# 项目阶段二：综合项目

【项目选取】

从项目指导书“项目阶段二：综合项目”的项目一和项目二中选取2个项目作为最终考核的项目，项目系数总和应大于等于1.0。若项目系数总和大于1.0则按1.0计算。

计算考核成绩：项目最终得分 = 项目考核所得总分\*系数

# 项目一 ：数据结构基础项目

**【项目概述】**

1 实验类别：设计类、创新类

2 学时要求：40学时

3 组织形式：独立实验

**【项目内容】**

从以下数据结构基础项目中，选择1~2个项目编码实现。

项目列表如下：

1.1 不带头结点的单链表的实现和相关操作。【系数0.5】

1.2 带有头结点的单链表的实现和相关操作。【系数0.5】

1.3 不带头结点的循环单链表的实现和相关操作。【系数0.5】

1.4 带有头结点的循环单链表的实现和相关操作。【系数0.5】

1.5 不带头结点的双链表的实现和相关操作。【系数0.5】

1.6 带有头结点的双链表的实现和相关操作。【系数0.5】

1.7 不带头结点的循环双链表的实现和相关操作。【系数0.5】

1.8 带有头结点的循环双链表的实现和相关操作。【系数0.5】

1.9 顺序循环队列的实现和相关操作。【系数0.5】

1.10 链式队列的实现和相关操作。【系数0.5】

1.11 顺序栈的实现和相关操作。【系数0.5】

1.12 链式栈的实现和相关操作。【系数0.5】

1.13 用单链表表示集合，解决集合的交、并、差等运算。【系数0.5】

1.14 一元多项式的运算。选择某种数据结构作为一元多项式的存储结构，并在此基础上完成以下运算（至少选择三个运算）：（1）给定变元的值，计算该一元多项式的值；（2）两个一元多项式进行加、减、乘、除（加、减、乘、除各算一个运算）。【系数0.5】

1.15 杨辉三角。杨辉三角大家都知道吧，现在要求用队列编程实现杨辉三角的输出。【系数0.5】

1.16 约瑟夫环问题。采用某种数据结构表示约瑟夫环：设编号为1，2，3，…,n的n（0<n<50）个人顺时针方向围成一个环，每人手中持有一个正整数密码。开始时任选一个正整数m作为报数上限，从编号为1的人开始起顺序报数，报到m时停止报数，同时，报m的出列，并将他持有的密码作为新的m值，从他的下一个人开始重新从1报数。如此下去，直到所有人出列为止。求出出列编号序列。【系数0.5】

1.17 饭桌上的报数游戏。设n（0<n<20）个人围着一张圆桌坐下，按顺时针方向给每人编号1，2，3，…,n。随机选定一个正整数m(m<n，且为素数。注：这个要求得由程序来确保)，一开始时，任意指定一人作为起始，从他开始从1起报数（当然是按顺时针方向向下），报到m及m的倍数的人喝一杯酒，该人停一轮报数。当每个m在使用过程中有3个人喝了酒，则必须换一个新的m（新m必须确保是以前没用过的），并从刚刚喝酒的那个人的下一个从1开始报数。如此下去，当有人喝到了他自己的第5杯酒时，游戏停止。程序要显示出所有人喝酒的杯数，同时将第一个喝到5杯酒的人指出来。【系数0.5】

1.18 二叉树的遍历。选择某种存储结构，建立一棵二叉树，并实现二叉树的4种遍历算法（先序、中序、后序、层序）。【系数0.5】

1.19 二叉树构造问题。根据输入的二叉树前序序列（以单个字符作为一个结点的信息）和中序序列，来构造一棵二叉树，然后输出该树的后序序列，以及该树中所有度为1的结点。【系数0.5】

1.20 另一个二叉树构造问题。根据输入的二叉树后序序列（以单个字符作为一个结点的信息）和中序序列，来构造一棵二叉树，然后输出该树的前序序列，以及该树中所有度为1的结点。【系数0.5】

1.21 哈夫曼树的构造。（1）从文件中读入字符集的个数n，以及这n个字符和它们对应的n个权值，构造一棵哈夫曼树；（2）利用该哈夫曼树对文件正文进行编码，将编码保存在文件中；（3）利用该哈夫曼树将编码结果进行译码，得到的结果保存在文件中。【系数1.0】

1.22 平衡二叉树（SB树或红黑树）的实现和常见操作。【系数1.0】

1.23 哈希表的实现和常见操作。【系数1.0】

1.24 跳表结构的实现。【系数1.0】

【项目选择要求】

如果单纯从“项目一”中选择两个系数都为0.5的数据结构基础项目，则其中一个项目必须按“学号尾数与题号相同”的原则固定下来，另一个项目则可从其余基础项目中任选一个。另一个项目也可以选择控制台应用项目中系数为0.5的项目。

【项目考核基本原则】

（1）“项目一”中的数据结构基础项目，要求实现该种数据结构本身比较完备的操作。

（2）如果是利用某种数据结构来解决问题，则需要对待解决的问题考虑比较全面。

# 项目二 ：控制台应用项目

**【项目概述】**

1 实验类别：设计类、创新类

2 学时要求：40学时

3 组织形式：独立实验

**【项目内容】**

从以下控制台应用项目中，选择1~2个项目编码实现。

项目列表如下：

* 1. 过河智力游戏的建模与解法。【基础系数0.5】
  2. 猜数字（就是几A几B的那个），学生按照固定接口写函数（能接受XAXB，能返回下一次的猜测），老师提供代码（能够产生随机数，获取学生猜测，并给出XAXB，统计猜测次数），这个小项目可比赛，使用全部720组(次序随机)比总猜测次数最少。【基础系数0.5】
  3. 八数码拼图，学生按照固定接口写函数（能接受3\*3的1-9随机顺序数组，返回0123构成的序列，0123代表左上右下移动）。老师提供代码（能够产生随机初始数组，检验学生返回的0123序列是否能够还原拼图，统计步骤），这个小项目可比赛，使用若干组随机数据，比总还原步骤最少。【基础系数0.5】
  4. Bmp文件数字水印(数字签名)，看谁做的漂亮，个性化展示。【基础系数0.5】
  5. 数独求解，教师程序出题、判题，学生程序答题。【基础系数0.5】
  6. 内部排序竞速，算法随便，不可调用qsort等函数，比int、double、字符串三种类型数组的排序，数据量从1万、10万、100万、1000万、1亿，每个数量级三组（随机、基本有序、基本逆序）。教师提供程序生成数据、验证排序正确性。【基础系数0.5】
  7. 外存排序竞速，不考虑操作系统，内缓存不超过1G，外存10G左右随机浮点数，排序。教师提供程序生成数据、验证排序正确性。如果考虑时间过长，可将排序数据量和允许使用缓存的大小降低数量级。【基础系数0.5】
  8. 21点游戏。【基础系数0.5】

# 控制台应用项目需求描述

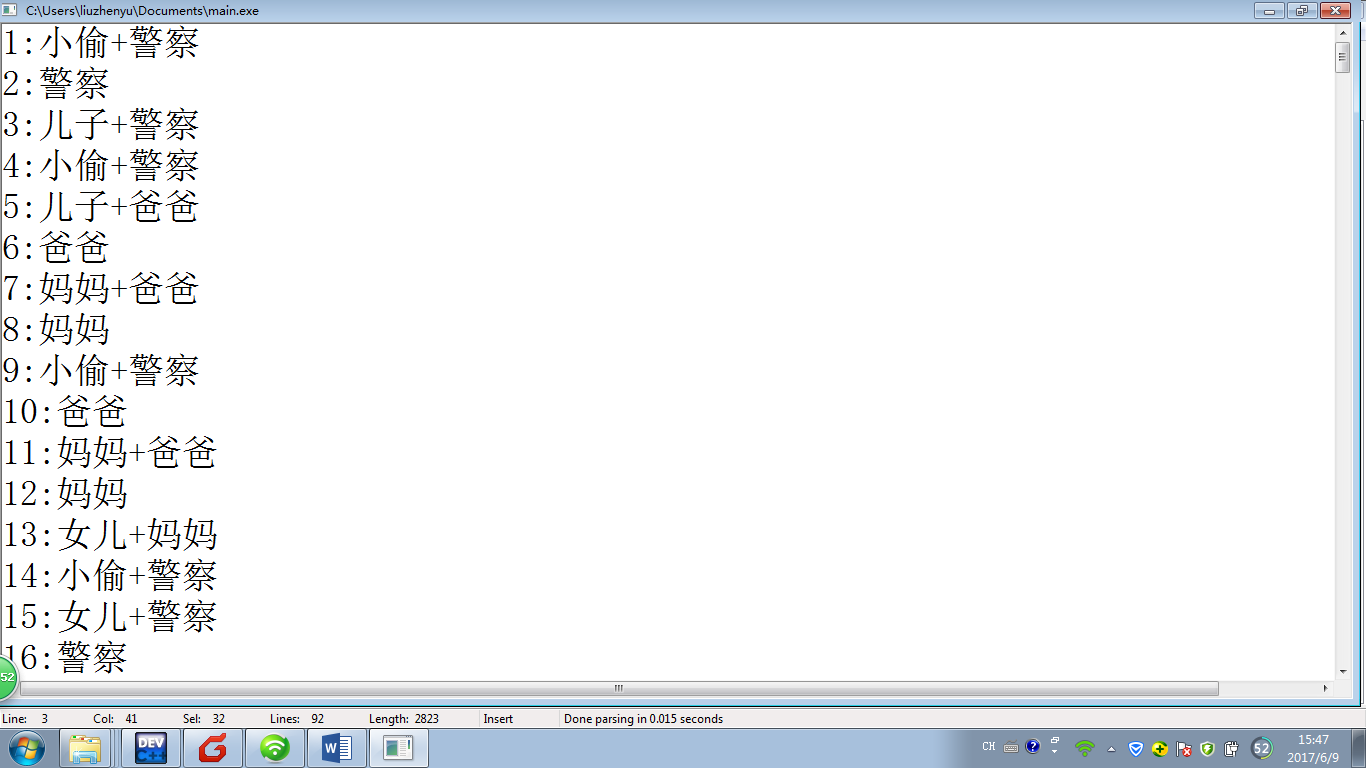
2.1.过河智力游戏

游戏规则：一家6口（爸爸、妈妈、两个女儿、两个儿子）及警察和小偷要从河这边渡到河对岸。在河这边仅有一艘小舢板可以把他们载到对岸。可是，只有爸爸、妈妈和警察能够驾船，不论成人与小孩，每程只能承载二人。在渡河过程中，你要避免以下三种情况的发生：

1. 当警察与小偷分开时，小偷会伤害一家6口；
2. 当爸爸看见妈妈离开时，爸爸便会教训女儿；
3. 当妈妈看见爸爸离开时，妈妈便会教训儿子。

游戏的玩法参见guohe.swf。

要求学生给出一组过河的可行解，例如：



本项目可锻炼学生建模能力、位运算的能力，养成使用枚举的习惯，算法方面锻炼“深搜”或“宽搜”实现能力。

2.2.猜数字

猜数字，计算机随机生成1个3位或4位的十进制整数，各个数位上的数字各不相同，用户给出猜测，计算机回答XAYB（X个位置和数字猜中，Y个数字猜中但位置错误），重复用户猜计算机回答的过程，直到猜中为止（3A0B，4A0B）。

这里，将给出若干组相关的数据供猜测（即第i把猜123，第j把不可能是123）。比谁的程序猜测的总次数最少。

文件“猜数字测试代码.txt”是给学生的，函数int guess(int A,int B)是学生需要完成的，其中参数A和B就是计算机根据上次猜测返回的结果，若是3A0B（以猜3个数为例），表示新一把猜测的开始。目前，guess中的代码，可供人机交互，学生提交时应该删除。

2.3.八数码拼图

规则：8数码拼图，9宫格中随机放置数字0-8，0作为空位可与上下左右的数字交换，最终排列为

012

345

678

要求学生利用搜索算法求解100组问题，得分计算公式为：总用时毫秒数/10.0+总步骤数。

为教师提供的代码中，分别使用标准宽搜和A\*算法完成了求解，使用标准宽搜可在100毫秒左右计算出最优解（最少步骤，一般为20步左右），如果使用启发式搜索A\*可提高求解速度，但步骤比最优解多（为老师提供的代码中，使用了哈密顿距离和逆序数两种启发函数，一般会在3毫秒左右计算完成，步骤60步-80步左右）。

本项目希望学生练习搜索算法，尽量提高计算速度，减少解的步骤，并在计算速度和步骤数中做一个折中。

提供的文件“8数码拼图测试代码.txt”，可在C环境下运行，可提供给学生测试，其中

typedef struct //图形，其中ij表示0（即空位）的位置

{

char a[N\*N];

char ij;

}G;

G表示图形的数据结构，a是N\*N的数组表示0----N\*N-1（N一般为3，当然学生要做更大的则更好），ij表示0的位置。

函数void init(G \*g)伪随机初始图形（这样不同学生比赛时，使用相同的测试数据），仅生成有解的（8数码问题9!个状态中，仅一半有解）。

函数void print(const G\* g)用于打印图形，可给学生调试用。

函数bool move(G\*g,int n)是移动空位的，参数n取值为0-3，分别代表左、上、右、下，这里的“左”实际是非0数向左，空位0向右（可以想象手机玩拼图时左滑操作）。返回值bool表示移动操作是否成功。

函数bool isok(const G\* g)用于测试是否达到最终状态；

函数int player(const G\*g,int \* step)是学生需要完成的函数，参数g表示待解的图形，参数step是写入学生答案的数组，里边按次序存放0-3即上下左右这些操作，返回值是step的长度，即步骤数，main函数中就利用这个step来测试学生提供的答案是否正确。目前，这个函数中的内容，不是求解过程，而是提供了人机交互接口，学生可以通过输入0-3玩一下拼图，但提交的代码应该将这些去掉。

int player(const G\*g,int \* step)

{

//学生代码从此开始，g是将初始图形，

//解决方法写在数组step中，里面就写0123表示左上右下，返回值为数组step的长度，即步数

int i=0;

G my=\*g;

while(!isok(&my))

{

int x;

print(&my);

scanf("%d",&x);

move(&my,x);

step[i++]=x;

}

return i;

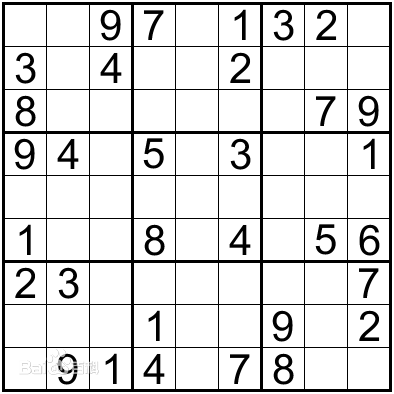
}

main函数中，随机产生M组初始数据，调用player函数求解，验证解的正确性，计时、计步骤等。

2.4.BMP文件加水印

BMP文件签名，要求学生自主学习24位BMP文件的格式，并编写程序实现能在不同尺寸的BMP文件上加上自己的名字或标志。

教师提供不同尺寸、不同背景颜色的图片，然后，主观评判或让学生投票，谁的程序签名签的好看。

2.5.数独游戏

游戏规则：数独是源自18世纪瑞士的一种数学游戏。玩家需要根据9×9盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫（3\*3）内的数字均含1-9，不重复。在本项目中，学生需要对1000道数独问题进行求解，在保证解的正确性的前提下，保证程序的运行时间少于模拟学生端程序（运行时间约200秒）。

### 数独生成程序(sudokuGenerate.cpp)

**生成数独的算法思想**：回溯法，递归实现深度优先搜索

**函数简介**：

* void randomFirstRow(char a0[], int n) /\*随机生成第一行\*/
* int Digit(char a[][COL], int i, int j)/\*递归填充一个数字\*/
* void createSudoku(char a[][COL])/\*封装后的数独生成函数\*/
* void createStartinggrid(const char a[][COL], char b[][COL], int numDigits)/\*随机生成初盘\*/
* int checkSudoku(const char a[][COL])/\*按规则检查函数\*/
* void printToFile(const char a[][COL], const char filename[])/\*打印数独数组到文件\*/
* void createSudokuToFile(void)/\*批量生成数独初盘到文件中\*/

**主要变量**：

#define MAXNUM 1000 /\*数独游戏个数\*/

#define SEED 20171212 /\*用于随机数生成的种子\*/

### 模拟学生端程序(sudokuStudentEnd.cpp)

**解决数独问题的算法思想**：与生成数独一致。

**函数简介**：

* int checkSudoku(const char a[][COL])/\*按规则检查函数\*/
* void printSudoku(const char a[][COL])/\*打印数独到屏幕\*/
* void readFromFile(char a[][COL],

const char filename[])/\*从一个文件中读取数独\*/

* int Solve(const char a[][COL],char b[][COL], int i,

int j)/\*递归解决数独\*/

* void solveSudoku(const char a[][COL],

char b[][COL])/\*封装后的解数独函数\*/

* 主函数负责遍历读取sudoku目录中的全部数独初盘，调用解数独函数，并计算时间，检查解的正确性，如果出错立刻输出错误提示并结束程序。最后输出求解全部数独问题花费的时间。

**主要变量**：

#define MAXNUM 1000 /\*数独游戏个数，初始阶段可以设置成100，即 只做前100道题\*/

## 项目资源：

(1)包含1000道数独问题的名为sudoku的文件夹。

(2)sudokuStudentEnd\_toBeSolved.cpp文件（学生需要补充readFromFile函数读取文件，补充solveSudoku函数求解数独问题）。

## 评分标准：

(1)正确求解1000道数独问题。（可得60%的分数）

(2)解正确的前提下，程序运行时间少于模拟学生端程序。（可得满分）

2.6.内部排序竞速

内部排序竞速，算法随便，不可调用qsort等函数，比int、double、字符串三种类型数组的排序，数据量从1万、10万、100万、1000万、1亿，每个数量级三组（随机、基本有序、基本逆序）。教师提供程序生成数据、验证排序正确性。

【得分标准】

（1）排序结果正确得到满分的0.5

（2）所有条件下排序总体耗时少于50分钟，得到满分的0.6

（3）如果能够考虑到利用多线程多核技术进行排序，得到满分的0.9(如果有学生采用的话)

（4）班级中所有选择该题目的学生，排序用时最少的得到满分的1.0，用时第二少的得到满分的0.9，第三少的得到满分的0.8，第四少的得到满分0.7，其余学生都参考准（1）、（2）

【项目资源】

（1）程序文件说明：

stdafx.h 标准库头文件

SortEngine.h 内排序核心功能的声明，包括生成随机数，标准排序方法等声明

SortEngine.cpp 内排序核心功能的定义

SortCompet.cpp 内排序裁判程序及主函数定义

GamePlayer.h 学生排序算法的声明

GamePlayer.cpp 学生排序算法的定义与实现

  （2）教师使用说明：

a. 学生写好自己的排序算法后，直接到SortCompet.cpp，Line 95，将该函数第一个参数换成GamePlayer1即可。

b. 可以将所有程序给学生，让学生参考裁判程序以及标准排序算法代码。

c. 需要给学生强调空间复杂度，归并排序使用时注意内存开销。

（3）学生使用说明：

a. 学生实现的算法仅限在GamePlayer.cpp中，留给学生的函数定义已经写好。

b. 学生不准修改定义好的函数结构，如果有自己特殊的排序算法函数的参数或返回类型等需求，可在GamePlayer.cpp中自己定义函数，使用定义好的函数调用学生自己写的排序算法函数。

c. 学生需要实现自己排序时所需的比较函数，作为函数指针传入。

2.7.外部排序竞速

外存排序竞速，不考虑操作系统，内缓存不超过1G，外存10G左右随机浮点数，排序。教师提供程序生成数据、验证排序正确性。如果考虑时间过长，可将排序数据量和允许使用缓存的大小降低数量级。

【问题及要求描述：】

对于256MB = 256 \* 1024 \* 1024 个无符号整数，以二进制文件流连续存储占用1GB硬盘空间，对这个文件进行由小到大排序，限制内存使用128MB。

【考核标准：】

检验内存需要老师亲自检查。

第一次分配内存是在 将1G文件 分成8次读取 每次读128MB的时候。

这时候会分配128MB的内存，每次利用这128MB内存，进行内排序。

这里内排序，学生可以随意调用排序。但如果使用归并内排时会产生额外的内存空间，老师酌情提醒吧。

第二次分配内存是在 已经分成8个有序段 做外排归并的时候。

这时候，

如果是直接归并 只会分配8个4字节的空间。

如果用败者树 会分配8\*2个4字节空间。

如果使用 缓冲区 那么会将 128MB - 败者树空间 的大小 做成8个缓冲区。

这里注意，千万不可以是将2个128MB直接放入内存归并，这样就超出内存了（PPT里有讲解）。

同时检查内存释放，避免泄露。

其实 可以让学生主动讲其内存分配的部分代码！

【得分标准】

（1）最基本标准 0.5

内存不超过128MB的前提下，排序文件正确（MD5与qresult\_data256MB.data的MD5相同）

（2）良好0.8

在1的标准上如果使用败者树 或者 哈夫曼树 优化，酌情加分！

（3）优秀1

在2的标准上如果使用缓冲区优化，给满分。

注意以下几点：

（1）文件IO都是二进制流。fopen时，第二个参数要带b。

（2）熟练使用 fread fwrite函数可参阅以下网址：

http://www.cplusplus.com/reference/cstdio/fread/

http://www.cplusplus.com/reference/cstdio/fwrite/

（3）如果调用C的Qsort int (\*compar)(const void\*,const void\*) 无符号整数不能直接用减法！

（4）如果带缓冲区，缓冲区一定是4字节的整数倍，以及每个缓冲区的游标，

以及8个有序文件是否读空的标记！缓冲区用malloc模拟二维数组实现。

2.8. 21点游戏

**一、规则：**

该游戏由2到6个人玩，使用除大小王之外的52张牌，游戏者的目标是使手中的牌的点数之和不超过21点且尽量大。21点一般用到1-8副牌。

扑克点数的计算规则是：2至9牌，按其原点数计算；K、Q、J和10牌都算作10点，记作T；A牌既可算作1点也可算作11点，当玩家停牌时，点数一律视为最大而尽量不爆，如A+9为20，A+4+8为13，A+3+A视为15。

开局时，庄家给每个玩家牌面向上发两张牌（明牌），再给庄家自己发两张牌，一张明牌，一张暗牌（牌面朝下）。

当所有的初始牌分发完毕后，如果玩家拿到的是A和T（无论顺序），就拥有黑杰克（Black Jack）；若庄家的明牌为T，且暗牌为A，应直接翻开并拥有Black Jack；如果庄家的明牌为A，则玩家可以考虑买不买保险（Insurance），保险金额是赌注的一半且不退（交给庄家），此时，如果庄家的暗牌为10点牌（构成Black Jack），那么翻开此牌，购买保险的玩家得到1倍赌注；如果庄家没有Black Jack则保持暗牌，玩家继续游戏。若玩家为Black Jack且庄家为其他，玩家赢得2倍赌注；若庄家为Black Jack且玩家为其他，庄家赢得赌注；若庄家和玩家均为Black Jack，平局，玩家拿回自己的赌注。

接下来是正常的拿牌流程：首名非黑杰克玩家选择拿牌（Hit）、停牌（Stand）、加倍（Double）或投降（Surrender，庄家赢得一半赌注）；所谓加倍：玩家在拿到前两张牌之后，可以再下一注与原赌注相等的赌金，然后只能再拿一张牌。若选择拿牌，则后续只能选择拿牌或停牌。在发牌的过程中，如果玩家的牌点数的和超过21，玩家就输了——叫爆掉（Bust），庄家赢得赌注（哪怕庄家也超过21点）。假如玩家没爆掉，又决定不再要牌了（停牌，或因加倍、投降而终止），则轮到下一名非黑杰克玩家选择。如果玩家违规，即给出了保险（用i表示）、拿牌（用h表示）、停牌（用s表示）、加倍（用d表示）和投降（用u表示）之外的动作符号，则算作违规，会扣除所有本次的赌注给庄家。

当所有未爆掉的玩家停止拿牌后，庄家翻开暗牌，并持续拿牌直至点数不小于17（若有A，按最大而尽量不爆计算）。假如庄家爆掉了，那他就输了，玩家赢得1倍赌注；否则那么比点数大小，大为赢。点数相同为平局，玩家拿回自己的赌注。

每位玩者和庄家之间的游戏都是独立的，因此在同一局内，庄家有可能会输给某些玩家，但也同时击败另一些玩家。

每次洗牌可以玩若干局，如果剩余的牌数不足总数的1/4，则在完成本局游戏后重新洗牌、切牌。

初始情况下每个玩家有1000元的赌资，每次可以下注的金额不低于10元。

满足以下条件之一的玩家结束游戏：

1. 所有赌资都已经输光；
2. 洗过5次牌，第6次洗牌时游戏停止；
3. 总局数达到玩家数量\*50局。

多人玩该游戏，相互之间PK，首先按照剩余赌资排序，剩余赌资多的胜，如果剩余赌资相等且都大于零，则并列；如果两个玩家的剩余赌资均为零，则比较轮次，轮次多的获胜，如果轮次相同，则并列。

**二、选取该项目的得分标准：**

1. 能够正常玩游戏（不违反游戏规则），得到基础的分值，即该项目本身分值的60%；
2. 能够根据当前牌型智能选择不同策略（保险、加倍、投降、拿牌、停牌），且能够保持对于不加任何策略，每次直接停牌的玩家90%以上的胜率的，可以增加20%；
3. 能够根据目前已经出过的所有牌，推断剩余牌中各个点数出现的概率，并相应调整不同策略的，且能够保持对于第（2）中情况玩家60%以上的胜率的，可以继续在（2）的基础上增加20%；
4. 能够提出更有效策略的，酌情加分。

关于程序的说明

项目一共包含5个文件：

* main.c：包含主程序；
* twentyone.h：包含游戏中所需要的各种结构体以及主要函数的声明；
* twentyone.c：包含游戏中所需要的主要函数的实现；
* mycodes.h：包含游戏中需要自行编写代码的函数的声明；
* mycodes.c：包含游戏中需要自行编写代码的函数的实现，自己的策略要写到该文件各个函数中。具体而言，该文件中包含四种动作，即四个函数，分别是（i可以取值1-5，代表最多5个玩家）：
* beti：玩家i的下注函数，每个玩家可以根据自己的策略下赌注，每次下注最少10元；
* first\_actioni：玩家i的首轮动作，每个玩家可以根据自己首次拿到的两张牌和庄家拿到的牌来根据自己的策略决定进行哪种操作（买保险、拿牌、停牌、加倍、投降）；
* continue\_actioni：玩家i的后续动作，每个玩家可以根据当前自己已有的牌决定下一步执行哪种操作（拿牌、停牌）；
* showi：玩家i看所有人的牌，可以根据这些信息决定下一轮的一些策略。

以上每个函数中都留出了相应的空间来供参与者写自己的策略，写自己的代码时要写到“#else”和“#endif”之间，其他地方的内容不要动。

目前每个函数以及主函数中都有条件编译选项，该选项打开时，可以人工玩。把mycodes.c文件中第二行的#define PERSON注释掉，则变成自动玩（根据自己提供的各种策略自动玩，最终满足了某种结束条件时会显示结果）。