

电子信息与通信学院

实 验 报 告

|  |  |
| --- | --- |
| 实验名称 | 扑克牌 |
| 课程名称 | C语言程序设计 |

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | Hale | 学号 | U2021XXX |

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 2022.2.8 | 地点 | 居家 |

|  |  |  |  |
| --- | --- | --- | --- |
| 成绩 |  | 教师 | 刘威教授 |

# 实验目的

学习使用C语言进行程序设计，理解基本程序设计思想，掌握C语言基

本代码规范、基础语法以及文件操作，培养大程序结构设计、多文件工程操作的能力。

# 实验环境

操作系统：Windows 10

编程工具：Dev-C++ 5.11

# 实验：扑克牌系列第八题

## 需求分析

在扑克牌系列第七题的基础上，修正逻辑，实现牌面值3最小、2最大；支持更多的组合类型、包括“多连对”、“三带一”、“飞机带翅膀”；并且支持更灵活的组合优先级设置。同时实现对整个牌局的记录和呈现。

## 数据结构设计

程序使用C语言编写，未使用C++的类，并且没有明显数据结构层面的设计，但是利用了结构体来存储基本信息并且基于数组实现了整个过程。

## 模块与流程设计

核心函数主要是：

getpriGroup、getMinGroup、getMinSeries、planCombs、getBiggerComb、getNextComb

上述六个函数是整个扑克牌的核心，在初始状态需要对手牌进行规划，于是调用planCombs函数对每个玩家进行规划，在此过程中，调用getMinGroup实现对“对子”、“单张”、“炸弹”的组合；调用getpriGroup函数实现对“多连对”、“三带一”、“飞机带翅膀”的组合；调用getMinSeries函数实现对“顺子”的组合；在整体牌局进行出牌时，不断调用getBiggerComb函数实现对牌局的进行更新，当别人都已经放弃出牌时，调用getNextComb函数实现赢家连续出牌直到出现某位玩家无牌可出，进而牌局结束。

## 代码实现

代码实现具体代码详见附件。

## 代码测试

单元测试（1）：getMinSeries函数

测试用例输入：1

测试结果：



结论：功能正常

单元测试（2）：getMinGroup函数获取炸弹组合

测试用例输入：3

测试结果：



结论：功能正常

单元测试（3）：getMinGroup函数获取对子组合

测试用例输入：3

测试结果：



结论：功能正常

单元测试（4）：getpriGroup函数获取“连对”组合

测试用例输入：2

测试结果：



结论：功能正常

单元测试（5）：getpriGroup函数获取“三带一”组合

测试用例输入：2

测试结果：



结论：功能正常

单元测试（6）：getpriGroup函数获取“飞机带翅膀”组合

测试用例输入：18

测试结果：

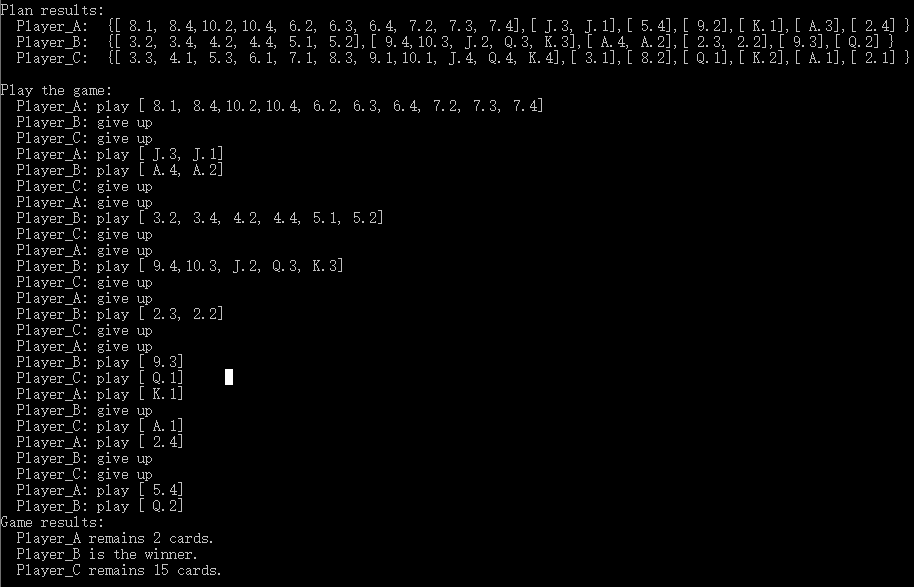


结论：功能正常

集成测试：

测试用例输入：18

测试结果：



结论：功能正常

# 实验中以及课程学习中的问题总结

根据自己心得体会设计二级标题，下面只是例子，可以修改。每个问题都应该有案例，有分析，有总结。

## 概述

在这次实验中，充分复盘了C语言课程中学习到的知识，对多文件程序结构的设计和大型模拟类型程序的实现有了一定程度的尝试，计算思维得到了提升，调试代码的能力和长时间编程的毅力得到了锻炼，最终成果实现完整，基本满足实验要求。

## 代码规范与调试技巧

本次代码实现沿用主流代码格式，缩进工整，但因文件编码不同，一些中文注释会在不同编译器上出现乱码，下次注释将尽可能使用英文进行；调试过程中，应该首先考虑根据提示是否是自己的拼写有误，之后判断是否数组越界等常见错误，最终考虑是否是自己代码设计的思路和逻辑是否出现错误。

## 后续学习与改进计划

在改进计划中，将使用C++进行面向对象编程，对本代码进行重构，并且实现一定的交互功能，让用户不单单只是输入一个数字，而是参与到过程中去，真正实现代码为人类服务的属性。而后续学习将继续阅读《Effective C++》来提升自己代码能力和编写代码的整洁程度。