子的新分数，再把这个新分数化简

**4.6分数排序的用户输入模块设计**

首先，在用户的每一次输入前，都使用system（“cls”）清除屏幕，保持界面整洁、友好。

然后，提示用户输入一组分数，把用户输入的分数储存到一个字符串变量input中，并判断该组分数是否正确，如果正确，则按照用户的排序要求进行相应的输出，否则，提示输入错误，等待用户重新输入

**4.7一组分数的判断模块设计**

①判断用户输入的是不是“#”，如果是，则返回菜单

②判断input的第一个字符是不是“+”或者“-”，如果是，则将这个字符储存到字符串变量sign中。这样做可以避免用户在输入正数带“+”或输入负数带“-”时被程序判断为输入错误

③依次对input中的每个字符进行判断，如果是数字字符，则将他储存到数组num中，直到遇到非数字字符。如果没有一个数字字符，就判断用户输入错误

④通过数组num，把之前判断到的数字储存到整数型x1中，并判断sign1是不是“-”，如果是，把x1\*（-1），实现负数的输入

⑤判断遇到的字符是不是“/”，如果是，重复②③④的操作，得到x2，如果x2=0，判断用户输入错误，避免用户输入无效分数，否则，创建一个分子为x1，分母为x2的分数test，并把它储存到vector容器nums中；如果不是“/”，判断用户输入错误

⑥判断上述操作后遇到的字符是不是“，”，如果不是

⑦最后，重复上述操作，直到遇到的字符是“<”或“>”，或者输入错误

**4.8分数的排序模板设计**

采用冒泡排序法，从小到大排序。

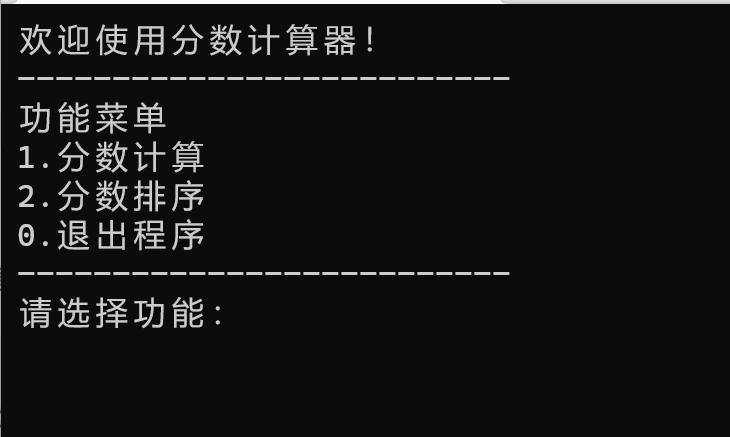
从左到右依次比较相邻分数，如果左边的分数较大，就把它和右边的分数互换位置。第一轮比较后，所有分数中最大的那个数就会浮到最右边；第二轮比较后，所有数中第二大的那个数就会浮到倒数第二个位置……就这样一轮一轮地比较，最后实现从小到大排序。

在冒泡排序中，每一轮的比较次数都会减少，同时避免了在分数已经从小到大排序的情况下继续比较，缩短了程序运行时间，提高了程序的运行效率

**五、系统实现**

1. **主菜单**

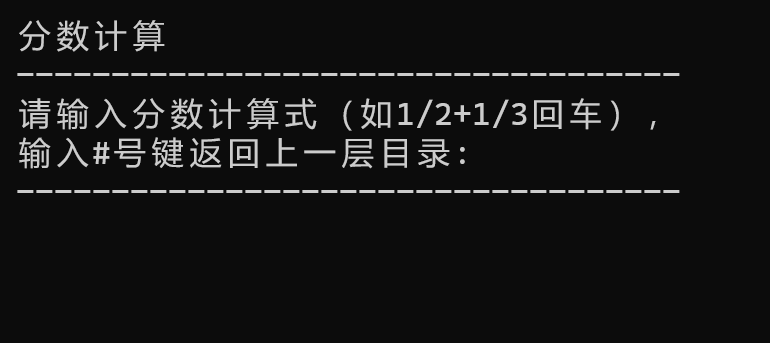
打开分数计算器后，将会进入一个欢迎菜单，也就是本程序的主菜单，在这样有如下几个选项：



根据需要输入相应的数字

**2.分数计算**

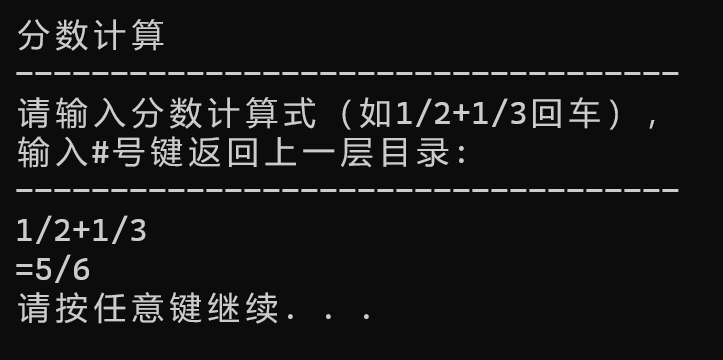
在主菜单输入数字1后，将会进入如下的分数计算界面：



本程序支持两个分数的加减乘除运算，会把计算得到的结果化简后再返回给用户。

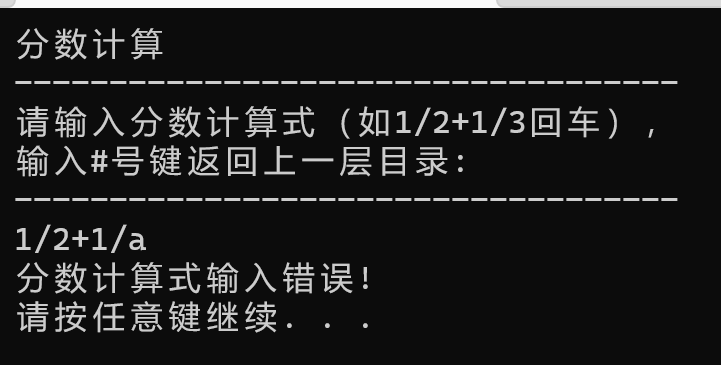
如果想重新选择功能或者退出程序，可输入“#”键返回主菜单。

如果输入正确的分数计算式，本程序会给出计算结果，并等待用户的下一次输入



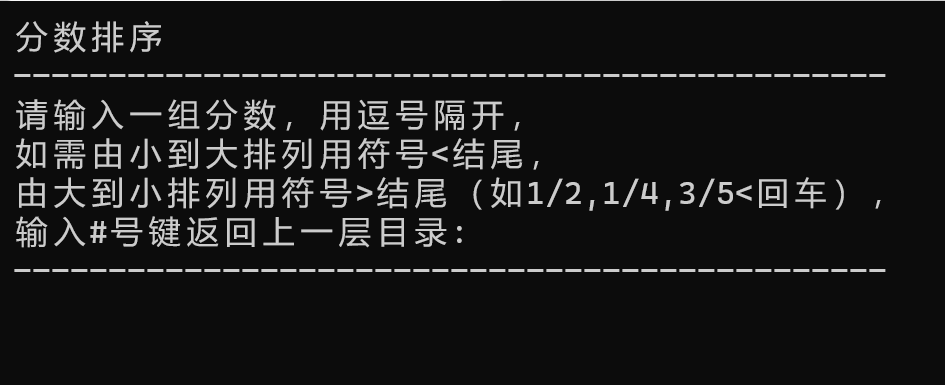
注意：不要在分数计算式中输入空格

如果输入了错误的分数计算式，本程序会提示输入错误，并等待用户下一次输入



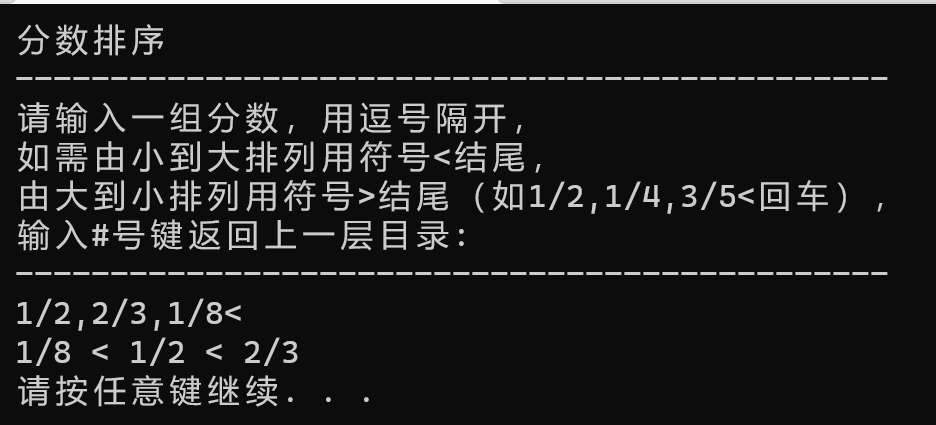
**3、分数排序**

在主菜单输入数字2后，将会进入如下的分数排序界面：



本程序支持多个分数的排序，可根据用户需求，将用户输入的一组分数排序后，按从小到大或从大到小的顺序输出

如果输入一组正确的分数，本程序会给出按照用户要求排序后的分数



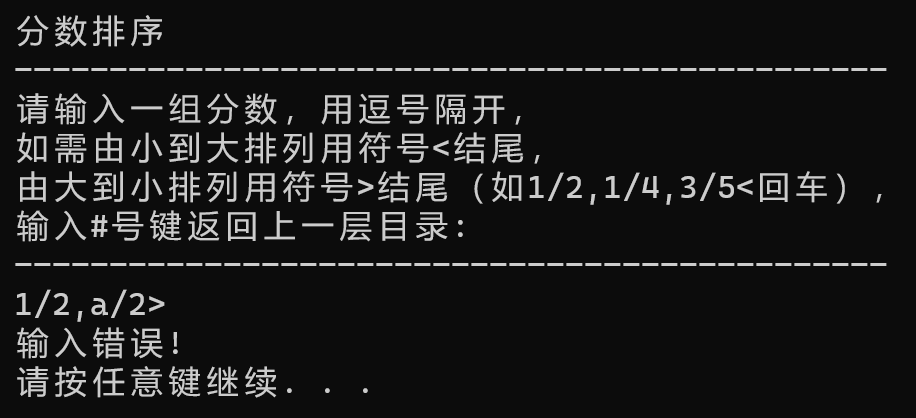
注意

①不要输入空格

②分数之前用“，”隔开

③最后一个分数后面要加上“<”或“>”

如果输入的分数不正确，本程序会提示输入错误，并等待用户的下一次输入



**六、总结**

本次实验项目，在指导老师的精心教导下，我学会了如何用C++编写一个简单的应用程序。首先要对程序的设计要求有一个比较明确的认知，熟知程序的功能，然后系统分析与系统设计，解决编写程序中遇到的问题，最后是代码设计，把理论变成现实。

我编写程序的过程虽然辛苦，但我通过本次实验项目，进一步加深了对C++语音的理解和掌握，提高了自己的实践能力和计算机的综合运用能力。

本程序实现了简单的分数计算和分数排序，界面整洁、友好，但仍存在许多不足之处，如本程序只适用于分子分母都是整数的分数，不适应于分子或分母是非整数的分数；本程序只能对两个分数进行运算，无法对多个分数进行运算。

**七．参考文献**

**刘瑞芳,肖波,许桂平等.C++语音程序设计[M].西安：西安电子科技大学出版社.，2019**