

**信息与软件工程学院**

**项目报告**

课程名称： 程序设计项目实践（PBLF）

学 期： 2021-2022第1学期

项目名称： 烧脑游戏串烧

授课教师： 陈伟

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 学号 | 姓名 | 贡献评估 |
| 1（组长） | 2021090906016 | 刘芮利 | A+ |
| 2 | 2021090906020 | 张庭恺 | A+ |
| 3 | 2021090906002 | 赵晗 | A+ |
| 4 | 2021090906010 | 钟城伟 | A+ |
| 5 | 2021090906008 | 黄培坤 | A+ |

目录

[1 项目简介 3](#_Toc92043022)

[1.1 考核方式 3](#_Toc92043023)

[1.2 项目题目及内容简介 3](#_Toc92043024)

[1.3 项目组成员与分工 4](#_Toc92043025)

[2 需求分析 4](#_Toc92043026)

[2.1 选题的依据 4](#_Toc92043027)

[2.2 功能需求 5](#_Toc92043028)

[3 系统设计 6](#_Toc92043029)

[3.1 总体设计 6](#_Toc92043030)

[3.2 模块设计 7](#_Toc92043031)

[4 系统实现 8](#_Toc92043032)

[4.1 主程序（游戏框架） 8](#_Toc92043033)

[4.1.1 所有函数及其变量声明 8](#_Toc92043034)

[4.1.2 读取数据 10](#_Toc92043035)

[4.1.3 写入数据 11](#_Toc92043036)

[4.1.4 注册新用户 12](#_Toc92043037)

[4.1.5 登录 13](#_Toc92043038)

[4.1.6读取密码 14](#_Toc92043039)

[4.1.7 输出Logo 14](#_Toc92043040)

[4.1.8 颜色控制 15](#_Toc92043041)

[4.1.9 排名系统 15](#_Toc92043042)

[4.1.10 主菜单 16](#_Toc92043043)

[4.1.11 主函数 17](#_Toc92043044)

[4.2 分支程序（游戏和加解密模块） 18](#_Toc92043045)

[4.2.1 加解密模块 18](#_Toc92043046)

[4.2.2 猜数游戏 19](#_Toc92043047)

[4.2.3 取石子博弈 23](#_Toc92043048)

[5 测试 28](#_Toc92043049)

[6 总结 32](#_Toc92043050)

# 1 项目简介

## 1.1 考核方式

总成绩 = 项目和项目文档成绩(40%) + 汇报幻灯片成绩(20%)

+ 表达能力(20%) + 团队合作(20%)

## 1.2 项目题目及内容简介

* 项目题目：烧脑游戏串烧（小游戏合集）
* 内容简介：项目成果是一款益智游戏主题的小游戏合集，具体包括猜数游戏、猜拳游戏、取石子博弈、吃豆人、长城游戏、俄罗斯方块，每一个游戏有多个关卡和不同模式可供选择。支持多用户注册、登录和PK，天梯排名系统可以显示所有用户的游戏得分，同时每个游戏也可以查看历史得分记录排名。游戏数据使用文件保存和读取，为了保障数据安全，程序数据采用RSA非对称加密算法对游戏数据进行加密。每个游戏的简要介绍如下：
  + - 猜数游戏：玩家猜测程序随机生成的数字，有一定次数限制。根据猜测次数和用时计算得分。
    - 猜拳游戏：玩家与电脑进行猜拳，通过模拟石头剪刀布来实 现。玩家可进行多次游戏来对自己今天的运气进行没有科学依据的判断。
    - 取石子博弈：玩家和AI每次可从任意一堆石子里取出任意多枚 石子扔掉，可以取完，不能不取，每次只能从一堆里取。取走最 后一颗石子的一方获得胜利。根据AI难度、撤回次数和用时计算得分。
    - 吃豆人：玩家可使用键盘操控吃豆人躲避敌人攻击获取分数，可通过增幅道具增加速度
    - 长城游戏：在NN方阵中有N个‘/’，请玩家数出移动‘/’使之形成一条直线“长城”的最小步数。答案正确则获胜。有三次回答机会，并计时，所用时间越少或回答关卡数越靠后分数越高。
    - 俄罗斯方块：玩家可以进行四种难度模式选择，难度越大即俄罗斯方块下落速度越快，满行即清除并且记1000分

## 1.3 项目组成员与分工

成员：刘芮利、赵晗、黄培坤、张庭恺、钟城伟

刘芮利：游戏框架设计、头文件设计与编写、代码后期合成、登录/注册系统、关卡切换、天梯排名系统、加解密模块、数据存储、猜数游戏编写、取石子博弈游戏编写

赵晗： 猜拳游戏编写（使用Windows.h头文件实现对system和sleep函数的应用，使程序更具有现实性和灵动性;独立设计case\_win和echoinput两个函数对主函数进行“瘦身”;利用时间作种子生成随机数实现真随机）

黄培坤：编写小游戏俄罗斯方块，学习并使用颜色函数，解决碰撞判断法性问题以及研究方块旋转逻辑

张庭恺：吃豆人游戏的编写

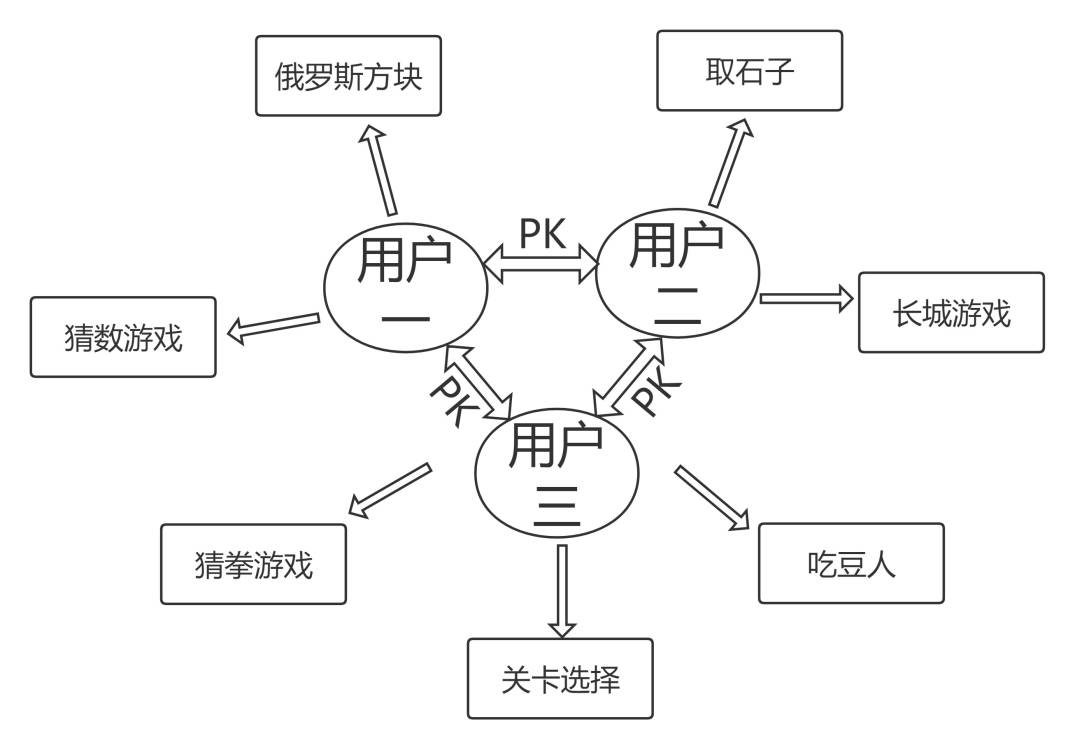
钟城伟：长城游戏编写（包括1.最短步数的计算，等价于最佳匹配，使用了KM算法，大大缩短了时间复杂度，由缩减为。2.作了获胜或失败后的美化。3.追踪生成答案方阵。4.利用字符数组读入用户输入，再进行判断，增强程序的排错能力，避免出现bug。5.使用时间函数，以达到计时目的。6.巧妙使用多个goto语句，以达成选择性循环。）

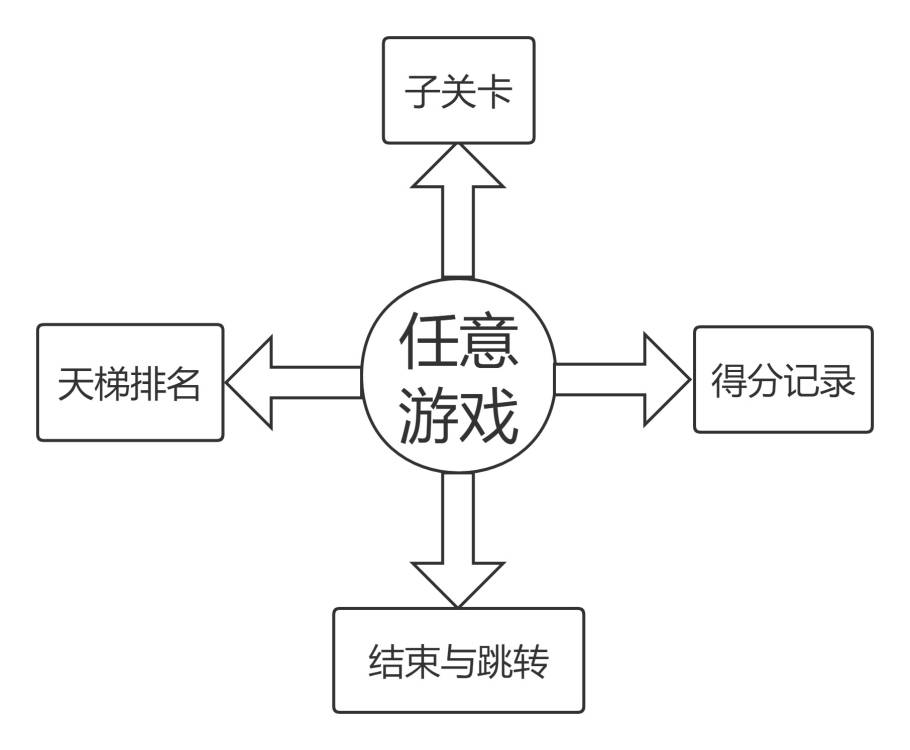
# 2 需求分析

## 2.1 选题的依据

* 俄罗斯方块、Nim博弈、吃豆人等都是经典小游戏，通过这学期所学知识，复现经典小游戏提高C语言代码能力，同时可供闲暇之余消遣娱乐
* 多用户登录/注册、用户得分排名、历史得分记录等是小游戏体验的基本需要，项目对这些功能进行了简易还原，以此提高了游戏的可玩性和竞技性
* 为防止游戏数据被恶意篡改，数据存储安全对程序正常、可靠的运行至关重要。游戏采用现今最广泛运用的RSA非对称加密算法对数据进行加解密，保障数据安全

## 2.2 功能需求

下面通过UML示例图说明程序功能：



# 3 系统设计

## 3.1 总体设计

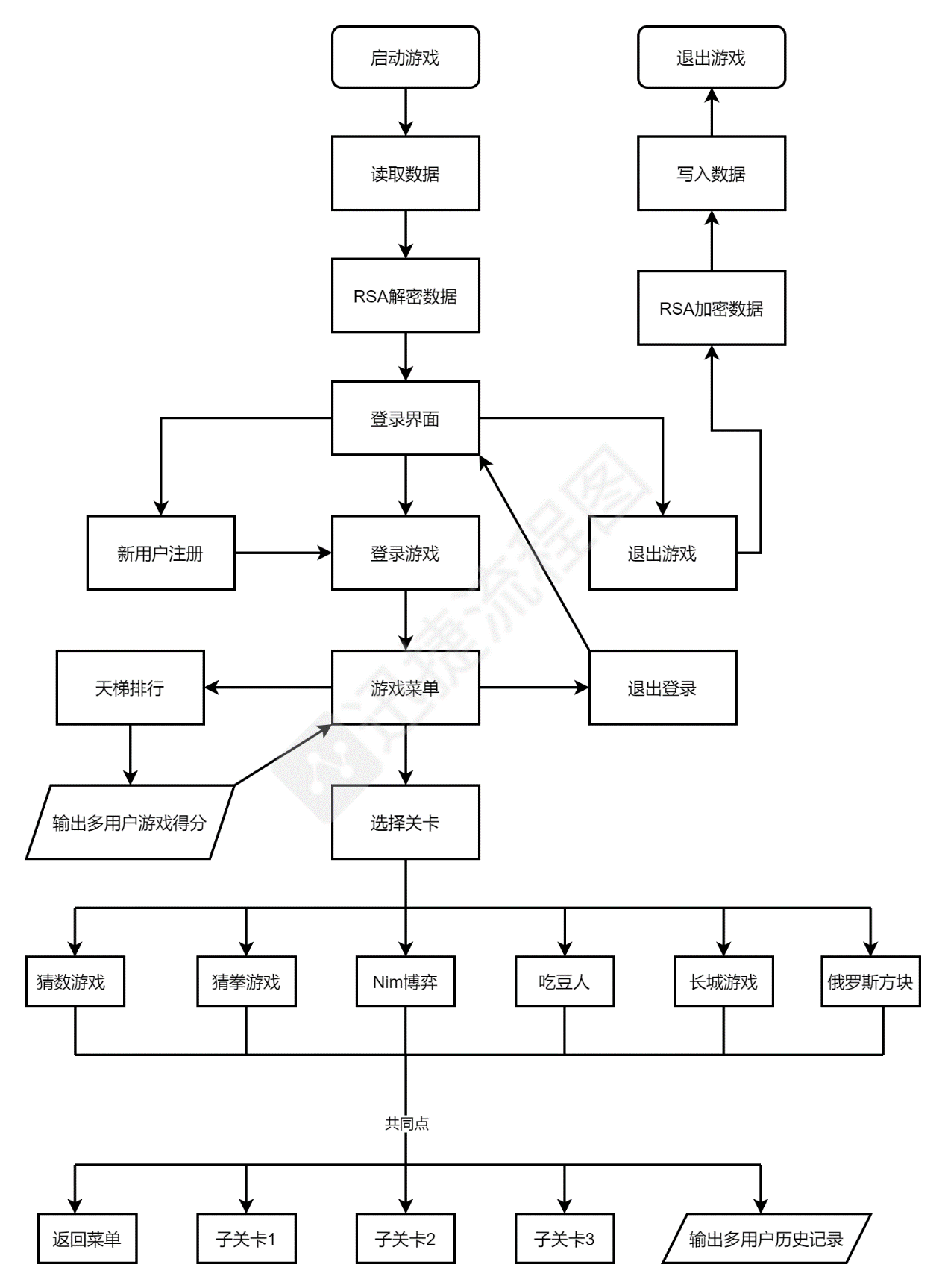
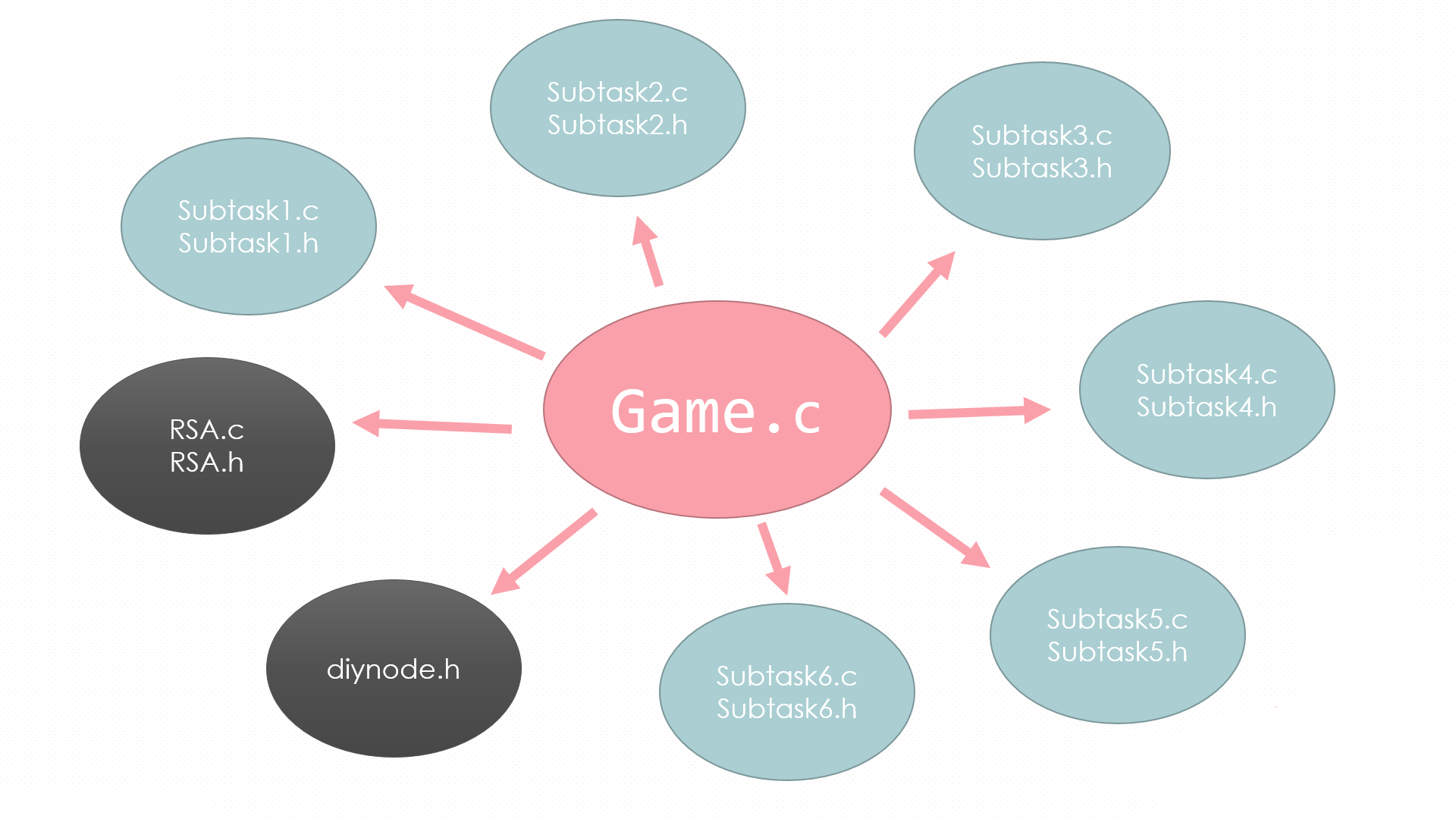


图3-1

## 3.2 模块设计



* **Subtask.h**

1. int Max(int, int); *//求最小值函数*
2. int Min(int, int); *//求最大值函数*
3. void Switchtask(); *//切换关卡函数*
4. void Subtask3(); *//你的游戏函数入口，这里以Subtask3举例*
5. void ShowRank(int); *//展示对应游戏用户排名*

* **Subtask.c**

1. #include "Subtask2.h" *//引入你的头文件，这里以Subtask2举例*
2. #include "diynode.h"  *//引入自定义结构体类型,*
3. #include<stdio.h>
4. extern node Data[]; *//共享游戏数据Data*
5. extern int id, taskid; *//共享id, taskid, id为当前用户对应的数字序号, 用作Data[]下标, taskid表示当前正在进行或即将进入的关卡, 你很可能会用到*
6. *//你可以在下面写函数，定义变量等等*
7. void Subtask2(){ *//下面将是你的游戏入口, 确保return后将taskid更改为合适的值, 1~Gamecnt表示对应关卡, -1表示退出游戏*
8. puts("Subtask2:"); *//这只是一个演示测试而已, 建议删除*
9. Data[id].taskid = Data[id].taskid > 2 ? Data[id].taskid : 2; *//更新用户的最高关卡记录*
10. Switchtask(); *//切换关卡, 可在退出时调用*
11. }

* **Diynode.h**

1. #define GameCnt 8
2. typedef struct Node{ *//存储游戏数据的自定义结构体*
3. char username[12];
4. char password[35]; int taskid;
5. int data[GameCnt+5][10];
6. }node;

**注：以上模块为每个成员共用的代码模板，便于统一共享游戏数据和后期代码整合**

# 4 系统实现

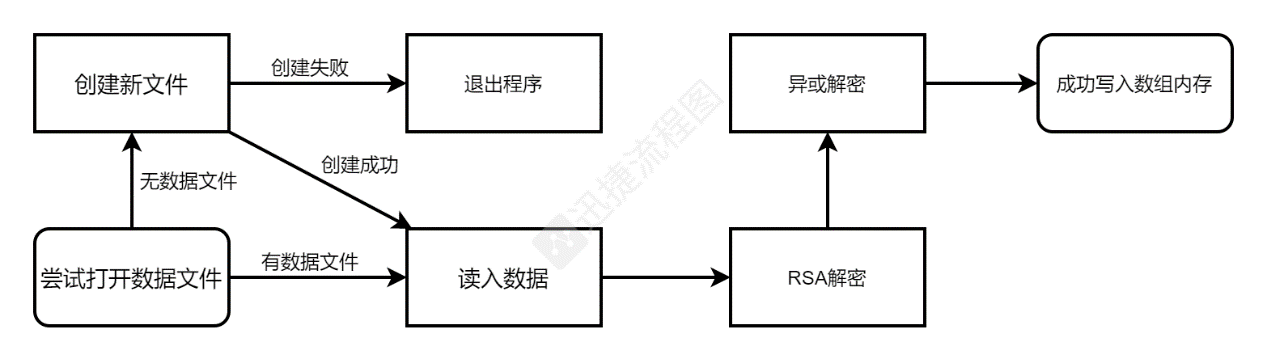
## 4.1 主程序（游戏框架）

整个游戏程序的主程序(Game.c)是游戏的核心框架，它作为一条线索将6个分散的小游戏串联成为一个有机整体。设计初衷是开发一个闯关式游戏，玩家每通过一个关卡便可开启下一个关卡，同时游戏拥有注册、登录、数据存储的功能。每个组员负责1~2个小游戏，采用多文件编写，统一数据存储规范，最后由我合并为1个完整的游戏合集。主程序的代码由我百分之百原创编写，当然起初也有很多设计缺陷，后来经过5次改动较大的迭代更新，不断完善，最终添加了加密模块、得分排名、密码遮掩、多子关卡等功能，程序抵抗非法数据输入的鲁棒性也变得更强。

### 4.1.1 所有函数及其变量声明

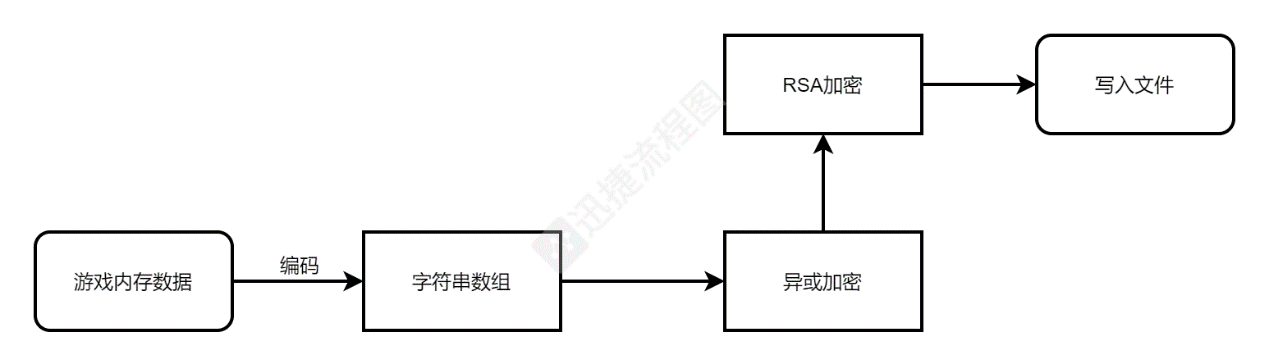
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. #include <time.h>
5. #include <windows.h>
6. #include <conio.h>
7. #include "diynode.h"
8. #include "Subtask1.h"
9. #include "Subtask2.h"
10. #include "Subtask3.h"
11. #include "Subtask4.h"
12. #include "Subtask5.h"
13. #include "Subtask6.h"
14. #include "RSA.h"
15. #define GameCnt 8
16. const long long mask = 3986329060925346962LL; *//密码加密掩码*
17. node Data[55]; *//游戏数据*
18. static char wr[1000005];
19. int id; *//n为用户数量, id为当前用户对应的数字序号, 用作Data[]下标, taskid表示当前正在进行或即将进入的关卡*
20. static int n, taskid;
21. static FILE \*io; *//文件指针*
22. static char graphtext[10][1005] = {
23. "|  \_\_\_\_\_\_              \_         \_\_\_\_\_\_                      \_                 \_\_\_\_\_                                    |\n",
24. "|  | \_\_\_ \\            (\_)        | \_\_\_ \\                    (\_)               |  \_\_ \\                                   |\n",
25. "|  | |\_/ / \_ \_\_  \_\_ \_  \_  \_ \_\_   | |\_/ / \_   \_  \_ \_\_  \_ \_\_   \_  \_ \_\_    \_\_ \_  | |  \\/  \_\_ \_  \_ \_\_ \_\_\_    \_\_\_  \_\_\_       |\n",
26. "|  | \_\_\_ \\| '\_\_|/ \_` || || '\_ \\  | \_\_\_ \\| | | || '\_\_|| '\_ \\ | || '\_ \\  / \_` | | | \_\_  / \_` || '\_ ` \_ \\  / \_ \\/ \_\_|      |\n",
27. "|  | |\_/ /| |  | (\_| || || | | | | |\_/ /| |\_| || |   | | | || || | | || (\_| | | |\_\\ \\| (\_| || | | | | ||  \_\_/\\\_\_ \\      |\n",
28. "|  \\\_\_\_\_/ |\_|   \\\_\_,\_||\_||\_| |\_| \\\_\_\_\_/  \\\_\_,\_||\_|   |\_| |\_||\_||\_| |\_| \\\_\_, |  \\\_\_\_\_/ \\\_\_,\_||\_| |\_| |\_| \\\_\_\_||\_\_\_/      |\n",
29. "|                                                                       \_\_/ |                                           |\n",
30. "|                                                                      |\_\_\_/                                            |\n",
31. };
32. int Max(int a, int b); *//求最大值*
33. int Min(int a, int b); *//求最小值*
34. static int findid(const char \*s);*//查询用户*
35. void Switchtask();*//切换关卡*
36. static void WriteData();*//写入数据*
37. static void ReadData();*//读取数据*
38. static int ReadPassword(char \*a, int op);*//输入密码，自动遮掩*
39. static void Register();*//注册新用户*
40. static int Login();*//登录*
41. static void print\_logo();*//打印Logo*
42. static void SetColor(int color);*//颜色控制*
43. static inline int calc(int i, int op);*//计算某一关键字分数*
44. static void quicksort(int l,int r, int op);*//快速排序*
45. void ShowRank(int j);*//（对某个游戏）展示排名*
46. static void MainMenu();*//主菜单*

### 4.1.2 读取数据



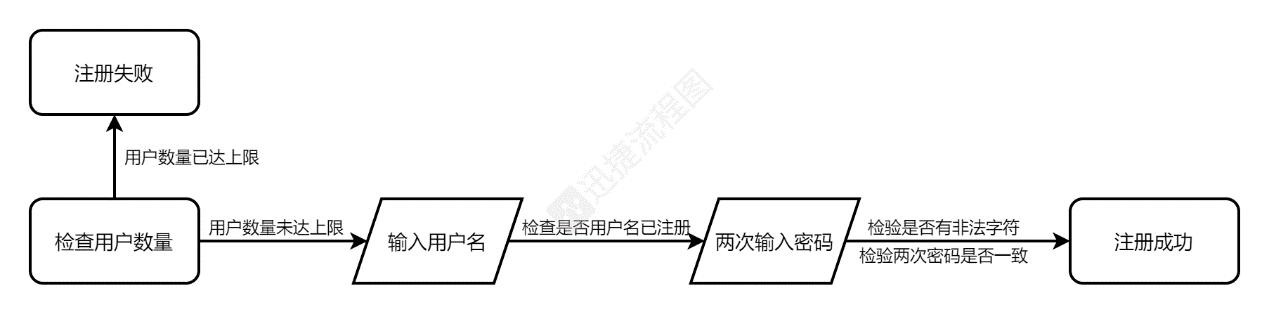
1. static void ReadData(){
2. io = fopen("./data.txt", "r");
3. if(io == NULL){
4. *//puts("Unexpected Error ??? \n(If first run, please try again!)");*
5. io = fopen("./data.txt", "w"); fclose(io); *//尝试创建新文件*
6. io = fopen("./data.txt", "r");
7. if(io == NULL){
8. puts("Unexpected Error ??? \n(If first run, please try again!)");
9. exit(0);
10. }
11. }
12. int rd\_n = 0;long long x; fscanf(io, "%d", &rd\_n);
13. for(int i = 0; i < rd\_n; ++i){
14. fscanf(io, "%lld", &x);
15. x = Decode(x); *//解密*
16. wr[i] = x ^ 170;
17. }
18. fclose(io);
19. FILE \*tempio = fopen("./tempdata.txt", "w");
20. fputs(wr, tempio);
21. fclose(tempio);
22. tempio = fopen("./tempdata.txt", "r");
23. fscanf(tempio, "%d", &n);
24. for(int k = 1; k <= n; ++k){
25. fscanf(tempio, "%s%s%d", Data[k].username, Data[k].password, &Data[k].taskid);
26. for(int i = 1; i <= GameCnt; ++i){
27. fscanf(tempio, "%d", &Data[k].data[i][0]);
28. for(int j = 1; j <= Data[k].data[i][0]; ++j) fscanf(tempio, "%d", &Data[k].data[i][j]);
29. }
30. }
31. fclose(tempio);
32. remove("./tempdata.txt");
33. }

### 4.1.3 写入数据



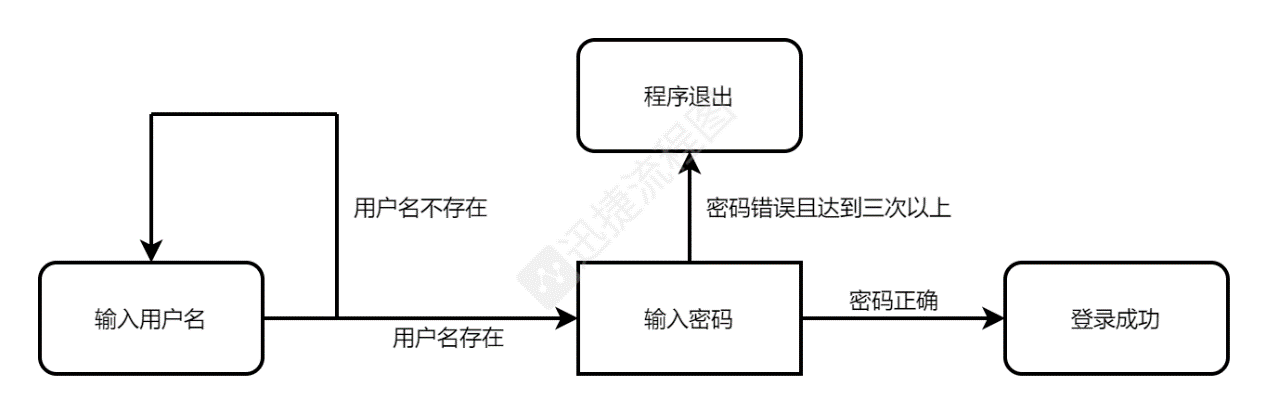
1. static void WriteData(){ *//保存，写入数据*
2. memset(wr, 0, sizeof(wr)); char \*p = wr;
3. p += sprintf(p, "%d\n", n);
4. for(int k = 1; k <= n; ++k){
5. p += sprintf(p, "%s %s %d\n", Data[k].username, Data[k].password, Data[k].taskid);
6. for(int i = 1; i <= GameCnt; ++i){
7. p += sprintf(p, "%d", Data[k].data[i][0]);
8. for(int j = 1; j <= Data[k].data[i][0]; ++j)
9. p += sprintf(p, " %d", Data[k].data[i][j]);
10. *//fputc('\n', io);*
11. p += sprintf(p, "\n");
12. }
13. }
14. io = fopen("./data.txt", "w");
15. int len = strlen(wr); fprintf(io, "%d\n", len);
16. for(int i = 0; i < len; ++i){
17. long long x = Encode(wr[i] ^ 170); *//加密*
18. fprintf(io, "%lld ", x);
19. }
20. fclose(io);
21. }

### 4.1.4 注册新用户



1. static void Register(){
2. if(n >= 50){
3. puts("用户数量已达上限吗，请联系管理员处理数据！");
4. system("pause"); return;
5. }
6. ++n; char s[34];
7. while(1){
8. printf("Username: "); memset(Data[n].username, 0, sizeof Data[n].username);
9. fflush(stdin); scanf("%10s", Data[n].username); fflush(stdin);
10. int flag = 0;
11. for(int i = 1; i <= n - 1; ++i)
12. if(strcmp(Data[n].username, Data[i].username) == 0){
13. printf("This username has been existed!\n");
14. flag = 1; break;
15. }
16. if(flag) continue;
17. while(ReadPassword(Data[n].password, 0) == 0);
18. while(ReadPassword(s, 1) == 0);
19. if(strcmp(Data[n].password, s)){
20. puts("两次输入密码不一致，请重试！");
21. continue;
22. }
23. Data[n].taskid = 0;
24. printf("注册成功，你的Username为 %s 请牢记！\n", Data[n].username); Sleep(2500);
25. break;
26. }
27. }

### 4.1.5 登录



1. static int Login(){
2. if(n == 0){
3. puts("No data availible!");
4. system("pause"); return 0;
5. }
6. printf("Username:");
7. char username[10];
8. fflush(stdin); scanf("%10s", username);
9. id = findid(username);
10. if(id == -1){
11. puts("This username is not existed.");
12. system("pause"); return 0;
13. }
14. *//printf("Password:");*
15. char st[35];
16. *//scanf("%lld", &x);*
17. *//if(x == (Data[id].password ^ mask)){*
18. for(int i = 1; i <= 3; ++i){
19. memset(st, 0, sizeof(st)); ReadPassword(st, 0);
20. if(strcmp(st, Data[id].password) == 0){
21. puts("登录成功！"); Sleep(1500);
22. return 1;
23. } else{
24. puts("Wrong Password!!!");
25. }
26. }
27. for(int i = 3; i >= 1; --i){
28. printf("输入密码错误达3次，程序将在 %d 秒后退出...", i);
29. Sleep(1000);
30. for(int j = 1; j <= 41; ++j)
31. printf("\b \b");
32. }
33. WriteData(); system("pause"); exit(0);
34. }

### 4.1.6读取密码

1. static int ReadPassword(char \*a, int op){
2. if(op == 0) printf("Password:");
3. else printf("Password[again]:");
4. int i = 0; memset(a, 0, sizeof a);
5. while (1) {
6. char c = getch();
7. if (c == '\r') { *//输入回车就退出循环*
8. break;
9. }
10. if (c == '\b') { *//delete键*
11. if (i > 0) {
12. a[--i] = '\0';
13. printf("\b"); *//输出\b，让光标回退一格*
14. printf(" \b");  *//输出空格覆盖掉要删除的字符，然后输出\b再把光标回退。*
15. }
16. continue;
17. }
18. if(c == ' '){
19. printf("\n请勿输入非法字符\n");
20. putchar('\n'); return 0;
21. }
22. if(i == 30){
23. printf("\nPassword is too long, please try again!\n");
24. putchar('\n'); return 0;
25. }
26. a[i++] = c;
27. printf("\*");
28. }
29. putchar('\n');
30. return 1;
31. }

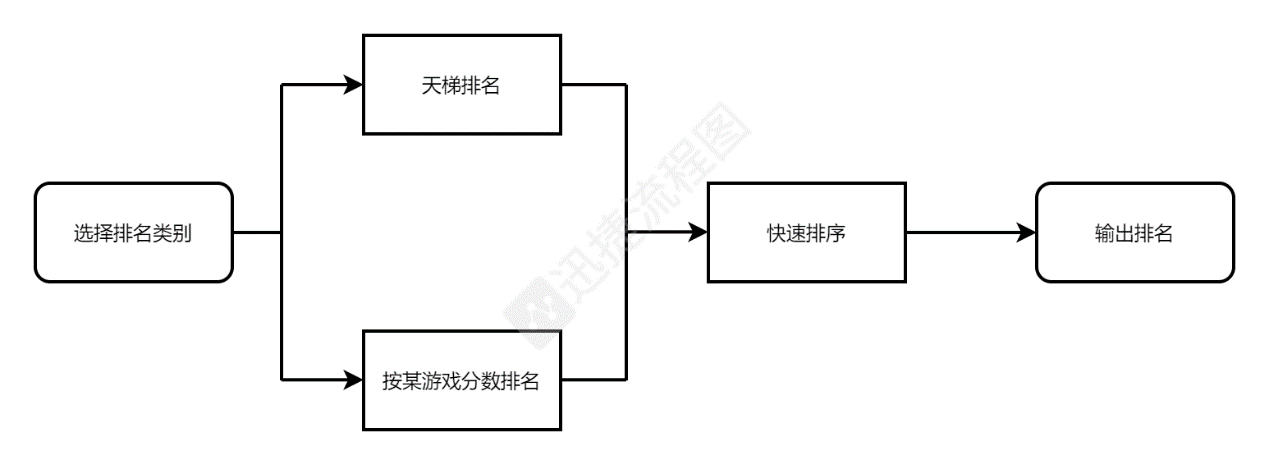
### 4.1.7 输出Logo

1. static void print\_logo(){
2. printf("╭=====================================================○●○●==========================================================╮\n");
3. printf("|                                          ★ ☆欢迎来玩烧脑游戏串烧☆ ★                                               |\n");
4. for(int i = 0; i < 8; ++i)
5. for(int j = 0; j < strlen(graphtext[i]); ++j)
6. putchar(graphtext[i][j]);
7. printf("╰=====================================================○●○●==========================================================╯\n");
9. }

### 4.1.8 颜色控制

1. static void SetColor(int color){
2. HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);
3. SetConsoleTextAttribute(hConsole, color);
4. }

### 4.1.9 排名系统

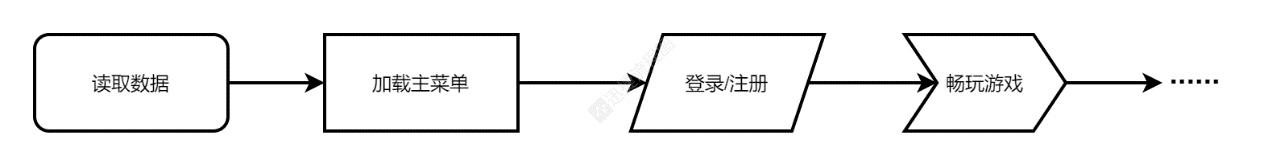


1. static inline int calc(int i, int op){
2. if(op == 0){
3. int ret = 0;
4. for(int j = 1; j <= GameCnt; ++j) ret += Data[i].data[j][1];
5. return ret;
6. }
7. else return Data[i].data[op][1];
8. }
9. static void quicksort(int l,int r, int op)
10. {
11. int mid = (l + r) >> 1;*//中间数*
12. int i = l, j = r;
13. do{
14. while(calc(i, op) > calc(mid, op)) i++; *//calc(int, int)支持不同关键字排名*
15. while(calc(j, op) < calc(mid, op)) j--;
16. if(i <= j){
17. if(id == i) id = j; *//id记着一起变*
18. else if(id == j) id = i;
19. node tmp = Data[i]; Data[i] = Data[j]; Data[j] = tmp;
20. i++; j--;
21. }
22. }while(i <= j);
23. if(l < j) quicksort(l, j, op);
24. if(i < r) quicksort(i, r, op);
25. }
26. void ShowRank(int j){
27. quicksort(1, n, j);
28. for(int i = 1; i <= n; ++i)
29. printf("\t\t\t\t   %10s   %4d   %4d   %7d   %7d   %7d   \n", Data[i].username, i, Data[i].data[j][1], Data[i].data[j][3], Data[i].data[j][4], Data[i].data[j][5]);
30. }

### 4.1.10 主菜单

1. static void MainMenu(){
2. int op;
3. while(1){
4. system("cls");
5. SetColor(0x0b);
6. print\_logo();
7. SetColor(0x0e);
8. puts("Welcome!");
9. puts("你有以下3个选择：");
10. puts("1.登录已有账号");
11. puts("2.注册一个新用户");
12. puts("3.退出游戏");
13. printf("请输入你的选择：");
14. op = -1; fflush(stdin); scanf("%d", &op); fflush(stdin);
15. if(op < 1 || op  > 3) {puts("不合法操作！！！"); system("pause"); }
16. else if(op == 1 && Login(io)) break;
17. else if(op == 2){Register(io); continue;}
18. else if(op == 3){WriteData(io); exit(0);}
19. }
20. }

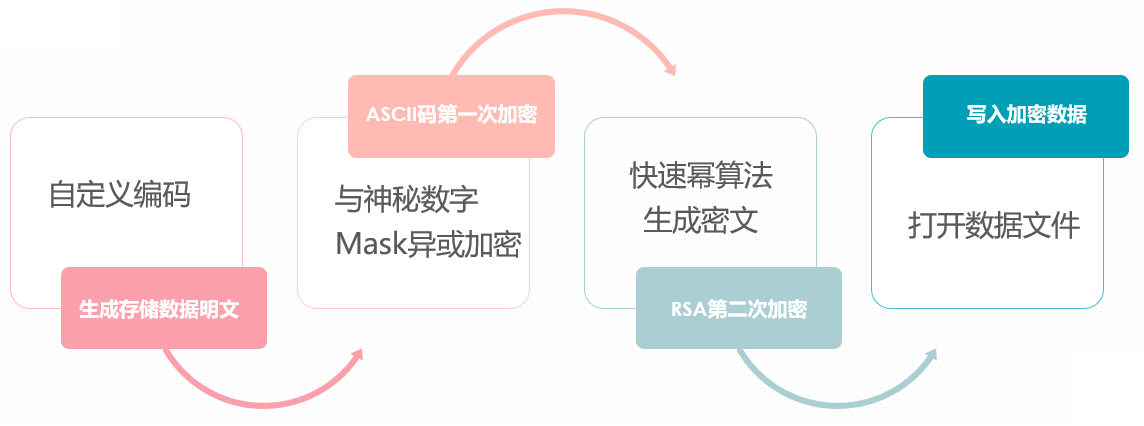
### 4.1.11 主函数



1. int main(){
2. //system("chcp 65001 >nul");
3. //system("color f0");
4. system("title 烧脑游戏串烧");
5. system("mode con lines=30 cols=125");
6. int op;
7. ReadData();
8. MainMenu();
9. while(1){
10. system("title 烧脑游戏串烧");
11. system("mode con lines=30 cols=125");
12. system("cls"); SetColor(0x0b); print\_logo(); SetColor(0x0e);
13. //printf("╭=============○●○●=======╮\n");
14. //printf("│  ★ ☆头脑风暴大作战☆ ★  │\n");
15. //printf("╰=======○●○●=============╯\n");
16. printf("1.选择关卡\n2.天梯排行\n3.退出登录\n");
17. printf("请输入你的选择:");  op = -1;
18. fflush(stdin); scanf("%d", &op); fflush(stdin);
19. if(op == 1){
20. Switchtask();
21. if(taskid == 1) Subtask1();
22. else if(taskid == 2) Subtask2();
23. else if(taskid == 3) Subtask3();
24. else if(taskid == 4) Subtask4();
25. else if(taskid == 5) Subtask5();
26. else if(taskid == 6) Subtask6();
27. else continue;
28. } else
29. if(op == 2){
30. quicksort(1, n, 0); system("cls");
31. printf("当前用户: %s\n", Data[id].username);
32. printf("\t\t\t\t     Username  名次  总分  猜数游戏  猜拳游戏  LRL52的博弈  吃豆人  长城游戏  俄罗斯方块\n");
33. for(int i = 1; i <= n; ++i)
34. printf("\t\t\t\t   %10s  %4d  %4d  %8d  %8d  %11d  %6d  %8d  %10d\n", Data[i].username, i, calc(i, 0), Data[i].data[1][1], Data[i].data[2][1], Data[i].data[3][1], Data[i].data[4][1], Data[i].data[5][1], Data[i].data[6][1]);
35. system("pause");
36. }
37. else if(op == 3) MainMenu();
38. else { puts("不合法操作！！！"); system("pause"); }
39. }
40. //WriteData(io);
41. return 0;
42. }

## 4.2 分支程序（游戏和加解密模块）

### 4.2.1 加解密模块



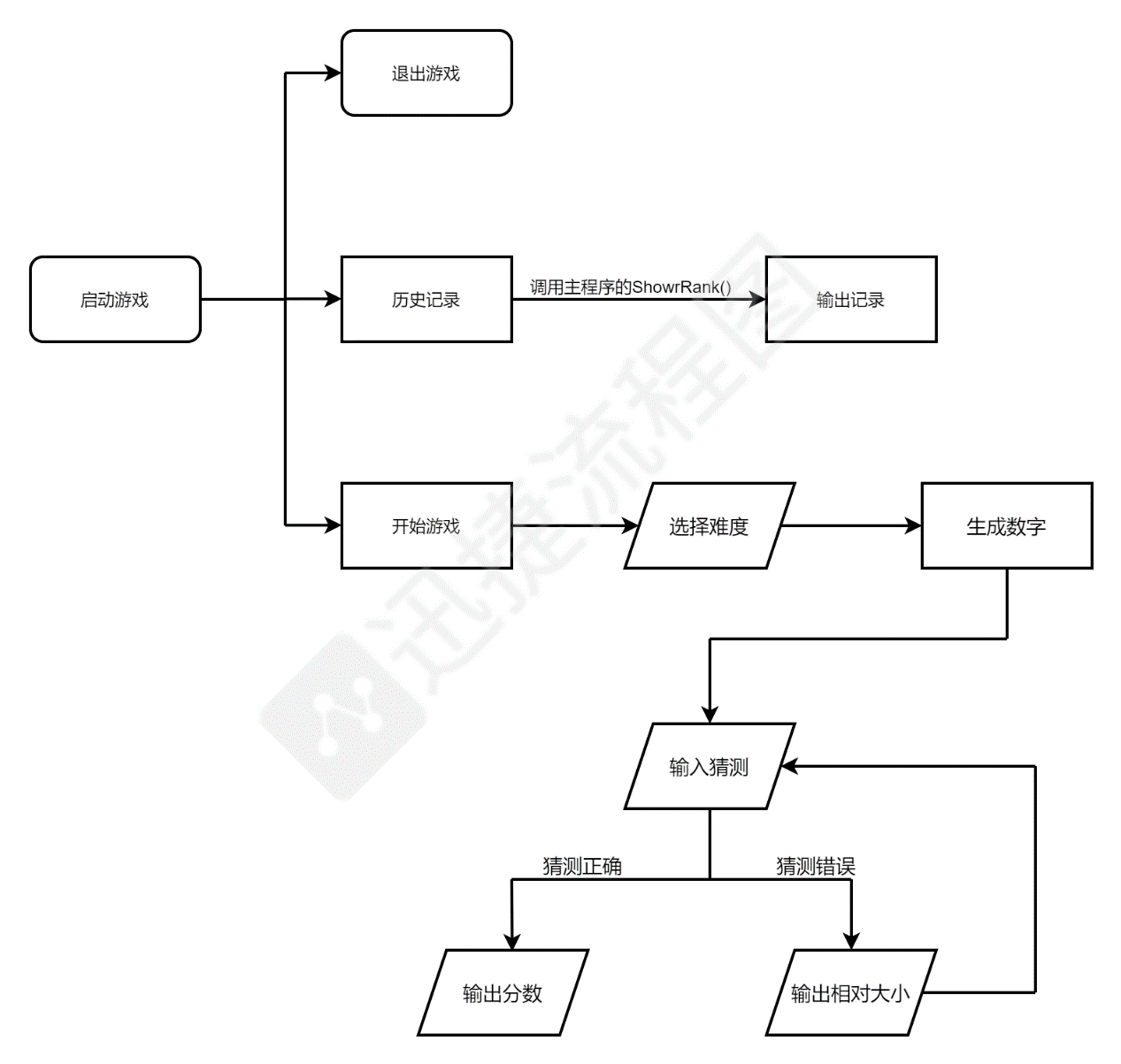
* RSA.c

1. #include "RSA.h"
2. #define int long long
3. static int e = 3049, d = 2361049, p = 194591273; *//公钥、私钥*
4. int ksm(int a, int b){*//快速幂算法，计算a^b mod p*
5. int ret = 1;
6. while(b){
7. if(b & 1) ret = 1LL \* ret \* a % p;
8. a = 1LL \* a \* a % p;
9. b >>= 1;
10. }
11. return ret;
12. }
13. int Encode(int x){return ksm(x, e);}
14. int Decode(int x){return ksm(x, d);}
15. #undef int

* RSA.h

1. #define int long long
2. int ksm(int, int);
3. int Encode(int);
4. int Decode(int);
5. #undef int

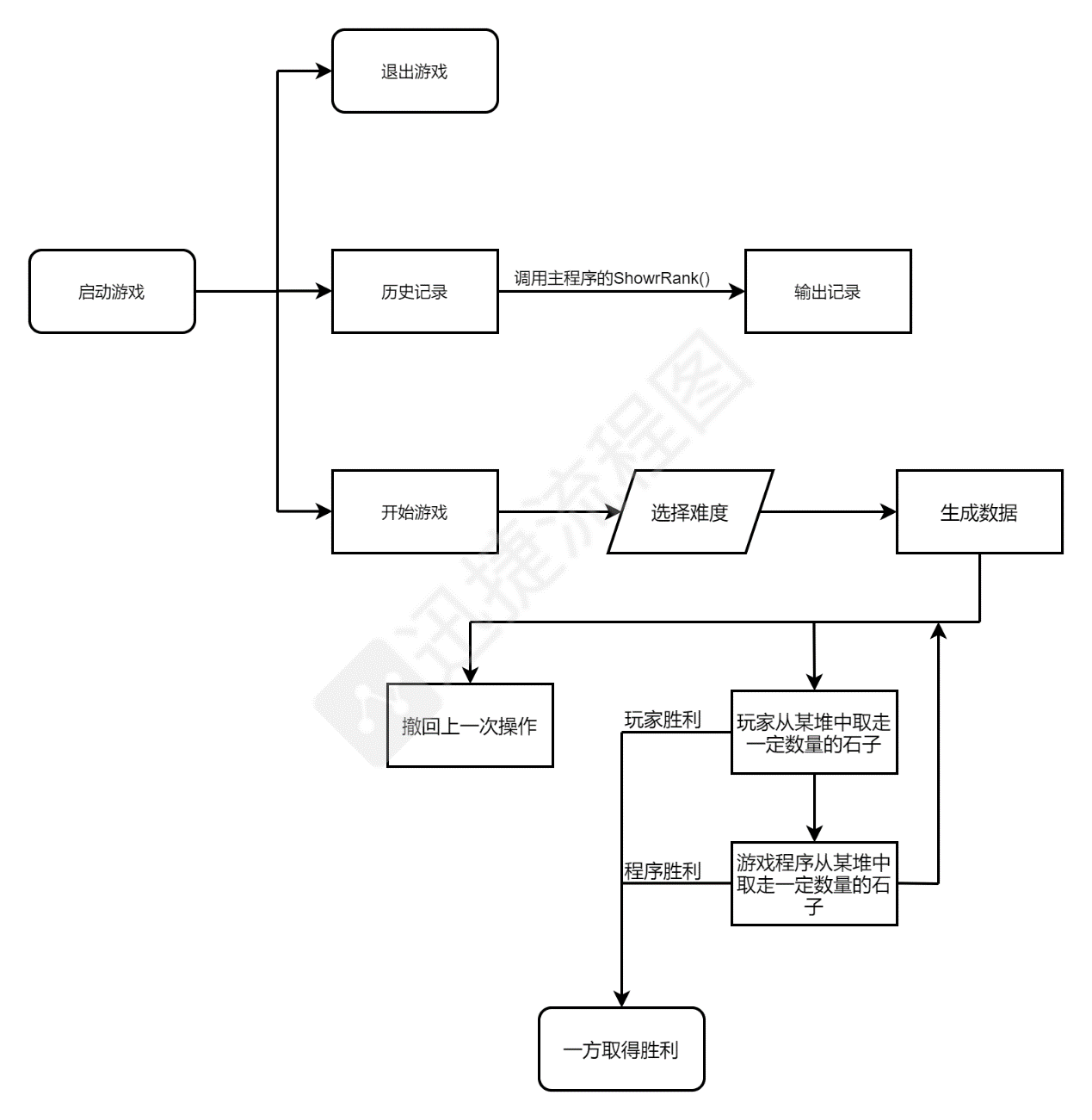
### 4.2.2 猜数游戏



这是游戏合集的第1个游戏，游戏原理比较简单，设计的初衷是为了检验一下分支程序代码能否和自己写的主框架代码进行合并且正常运行，同时给出一个多文件编写程序的例子，便于组员后期交付代码。游戏的原理应用了二分原理，为使得期望猜测次数最少，每一次猜测我们均选取区间的中位数，这样每一次都可以使答案区间缩小一半，在最坏情况下，也只需要次猜测就能得到正确答案，据此程序在第3子关卡也设定了最大猜测次数，超过该次数还未猜对视为游戏失败。

1. #include "Subtask1.h"
2. #include "diynode.h"
3. #include<stdio.h>
4. #include<stdlib.h>
5. #include<time.h>
6. extern node Data[];
7. extern int n, id, taskid;
8. #define MaxPoints Data[id].data[1][1]
9. #define Level Data[id].data[1][2]
10. static int lg[666666+5];
11. static int up[] = {0, 26, 520, 666666};
12. static int pts[] = {0, 1000, 2500, 6666};
13. static void Game(int k){
14. srand((unsigned)time(NULL));
15. int num = 1LL \* rand() \* rand() \* rand() % up[k] + 1, x, cnt = 0, maxcnt = lg[up[k]]+1;*//根据二分原理计算猜数次数上限*
16. clock\_t start = clock(), end; *//int Points = 0;*
17. if(k <= 2) printf("当前为第 %d 个子关卡，生成数的范围是1 - %d，你可以无限制猜数\n", k, up[k]);
18. else printf("当前为第 %d 个(BOSS)子关卡，生成数的范围是1 - %d，你最多只有%d次猜测机会\n", k, up[k], maxcnt);
19. while(1){
20. if(cnt > maxcnt && k >= 2){
21. puts("次数超过限制，你再想想，游戏失败!!!");
22. printf("正确答案是 %d \n", num);
23. system("pause");
24. return;
25. }
26. printf("请输入你的Guess:");
27. fflush(stdin); int ret = scanf("%d", &x); fflush(stdin);
28. ++cnt;
29. if(ret == 0) {
30. puts("不合法操作！");
31. continue;
32. }
33. if(x > num) printf("太大了\n");
34. else if(x < num) printf("太小了\n");
35. else{
36. puts("Congratulations! 你猜对了诶!"); end = clock(); double dur=(double)(end-start)/CLK\_TCK;
37. int this\_pts = pts[k] - Min((int)dur \* 2, 495) - Min(50 \* (cnt-1), 495);
38. Data[id].data[1][2+k] = Max(Data[id].data[1][2+k], this\_pts);
39. printf("当前为第 %d 个子关卡，本次得分 %d  用时：%.3lfs 最高得分 %d\n", k, this\_pts, dur, Data[id].data[1][2+k]);
40. MaxPoints = Max(MaxPoints, Data[id].data[1][3] + Data[id].data[1][4] + Data[id].data[1][5]);
41. Level = Max(Level, k);
42. Data[id].taskid = Data[id].taskid > 1 ? Data[id].taskid : 1; system("pause");
43. return;
44. }
45. }
46. }
47. void Subtask1(){
48. int op = -1; Data[id].data[1][0] = 5;
49. if(lg[2] == 0)
50. for(int i = 2; i <= 666666; ++i) lg[i] = lg[i>>1] + 1; *//递推计算以2为底数的对数值表*
51. while(1){
52. system("cls");
53. puts("欢迎来到猜数小游戏, 本游戏共有3个子关卡, 通过第1个即可进入下一个游戏:");
54. printf("LRL52随机生成一个给定范围的整数，你输入猜数猜数结果，LRL52会反馈给你猜数结果的相对大小\n");
55. printf("1.开始游戏\n2.历史记录\n3.退出游戏\n");
56. printf("请输入你的选择:");  op = -1;
57. fflush(stdin); scanf("%d", &op); fflush(stdin);
58. if(op == 1){
59. printf("你的最高记录是通过了前 %d 个子关卡，故你可以选择1 - %d中的任意子关卡\n", Level, Min(3, Level+1));
60. printf("请输入你的选择:"); op = -1; fflush(stdin); scanf("%d", &op); fflush(stdin);
61. if(op < 1 || op > Min(3, Level+1)){puts("不合法操作！！！"); system("pause");}
62. else Game(op);
63. }
64. else if(op == 2){
65. */\*system("cls"); printf("当前用户: %s\n", Data[id].username);*
66. printf("\t\t\t\t  猜数游戏[关卡1]总分   子关卡1   子关卡2   子关卡3  \n");
67. printf("\t\t\t\t  %18d   %7d   %7d   %7d   \n", MaxPoints, Data[id].data[1][3], Data[id].data[1][4], Data[id].data[1][5]);
68. system("pause");\*/
69. system("cls"); printf("当前关卡：猜数游戏[关卡1]   当前用户: %s\n\n", Data[id].username);
70. printf("\t\t\t\t     Username   名次   总分   子关卡1   子关卡2   子关卡3  \n");
71. ShowRank(1);
72. system("pause");
73. }
74. else if(op == 3) return;
75. else {puts("不合法操作！！！"); system("pause");}
76. }
77. }

### 4.2.3 取石子博弈



在许多地方曾经流行过这样一个小游戏：摆出三枚硬币，分别包含3枚、5枚、7枚。两人轮流行动，每次可以任选一堆，从中取走任意多枚硬币，可把一堆取光，但不能不取。取走最后一枚硬币者获得胜利。我们把这种游戏称为Nim博弈，本游戏idea来自它。

假设我们有堆石子，每堆石子有个

定理：

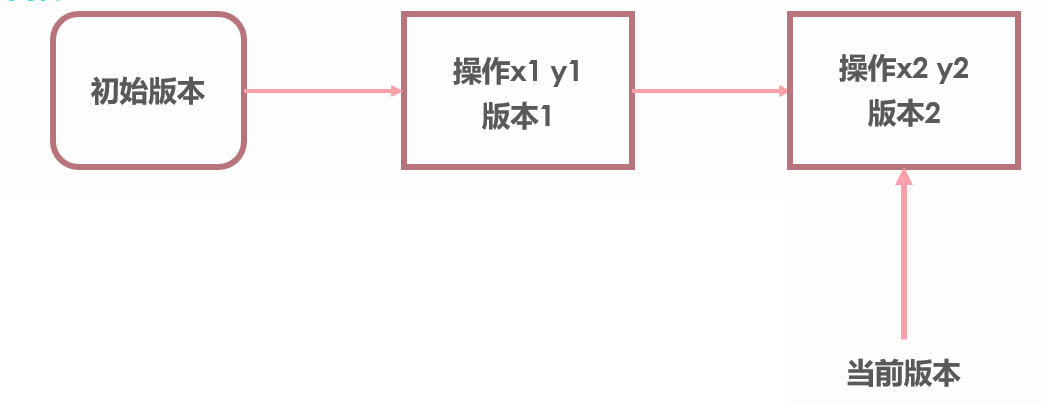
先手能取得胜利当且仅当

证明：

* Case 1:当前没有石子，则异或之和为0，为失败局面
* Case 2:当前有石子，且，则取走能取的任意数量的石子，剩下石子的异或之和必然不为0
* Case 3:当前有石子，且，记

考虑在二进制下的最高位1，则必然存在某堆石子的二进制该位也为1，令，则故取走第堆的石子，取走后

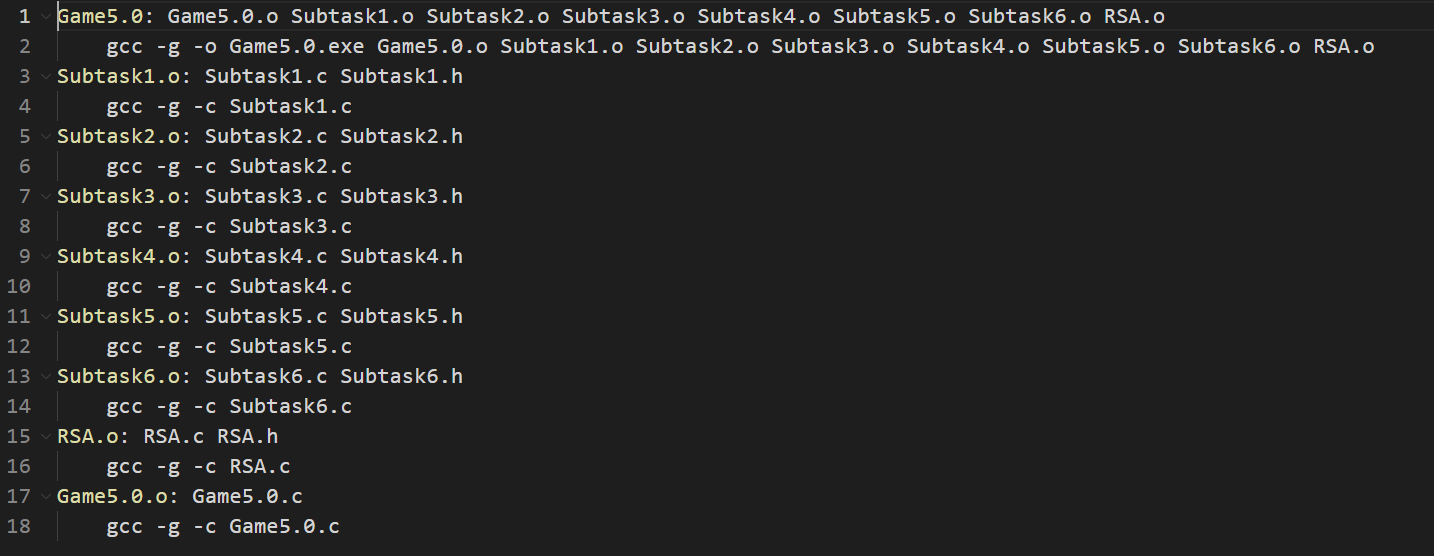
据此理论，我令构造了3个难度的关卡，同时程序可以找出当前局面的最优解，如果为必败局面，则随机操作。程序还使用了单向链表来记录每一次操作，实现操作撤回功能。



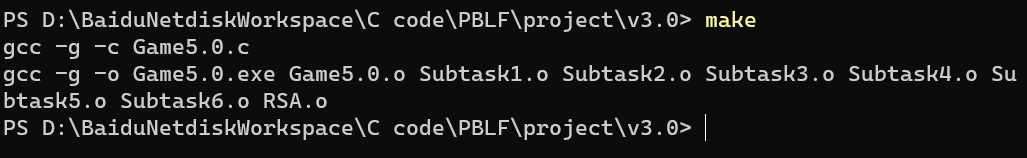
1. #include "Subtask3.h" *//引入你的头文件*
2. #include "diynode.h"  *//引入自定义结构体类型,*
3. #include<stdio.h>
4. #include<stdlib.h>
5. #include<time.h>
6. #include<windows.h>
7. extern node Data[]; *//共享游戏数据Data*
8. extern int id, taskid; *//共享id, taskid, id为当前用户对应的数字序号, 用作Data[]下标, taskid表示当前正在进行或即将进入的关卡, 你很可能会用到*
9. *//你可以在下面写函数，定义变量等等*
10. #define MaxPoints Data[id].data[3][1]
11. #define Level Data[id].data[3][2]
12. static int a[100+5];
13. static int N[] = {0, 3, 5, 7};
14. static int M[] = {0, 3, 25, 125};
15. static int pts[] = {0, 1000, 2500, 6666};
16. typedef struct ListNode{
17. int x, y;
18. struct ListNode \*prev;
19. }ListNode;
20. static ListNode \*head, \*tail;
21. static void print(int n){
22. printf("现有的%d堆石子数量依次为：\n", n);
23. for(int i = 1; i <= n; ++i) printf("%d ", a[i]);
24. putchar('\n'); putchar('\n');
25. }
26. static int checkend(int n){
27. for(int i = 1; i <= n; ++i) if(a[i] > 0) return 0;
28. return 1;
29. }
30. static void Insert(int x, int y){
31. ListNode \*cur = (ListNode\*)malloc(sizeof(ListNode));
32. cur->x = x, cur->y = y;
33. cur->prev = tail, tail = cur;
34. }
35. static void Delete(){
36. if(tail == head){
37. printf("不合法的操作！！！\n\n");
38. return;
39. }
40. static ListNode \*p;
41. a[tail->x] += tail->y, p = tail, tail = tail->prev, free(p);
42. a[tail->x] += tail->y, p = tail, tail = tail->prev, free(p);
43. puts("撤回成功"); Sleep(2000);
44. }
45. static void NimGame(int k){
46. printf("\n-------------------初始化游戏---------------------\n");
47. Sleep(1000); head = (ListNode\*)malloc(sizeof(ListNode)), tail = head;
48. if(k <= 2) printf("当前为第 %d 个子关卡: 共有 %d 堆石子，每堆石子的个数不超过 %d 个\n", k, N[k], M[k]);
49. else printf("当前为第 %d 个(BOSS)子关卡: 共有 %d 堆石子，每堆石子的个数不超过 %d 个，且你仅有3次撤回机会\n", k, N[k], M[k]);
50. printf("请每次输入用空格分隔的2个数x y\n表示将第x堆石子拿走y个\n输入-1表示撤回上一次你和LRL52的操作，输入-2表示退出本局游戏\n\n");
51. srand((unsigned)time(NULL)); int x = 0, y = 0, n = N[k], m = M[k], cnt = 0;
52. for(int i = 1; i <= n - 1; ++i) a[i] = rand() \* rand() % m + 1, x ^= a[i];
53. if(k == 1){
54. for(int i = 1; i <= m; ++i) if(x ^ i) {x ^= i; break;}
55. a[n] = x;
56. } else{
57. if(x > 0 && rand() % 5) a[n] = x;
58. else a[n] = rand() \* rand() % m + 1;
59. }
60. clock\_t start = clock(), end;
61. while(1){
62. print(n);
63. if(checkend(n)){
64. puts("You lose the game! ! !");
65. system("pause"); return;
66. }
67. while(1){
68. printf("请输入你的操作x y:");
69. fflush(stdin); x = 0; scanf("%d", &x);
70. if(x == -1){
71. if(k == 3 && cnt == 3){
72. puts("撤回次数已达上限，操作无效！");
73. continue;
74. }
75. Delete(); print(n);  fflush(stdin); ++cnt;
76. continue;
77. }
78. if(x == -2){fflush(stdin); return;}
79. scanf("%d", &y); fflush(stdin);
80. if(x > n || x < 1 || y <= 0 || a[x] < y){
81. puts("不合法的操作！！！");
82. continue;
83. }
84. a[x] -= y;
85. Insert(x, y);
86. break;
87. }
88. print(n);
89. if(checkend(n)){
90. puts("Congratulations, You win!");
91. end = clock(); double dur=(double)(end-start)/CLK\_TCK;
92. int this\_pts = pts[k] - Min((int)dur \* 2, 495) - Min(50 \* cnt, 495);
93. Data[id].data[3][2+k] = Max(Data[id].data[3][2+k], this\_pts);
94. printf("当前为第 %d 个子关卡，本次得分 %d  用时：%.3lfs 撤回次数：%d  最高得分 %d\n", k, this\_pts, dur, cnt, Data[id].data[3][2+k]);
95. MaxPoints = Max(MaxPoints, Data[id].data[3][3] + Data[id].data[3][4] + Data[id].data[3][5]);
96. Level = Max(Level, k);
97. Data[id].taskid = Data[id].taskid > 3 ? Data[id].taskid : 3; system("pause");
98. return;
99. }
100. Sleep(2000); x = 0;
101. for(int i = 1; i <= n; ++i) x ^= a[i];
102. if(x){
103. for(int i = 1; i <= n; ++i)
104. if((a[i] ^ x) < a[i]){
105. y = a[i] - (a[i]^x);
106. printf("LRL52决定将第%d堆石子拿走%d颗\n", i, y);
107. a[i] -= y; Insert(i, y);
108. break;
109. }
110. } else{
111. for(int i = 1; i <= n; ++i)
112. if(a[i] > 0){
113. y = rand() \* rand() % a[i] + 1;
114. printf("LRL52决定将第%d堆石子拿走%d颗\n", i, y);
115. a[i] -= y; Insert(i, y);
116. break;
117. }
118. }
119. }
120. }
121. void Subtask3(){ *//下面将是你的游戏入口, 确保return后将taskid更改为合适的值, 1~Gamecnt表示对应关卡, -1表示退出游戏*
122. Data[id].data[3][0] = 5;
123. int op = -1; Sleep(1000);
124. while(1){
125. system("cls");
126. printf("╭===================○●○●==================╮\n");
127. printf("│  ★ ☆欢迎来到Subtask3: LRL52的博弈☆ ★    │\n");
128. printf("╰=======○●○●==============================╯\n");
129. printf("游戏规则：你有若干堆给定数量的石子\n");
130. printf("你和LRL52每次可从任意一堆石子里取出任意多枚石子扔掉，可以取完，不能不取，你拥有尊贵的先手资格\n");
131. printf("每次只能从一堆里取。最后没石子可取的一方就输了\n");
132. printf("1.开始游戏\n2.历史记录\n3.退出游戏\n");
133. printf("请输入你的选择:");
134. fflush(stdin); scanf("%d", &op); fflush(stdin);
135. if(op == 1){
136. printf("你的最高记录是通过了前 %d 个子关卡，故你可以选择1 - %d中的任意子关卡\n", Level, Min(3, Level+1));
137. printf("请输入你的选择:"); op = -1; scanf("%d", &op); fflush(stdin);
138. if(op < 1 || op > Min(3, Level+1)) {puts("不合法操作！！！"); system("pause");}
139. else NimGame(op);
140. }
141. else if(op == 2){
142. system("cls"); printf("当前关卡：LRL52的博弈[关卡2]   当前用户: %s\n\n", Data[id].username);
143. printf("\t\t\t\t     Username   名次   总分   子关卡1   子关卡2   子关卡3  \n");
144. ShowRank(3);
145. system("pause");
146. }
147. else if(op == 3) return;
148. else { puts("不合法操作！！！"); system("pause"); }
149. }
150. }

# 5 测试

Makefile文件：



终端输入make进行编译：



运行



登录

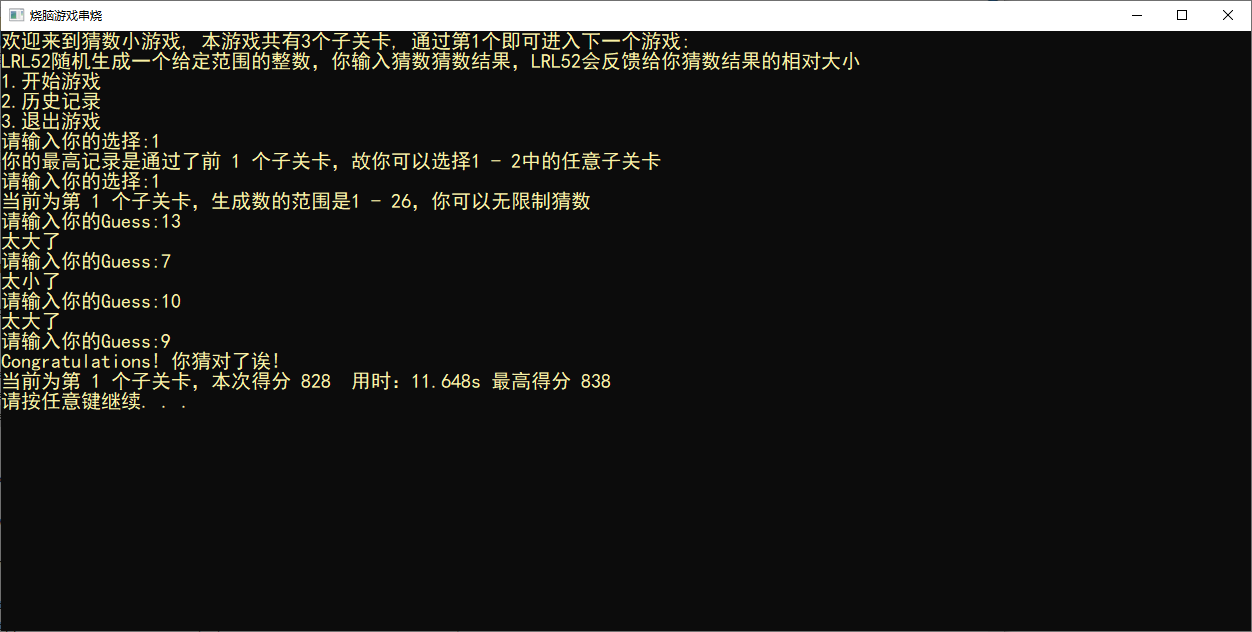




排行榜查看：



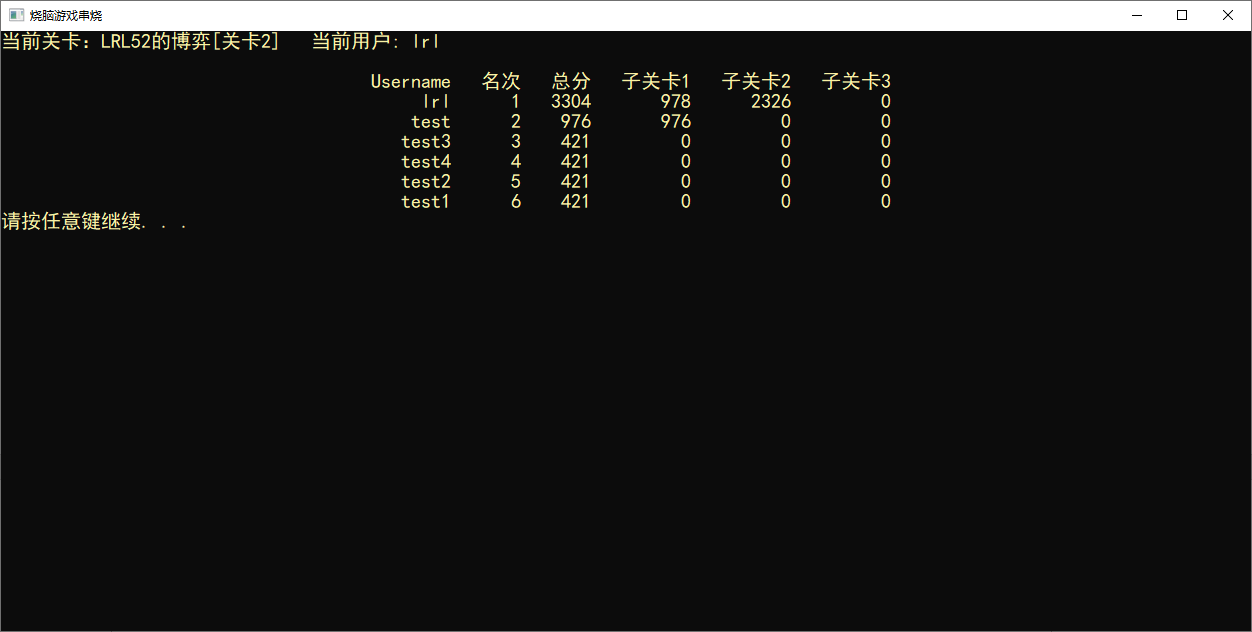
猜数游戏：



取石子博弈：



单个关卡排名：



# 6 总结

**刘芮利-2021090906016**

* 学到了什么？
* 学习了使用多文件编写程序的基本方法，了解了Makefile的基本编写格式，学会了几条简单的GCC编译命令
* 学会了如何使用文件进行数据读写，明白了foopen不同模式的区别
* 复习了快速排序算法和快速幂算法，将单向链表和RSA加密应用实践
* 学会了部分windows库函数的基本使用，找到了程序中文乱码的解决方法
* 痛点和难点？
* 多文件程序编写的函数和数据共享问题。起初我将自定义结构体node的声明和定义放在主程序中，在分支程序中使用extern关键字，但编译会提示没有过定义该结构体。后来又把声明和定义放在了Subtask1.h中，但是Subtask2.c又不会包含Subtask1.h，Subtask1.h并不是所又不是分支程序共同包含的头文件。最后干脆再定义一个头文件diynode.h，用于给出自定义结构体的声明，所有源代码包含该头文件，并将结构体的定义放在主程序中，其它程序使用extern关键字共享该结构体变量。
* 不同程序函数名、变量名冲突问题。起初本打算使用namespace来划分各个程序避免命名冲突，发现namespace是C++的语法，考虑到该项目是C程序设计实践，遂放弃改为尝试多文件编写。但多文件编写过程中我意外发现并没能解决变量名和函数冲突的问题，我在Subtask1.c中定义了Game()函数，在Subtask3.c就没法再次定义了，并且我还意外的发现组员程序定义的变量n在合并后就和我主程序定义的变量n是同一个变量了。经过查阅资料发现，可以使用static关键字，当它作用于全局变量时，可以让变量具有内部链接和文件作用域，对函数也有类似的效果。这样我把所有不共享的函数和全局变量全部加上static关键字，就成功解决了命名冲突问题。
* 自己的贡献
* 编写了主程序，加密模块，猜拳游戏，取石子博弈
* 统筹、设计和监督项目，后期代码整合，让组员编写的代码和主程序对接，实现数据的文件存储
* 协同组员合作完成答辩PPT制作和项目报告公共部分的编写
* 如何与他人合作？



* 上图是项目初期编写的合作指南，旨为组员的编写程序提供方向，便于后期代码整合。当然实践过程中我和组员都会遇到超出预料的困难和挑战，有人遇到问题大家一起沟通，共同商讨，出谋划策，是解决问题的不二法门。此外，我们秉承“他山之石，可以攻玉”的共同进步的理念，每当自己对某个问题有独到见解和解决办法的时候，我们都会及时分享心得。
* 总结与展望
* 整个项目全组成员勠力同心，按时完成了项目交付。虽然我们组每个成员相比其它组代码量更大，任务更重，但一分耕耘，一分收获，相信我们组每个成员的收获也更多。但仍有很多不足亟待改进，例如我的加密模块，因为这学期误打误撞步入了网安Crypto的学习，所以对各种加密算法有一定了解，我深知由于C语言不支持大整数限制了密钥长度，现在的加密只是纸老虎。我原本希望能实现基于快速傅里叶变换（FFT）的RSA加密算法（通过FFT实现大整数乘法），同时将密钥放在服务器上，通过socket套接字编程来实现密钥分配。不过后期囿于个人能力有限，期末考试临近，且还需要大量调试代码、测试组员代码，没有能够完成。道阻且长，行则将至，期望终有一日待个人能力的羽翼成熟之时，让程序充满更多的可能。