

**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 |
| 1（组长） | 2021090906016 | 刘芮利 |
| 2 | 2021090906020 | 张庭恺 |
| 3 | 2021090906002 | 赵晗 |
| 4 | 2021090906010 | 钟城伟 |
| 5 | 2021090906008 | 黄培坤 |

目录

[1 项目简介 3](#_Toc91942182)

[1.1 考核方式 3](#_Toc91942183)

[1.2 项目题目及内容简介 3](#_Toc91942184)

[1.3 项目组成员与分工 4](#_Toc91942185)

[2 需求分析 4](#_Toc91942186)

[2.1 选题的依据 4](#_Toc91942187)

[2.2 功能需求 5](#_Toc91942188)

[3 系统设计 6](#_Toc91942189)

[3.1 总体设计 6](#_Toc91942190)

[3.2 模块设计 7](#_Toc91942191)

[4 系统实现 8](#_Toc91942192)

[4.1 主函数 8](#_Toc91942193)

[4.1.1 所有函数及其变量声明 8](#_Toc91942194)

[4.1.2 读取数据 10](#_Toc91942195)

[4.1.3 写入数据 11](#_Toc91942196)

[4.1.4 注册新用户 12](#_Toc91942197)

[4.1.5 登录 13](#_Toc91942198)

[4.1.6读取密码 14](#_Toc91942199)

[4.1.7 输出Logo 14](#_Toc91942200)

[4.1.8 颜色控制 15](#_Toc91942201)

[4.1.9 排名系统 15](#_Toc91942202)

[4.1.10 主菜单 16](#_Toc91942203)

[4.1.11 主函数 17](#_Toc91942204)

[4.2 其他函数 18](#_Toc91942205)

[5 测试 18](#_Toc91942206)

[6 总结 19](#_Toc91942207)

# 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

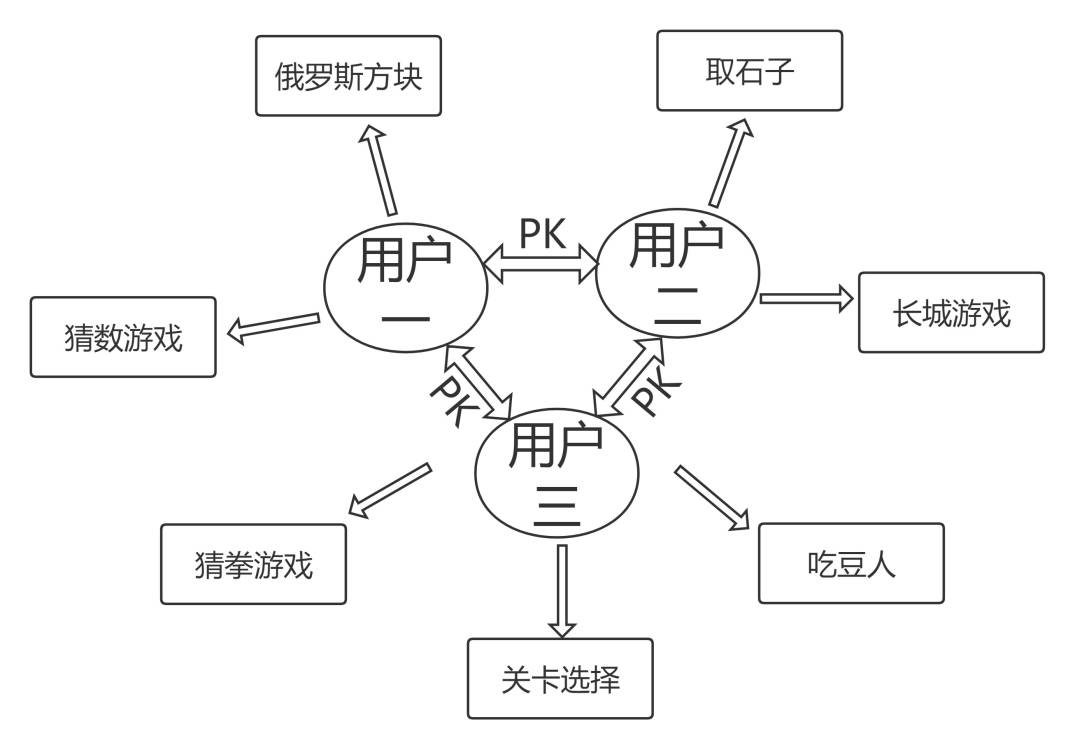
钟城伟：长城游戏编写（包括1.最短步数的计算，等价于最佳匹配，使用了KM算法，大大缩短了时间复杂度，由缩减为。2.作了获胜或失败后的美化。3.追踪生成答案方阵。4.利用字符数组读入用户输入，再进行判断，增强程序的排错能力，避免出现bug。5.使用时间函数，以达到计时目的。6.巧妙使用多个goto语句，以达成选择性循环。）

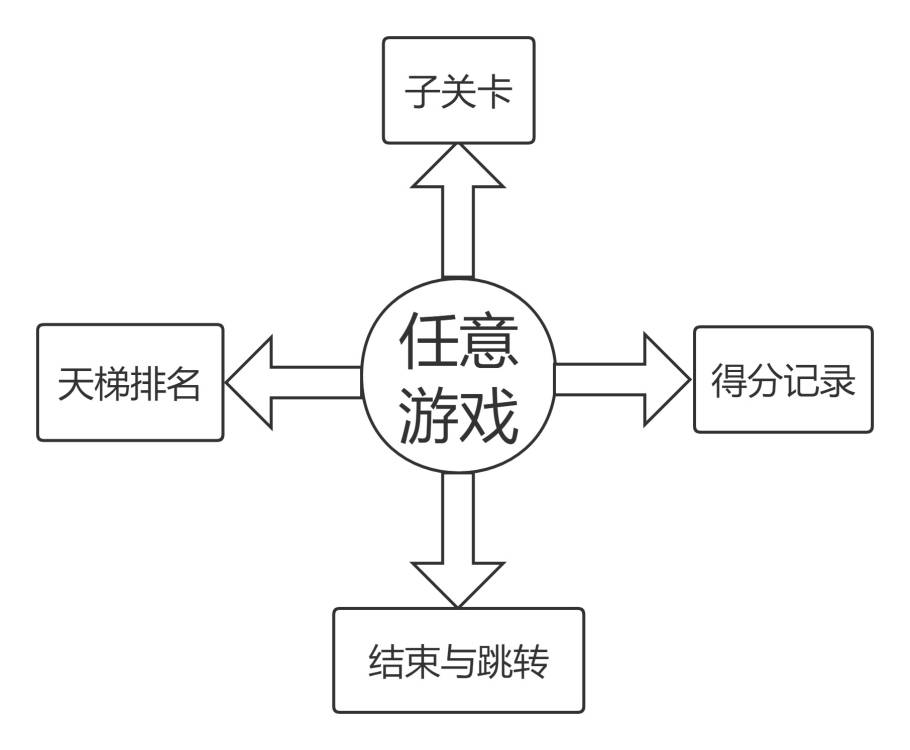
# 2 需求分析

## 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

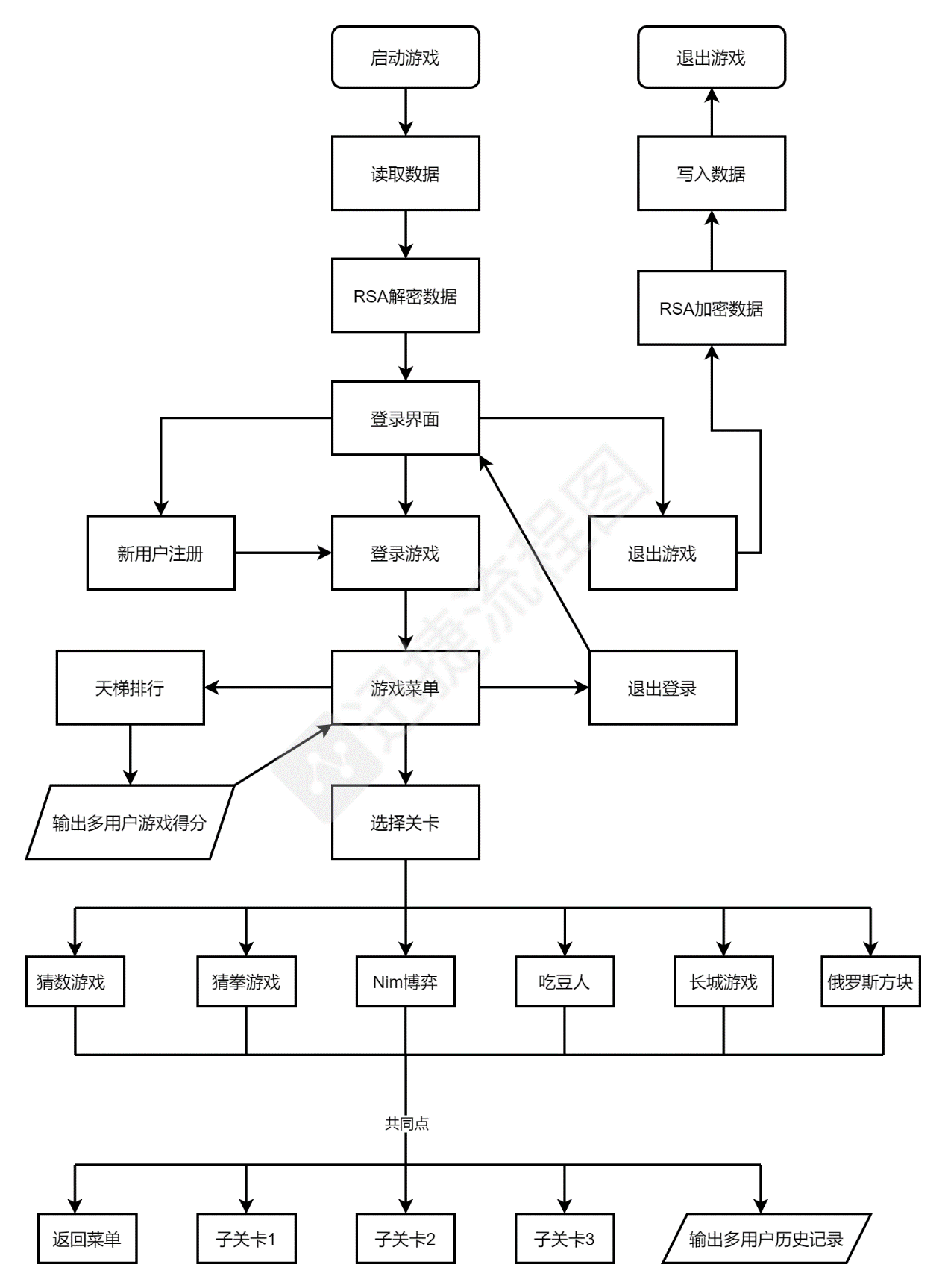
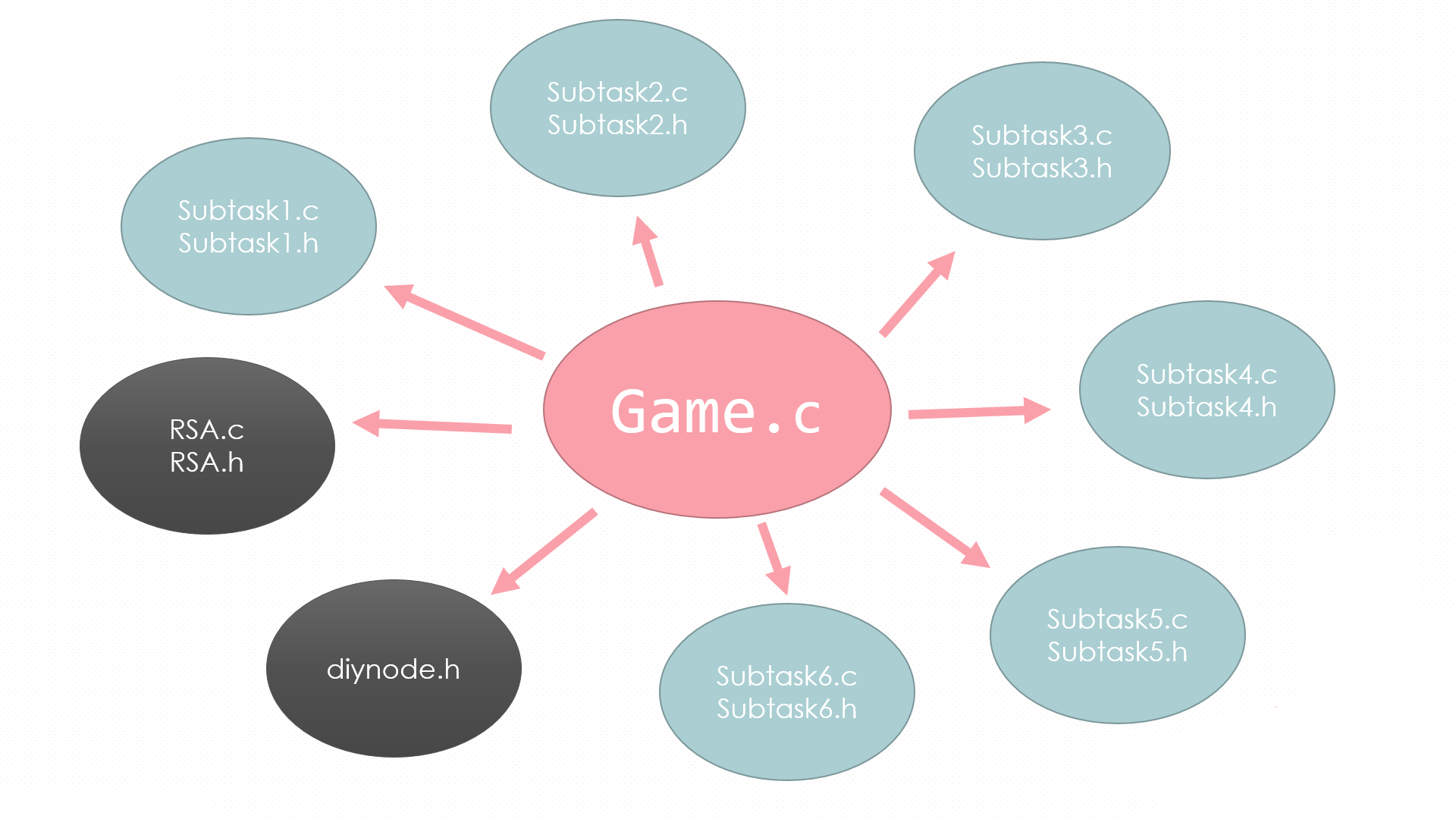


图3-1

## 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**

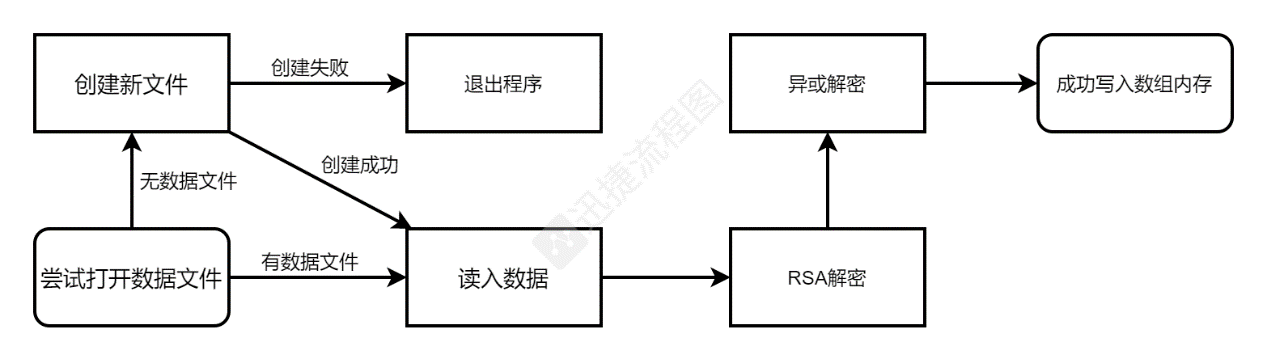
# 4 系统实现

## 4.1 主程序（游戏框架）

### 4.1.1 所有函数及其变量声明

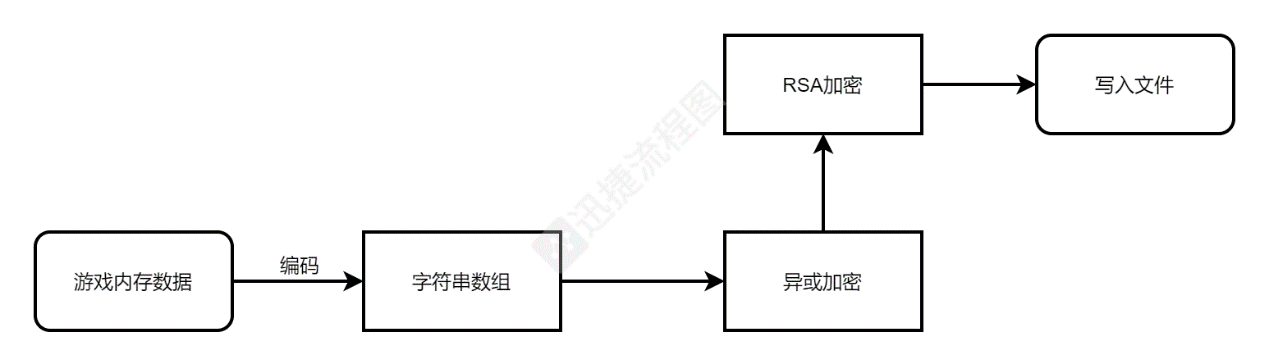
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. #include <time.h>
5. #include <windows.h>
6. #include <conio.h>
7. #include "diynode.h"
8. #include "Subtask1.h"
9. #include "Subtask2.h"
10. #include "Subtask3.h"
11. #include "Subtask4.h"
12. #include "Subtask5.h"
13. #include "Subtask6.h"
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



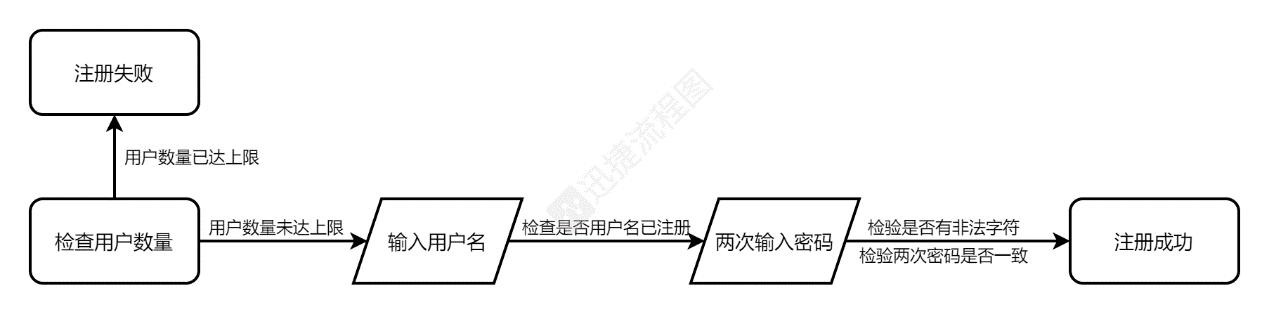
1. static void ReadData(){
2. io = fopen("./data.txt", "r");
3. if(io == NULL){
4. *//puts("Unexpected Error ??? \n(If first run, please try again!)");*
5. io = fopen("./data.txt", "w"); fclose(io); *//尝试创建新文件*
6. io = fopen("./data.txt", "r");
7. if(io == NULL){
8. puts("Unexpected Error ??? \n(If first 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



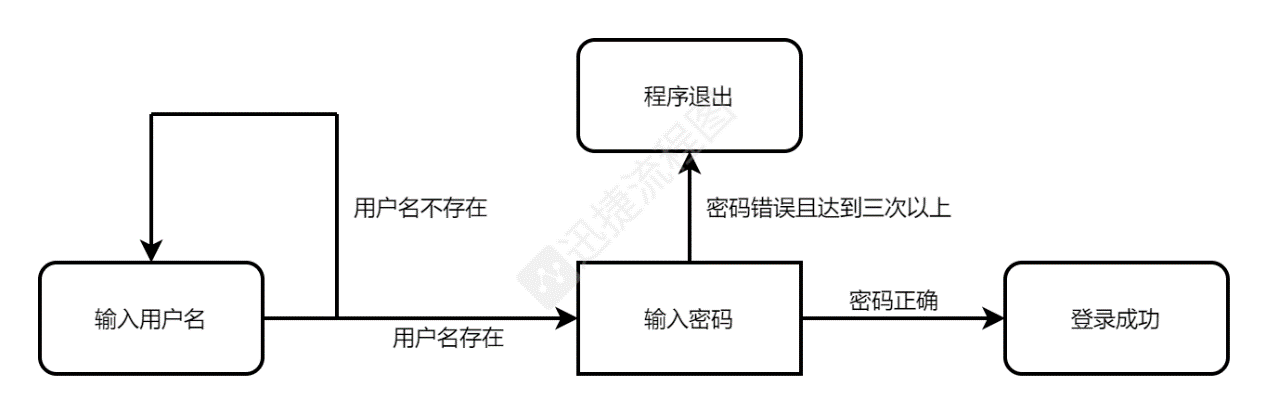
1. static void WriteData(){ *//保存，写入数据*
2. memset(wr, 0, sizeof(wr)); char \*p = wr;
3. p += sprintf(p, "%d\n", n);
4. for(int k = 1; k <= n; ++k){
5. p += sprintf(p, "%s %s %d\n", Data[k].username, Data[k].password, Data[k].taskid);
6. for(int i = 1; i <= GameCnt; ++i){
7. p += sprintf(p, "%d", Data[k].data[i][0]);
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. }

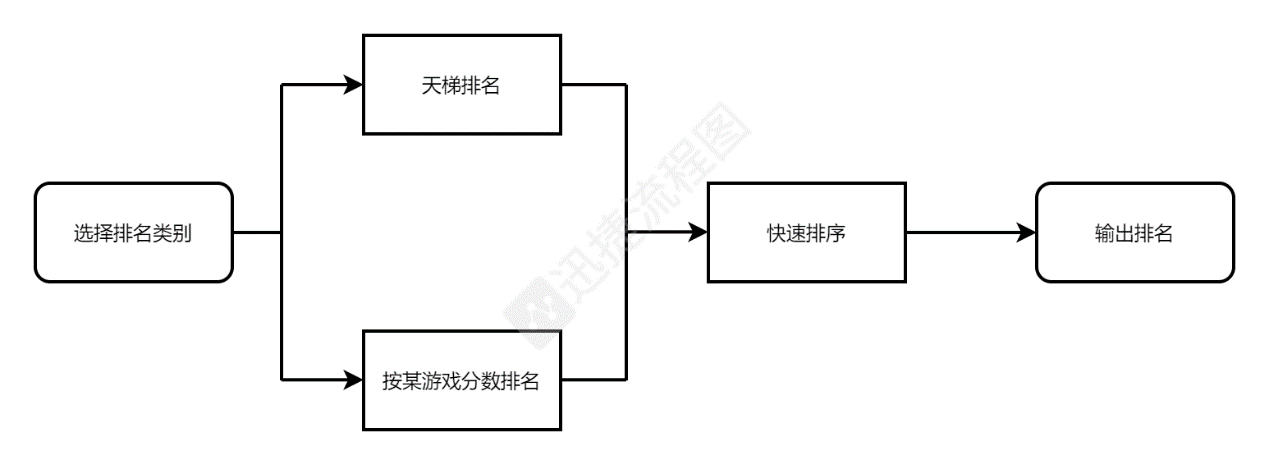
### 4.1.7 输出Logo

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

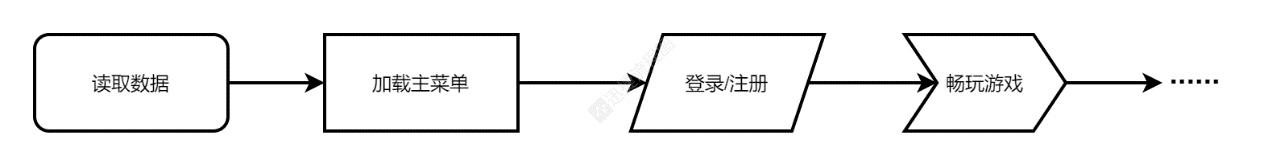


1. static inline int calc(int i, int op){
2. if(op == 0){
3. int ret = 0;
4. for(int j = 1; j <= GameCnt; ++j) ret += Data[i].data[j][1];
5. return ret;
6. }
7. else return Data[i].data[op][1];
8. }
9. static void quicksort(int l,int r, int op)
10. {
11. int mid = (l + r) >> 1;*//中间数*
12. int i = l, 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. }

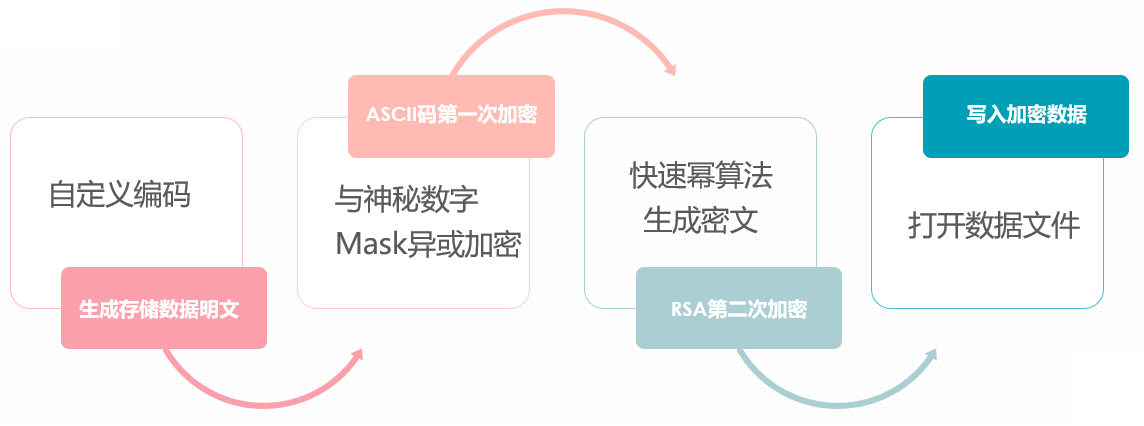
### 4.1.11 主函数



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

# 5 测试

# 6 总结

**姓名-学号：你的总结**

学到了什么？

痛点和难点

自己的贡献

如何与他人合作？