

2024-2025 春季学期 计算思维与程序实践 II 期中测试 (A 卷)

考试时间: 2025.04.23 16:50-18:50 共计 2 小时

考试形式: 机试、半开卷

(可携带含个人签名的一页 A4 幅面资料, 一本参考书。不允许带其他资料及上网)

【请仔细阅读答题说明】

1. 本试卷共 3 题。试卷总分 100 分。
2. **每题都提供了部分代码, 在此基础上按要求完成代码设计, 不得改变原代码结构。**
3. 注意代码可读性、书写格式规范性, 格式不规范每题可给予 1-5 分的扣分。
4. 提交的源程序文件用“学号 + f 题号”命名, 题号与试卷上的题号一致。例如, 学号为 61234567 同学试卷第 1 题的答题源程序文件名为 61234567f1.cpp。
5. **务必在考试结束前预留一定时间将答卷提交到服务器, 并检查是否所有答卷均已提交。未来得及上传答卷自行承担损失。**
6. **不能提前交卷, 离开考场!**

【答题步骤】

1. 在 **D 盘建立工作文件夹**(如: 文件夹命名为学号), 将下载的试卷拷贝至该文件夹下, 在此文件夹下完成代码编写及调试工作。**不要在 C 盘或网盘上编写调试程序!!!**以防死机等造成文件丢失。
2. 试卷下载方法: 考试开始时, 运行桌面批处理文件创建 Z 盘(网络盘), 在 Z 盘上找到试题。
3. 提交答卷时, **仅需提交答卷的源程序**文件(.cpp 文件), 并保证提交源代码文件命名正确。提交方法: 在**网盘 Z**建立以“学号+姓名”命名的“**答卷文件夹**”(例如: 学号 61234567 同学, 姓名张强, 答卷文件夹命名为: 61234567 张强), 用于存储上传的答卷代码文件。将答题的源代码文件拷贝至 **Z 盘“答卷文件夹”**中。

【试题】

第1题（30分）

【问题描述】

编程设计完成学生成绩和一组成绩的管理功能。成绩类 `Score`，用于定义一个学生成绩信息及相关处理函数。`ScoreManagement` 类用于实现一组成绩的创建，数据访问、排序等操作。代码定义了 2 个全局函数，分别用于输出一个学生成绩和一组学生成绩。主函数测试成绩管理功能的正确性。

【编程要求】

本题提供代码 `A1.cpp`，在此框架下，完善代码设计，通过 `main` 函数测试。

1. 学生成绩类 `Score`，包括数据成员数学、物理成绩，按照题目要求，完善类设计；
2. 重载比较运算符，实现两个学生成绩高低比较。比较原则：先用 `math` 进行比较，如果 `math` 相等，则用 `physical` 进行比较。该运算符重载满足排序算法设计需求；
3. 设计 `ScoreManagement` 类成员函数 `sort`，任选一种算法，实现降序排序；
4. 根据提示，完善主函数测试代码设计，验证设计的正确性。

第2题（30分）

【问题描述】

采用泛型编程技术，实现处理数据序列中可能存在重复数据的问题，统计其中不重复数据的个数（即去重后的数据个数）。如，整数序列 `{1, 5, 3, 1, 7, 5, 2, 1, 4}`，9 个数据中 1 和 5 多次出现，统计结果为 6 个，即出现 1、5、3、7、2、4 这些不相同的数据。

【编程要求】

本题提供代码 `A2.cpp`，在此框架下，完善代码设计，通过 `main` 函数测试。

1. 编写函数模板 `countDuplicates`，功能为统计序列中不重复数据的个数；
2. 定义 `MyRGB` 类，包含红、绿和蓝 `r/g/b` 分量的数据成员：
 - 1) 定义颜色的相等规则为若两个颜色的整数灰度值相等，则认为它们的值相等；
 - 2) 灰度值计算规则为使用标准加权平均灰度转换公式，并转换为整数：

$$Gray = int(0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B + 0.5)$$

3. 主函数完成测试。定义了一组 `RGB` 颜色，调用模板函数，统计基于灰度值去重的 `RGB` 颜色数量，输出不重复灰度值的颜色个数。

【简答题】

函数模板和函数重载是 C++ 中常用的两种机制，从函数原型设计和应用场景角度，两者有何不同？（在源程序代码文件最后注释区域给出问题解答。）

第3题 （40分）

【问题描述】

设计按照规则保存某种卡牌游戏的多张牌程序。卡牌的四边按照按左 上 下 右的次序显示次序有4个数字。程序设计了卡牌类Card，4个数据成员left, top, bottom, right, 用来描述标有左 上 下 右四个数字的卡牌，如图1所示：

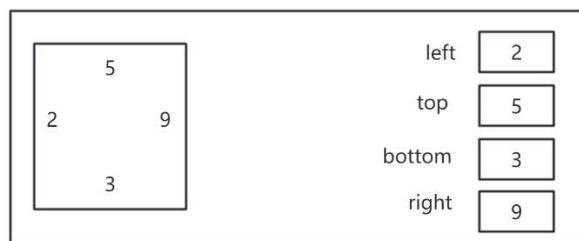


图1 卡牌示例

卡牌的保存要求每一张卡牌插入数组时相邻卡牌的左右相等衔接。以图2卡牌列表为例，第一张牌的右边的数字与第二张的左边相同，都是3，以此类推。

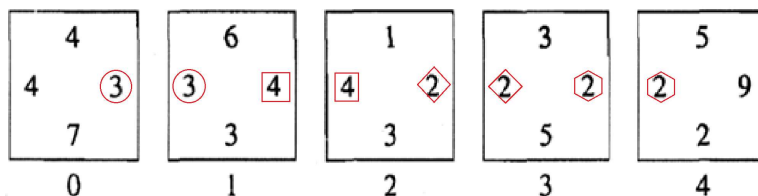


图2 卡牌列表示例

【代码说明】

题目已给代码注意说明如下：

1. 程序包含卡牌类Card、卡牌列表类 CardList和测试主函数；
2. 卡牌类Card，描述一张卡牌信息，包含数据成员left, top, bottom, right；
3. 卡牌列表类CardList，采用动态数组，按照规则存放多张卡牌：

1) 成员包括

```
int size;    // 卡牌数组容量
int number;  // 卡牌的实际个数
Card *list;  // 指向动态数组的指针
```

2) CardList类的带参构造函数CardList(const Card *arr, int size_)已经提供，其功能是根据传递的卡牌数组地址和长度，创建卡牌列表。首先创建指定大小的卡牌的动态数组，然后依次读取数组形参当中的每张卡牌，按照如下规则调用append函数，添加到list当中，每一张卡牌插入时和相邻卡牌的左右相等衔接，如图2所示。

3) 卡牌添加规则：

- a) list数组元素有效下标范围从0到number - 1，新卡片待插入的位置是从number 到 0递减选择；
- b) 从第一个待插入位置number开始进行判断，如无法插入则更改至下一个待插入位置判断；

c) 如果有任何位置可以插入，则调用insert函数进行插入，且提前返回；如果所有位置均无法插入则放弃这张牌的插入；

【编程要求】

本题提供代码 A3.cpp，在此框架下，完善代码设计，通过 main 函数测试。

1. 定义Card类构造函数；
2. 定义CardList类无参默认构造函数，将size、number都置为0，list置为nullptr；
3. 完善insert函数设计，配合append函数，实现卡牌插入，成员函数原型定义如下：
void insert(int index, const Card& card);
4. 根据main函数测试代码需求，完善CardList类设计(注意动态内存管理，有多个函数)；
5. 完成CardList类的输出运算符重载，实现main函数测试中卡牌列表输出，共有5个列表输出，结果如图3所示，每行表示一个卡牌列表的输出（包括：卡牌数，每张卡牌的4个数字）：

```
Total 6 cards: (4 4 7 3) (3 6 3 4) (4 1 3 2) (2 3 5 2) (2 4 9 2) (2 5 2 9)
Total 6 cards: (4 4 7 3) (3 6 3 4) (4 1 3 2) (2 3 5 2) (2 4 9 2) (2 5 2 9)
Total 0 cards: Empty
Total 0 cards: Empty
Total 6 cards: (4 4 7 3) (3 6 3 4) (4 1 3 2) (2 3 5 2) (2 4 9 2) (2 5 2 9)
```

图3 CardList类输出运算符结果样式

【思考题】

如果给卡牌类加上旋转功能，允许在插入失败时进行旋转尝试，那么 append 函数的逻辑应该如何变化，简述设计思路。（代码最后，以注释方式解答）