

**重 庆 理 工 大 学**

**课程设计**

**课程 程序设计基础I**

**题目 C语言综合程序设计**

院系名称 计算机科学与工程学院

班 级 计算机科学与技术二班

学生姓名 张晓君 学号 11503070234

指导教师 曹琼

评阅教师

时 间 2016年1月11日

目录

[目录 2](#_Toc440323265)

[①.11-24点游戏 3](#_Toc440323266)

[1 问题描述 3](#_Toc440323267)

[1.1 问题描述 3](#_Toc440323268)

[1.2 游戏描述 3](#_Toc440323269)

[2 功能分析 3](#_Toc440323270)

[2.1 帮我计算 3](#_Toc440323271)

[2.2 开始练习 3](#_Toc440323272)

[2.3 开始挑战 4](#_Toc440323273)

[2.4 双人模式 4](#_Toc440323274)

[2.5 设置 4](#_Toc440323275)

[3 算法分析 5](#_Toc440323276)

[3.1 穷举法 5](#_Toc440323277)

[3.2 制定重解消除规则 5](#_Toc440323278)

[3.3 模拟二叉树 5](#_Toc440323279)

[3.4 循环实现以及栈的恢复 6](#_Toc440323280)

[4 函数详解 6](#_Toc440323281)

[②.4-高级语言源程序注释部分的处理 8](#_Toc440323282)

[1问题描述 8](#_Toc440323283)

[1.1 程序描述 8](#_Toc440323284)

[1.2 程序描述 8](#_Toc440323285)

[2 函数详解 8](#_Toc440323286)

[③.18-单项选择题标准化考试系统 9](#_Toc440323287)

[1 问题描述 9](#_Toc440323288)

[2 程序流程（框图结构） 10](#_Toc440323289)

[3 函数详解 11](#_Toc440323290)

①.11-24点游戏

# 1 问题描述

## 1.1 问题描述

任意给出4张牌，计算能否用＋、－、×、÷将其组合成24。

输出其可能的组合式。

## 1.2 游戏描述

棋牌类益智游戏，要求结果等于二十四，一起来玩玩吧!这个游戏用扑克牌更容易来开展。拿一副牌，抽去大小王后(初练也可以把J/Q/K也拿去)，剩下1～10这40张牌(以下用1代替A)。任意抽取4张牌(称为牌组)，用加、减、乘、除(可加括号)把牌面上的数算成24。每张牌必须且只能用一次。如抽出的牌是3、8、8、9，那么算式为(9-8)×8×3=24。

它始于何年何月已无从考究，但它以自己独具的数学魅力和丰富的内涵正逐渐被越来越多的人们所接受。这种游戏方式简单易学，能健脑益智，是一项极为有益的活动。

# 2 功能分析

## 2.1 帮我计算

当我们用扑克牌玩24点时，抽出四张扑克牌，自己死活也想不出来，难道这四张扑克牌真的算不出来24点？

“帮我计算”是24点程序的核心内容。用户输入4张扑克牌，由计算机帮助用户得到所有解并输出到屏幕上。用户不禁感叹，自己万万没有想到，原来可以这样算出24点。

拓展：1、支持输入JQKA、jqka 甚至是小数。

2、支持24点的修改，用户可以改成36点、10点等等。

算法详见 算法分析 部分

## 2.2 开始练习

当你想锻炼一下自己算24点的能力时，那么请开始练习！

系统随机给出4张扑克牌，当然，这四张扑克牌是可以算出24点的。然后，亲们请输入算式比如 (13+1)\*2-4 然后按回车 程序内置的计算器（感谢Syncing同学的帮助，原谅我 写了三遍都写不对的递归+堆栈计算器）可以计算出结果并且判断是不是等于24，如果不是24的话，程序会告诉你正确的解法的。通过不断的练习，你的计算24点的水平会越来越高的。

## 2.3 开始挑战

当你的计算水平越来越高，练习模式是远远不能满足你的需求的，那么，挑战（天梯、闯关）模式才是你的归属。一次出错，立刻出局。准备好了吗？挑战一下你自己吧，看看自己能闯多少关。

## 2.4 双人模式

独乐乐不如众乐乐，两个人比比谁算的快也是不错的。输入用户A和用户B的姓名后，开始PK！A先算出来按空格，B先算出来按回车，如果两个人都算不出来，可以按其他键跳过。A按了空格之后，要输入有效的算式才会加分。如果答案错误，那么A是会扣分哦。结束PK后，会有简单的判定判断谁获胜或是平局。

## 2.5 设置

输入相应的数字，可以修改如下设置：

1.当前下限(最小的扑克牌):默认A

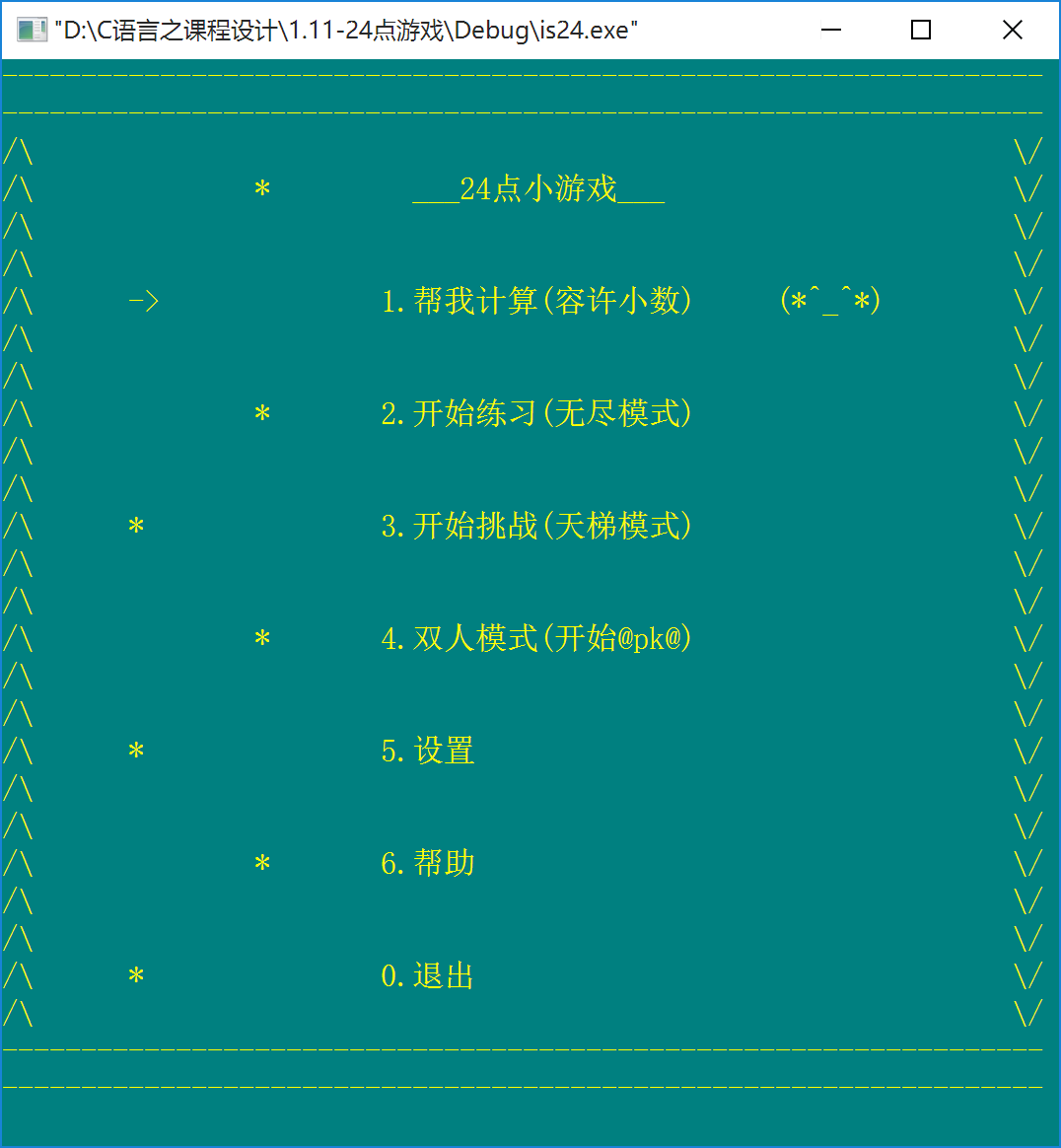
2.当前上限(最大的扑克牌):默认K

3.is 24点(该项设置不会在下次游戏开启时载入...)默认 24

8.重置所有设置(程序有bug时也可以重置)

9.下次运行游戏是否载入当前设置:默认 是

0.返回菜单



# 3 算法分析

## 3.1 穷举法

“给定 4 个数 a,b,c,d ,如何用加减乘除以及括号将其组合起来,得到结果24,是求解24点游戏的算法。由于 a,b,c,d 可以任意交换位置,故有 A44 =24种不同的排列。对于a ,b,c,d 的某个排列 A B CD ,中间可以安排 运算( 表示+、-、×、 /中的任何一个),即:A B C D 。如果选取所有不同的运算符组合, 3 个位置的运算符组合有

43 =64种排列。再考虑到运算的先后顺序,又有 5 种排列。它们分别是:(A B) (C D)、 ((A B) C) D 、(A (B C)) D 、A ((B C) D)、A (B (C D))。总共有 24×64×5=7 680 种不同的算法。此法穷尽了所有可能的组合,它不会遗漏任何一个可能的解。如果有 4 个数,按此法找不到结果为 24 的解,那就是 无解.”

——《彭复明　24 点游戏无重解算法的研究与探讨》

## 3.2 制定重解消除规则

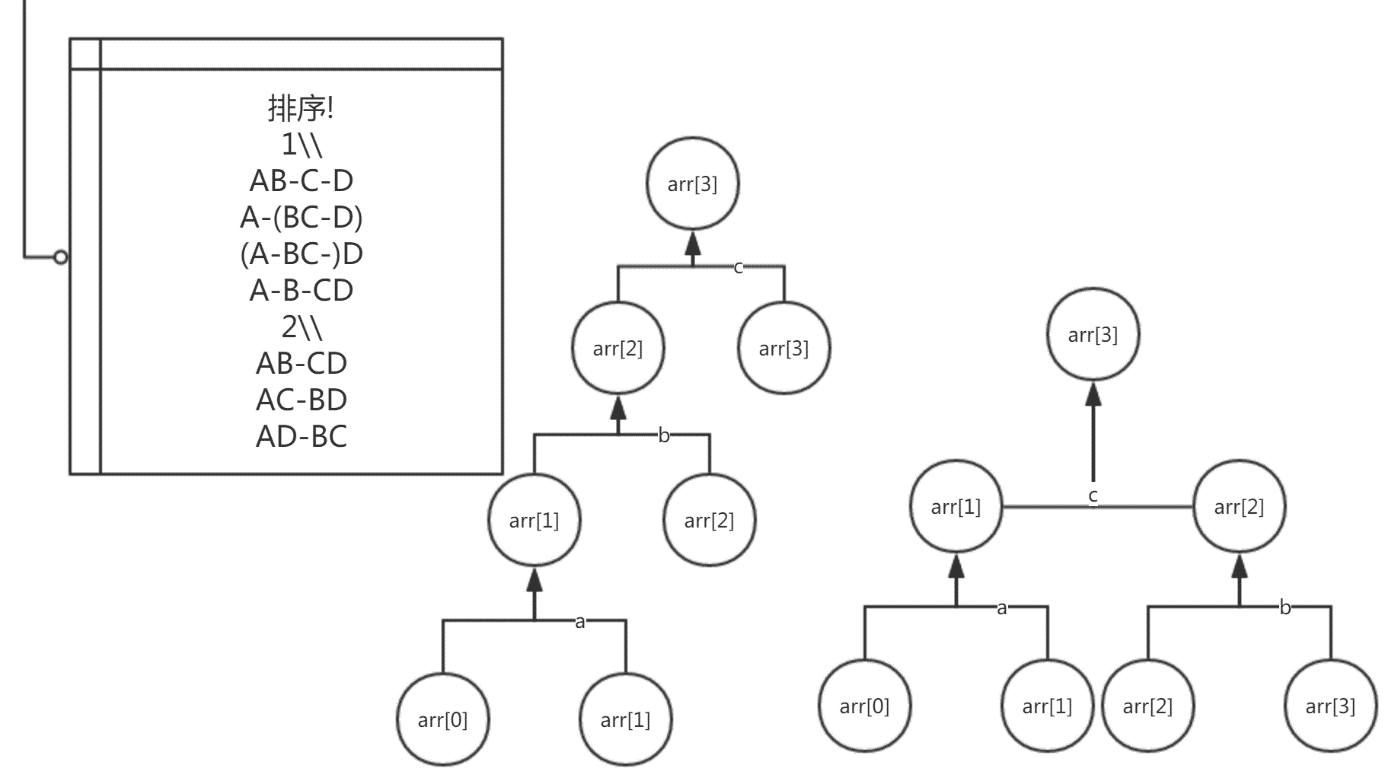
1.满足加法交换律的只输出1组解

2.尽量少的使用括号，最好不使用括号

3.如果有括号，那么括号必须放在最前

经过以上规则，可以消除大部分重复解。

## 3.3 模拟二叉树



## 3.4 循环实现以及栈的恢复

# 4 函数详解

void main(void);//程序入口

void menuPrint();//显示菜单并根据settings.mode的值显示光标

int move();//根据键盘的输入,显示菜单和光标,如果点击回车或数字则进入相应下一级函数

void exitGame(void);//退出程序

void userGet();//与userGetIn()配合使用,直到用户正确输入四张牌才退出死循环

void exercise();//练习模式核心函数

void tianti();//天梯模式核心函数

void pk();//pk模式核心函数

void print\_settings();//设置界面核心函数

void help();//帮助界面核心函数

void change\_settings(int num);//设置界面二级函数,传入要修改设置的内容序号,修改相应的设置

void save\_settings();//保存设置到本地文件

void initSettings\_temp();//初始化菜单光标 指向 1.帮我计算

void resetting();//重置所有设置

void read\_settings();//读取上次保存的设置.第一次打开游戏时显示帮助界面

char \* number\_to\_poker(float number);//把读入的数字变成扑克(字符数组)

float poker\_to\_number(char \*poker);//把扑克(字符数组)变成浮点数

void initAllFromCon();//恢复堆栈数组(从原始数组中)

void initArrFromCur();//恢复临时数组

float calc(float n1, float n2, char o);//传入数字n1.n2.和操作符(+-\*/),返回一个浮点数

void randomGet();//随机得到四个能计算出24点的扑克牌

void printResult\_1(int a,int b,int c);//传来三个操作符 读取数组cur[4]的四个数据 打印出来

void printResult\_2(int a,int b,int c);//等同于printResult\_1

int s\_first(int isPrint);//模拟平衡二叉树之单挂 传入参数为1时输出正解 0则不输出

int s\_second(int isPrint);//模拟平衡二叉树之双链 这两个函数是核心算法 返回1时则是可以算出 返回0就是算不出24点

int userGetIn();//帮我计算核心函数

char \* keyControl\_to\_charPointer(int key);//逗比函数 未完成的函数 多半是废了

char key\_to\_char(int key);//读取键盘输入 返回字符型 多半也是废了

int test(void);//测试函数

void translate(char str[],char exp[]);//对堆栈的细节处理

float cal\_value(char exp[]);//一个使用堆栈的计算器 支持() + - \* /

int zhabi();//调用translate和cal\_value函数

void goto\_pos(int x, int y);//光标移动函数

②.4-高级语言源程序注释部分的处理

# 1问题描述

## 1.1 程序描述

【任务描述】

将C语言程序中的所有注释都去掉，并将去掉注释的文件和注释分别放入一个新的文件中。

【功能描述】

（1）读取用户指定名字的源程序，例如，用户输入：exercise.cpp，程序能读取该文件进行处理。

（2）将文件中的注释（同一行中//之后的部分，以及/\*和\*/之间的部分，包括//、/\*和\*/）部分删除。

（3）将去掉的注释部分和删除注释后的C语言程序，分别保存到两个不同的文件中，文件名允许用户指定。

## 1.2 程序描述

程序在实现上述功能后，还可以直接在框框里面输入代码，在win10下可以直接Ctrl+V粘贴，但是在win7下则不可以，这是由于win32控制台的限制。同是，不能以各种键（如Esc）作为结束标志，也不能以回车作为结束标志。所以，我选择了@符号作为结束标志。如果代码中有@，推荐直接读取xxx.cpp就好。

# 2 函数详解

int first(void);//读取本地文件进行代码与注释分离

int second(void);//在框框里面输入代码分离注释

void menuPrint(int i);//根据光标的位置打印出对应的菜单

int move();//移动光标

int main(void);//主函数

③.18-单项选择题标准化考试系统

# 1 问题描述

【任务描述】

设计一个单项选择题的考试系统，可实现自动组卷功能。

【功能描述】

（1）管理员功能

试题管理：每个试题包括题干、4个备选答案、标准答案等信息。可进行试题添加、删除、修改、查询、保存、浏览等操作。

组卷功能：指定题目数、题目总分，生成试卷。将试卷输出到文件，将答案输出到另一个文件。

用户管理：对用户信息进行添加、删除、修改、查询、保存、浏览等操作。

用户答题情况管理：指定用户，统计其做试卷的总次数，以及每次所得的总分。

（2）用户功能

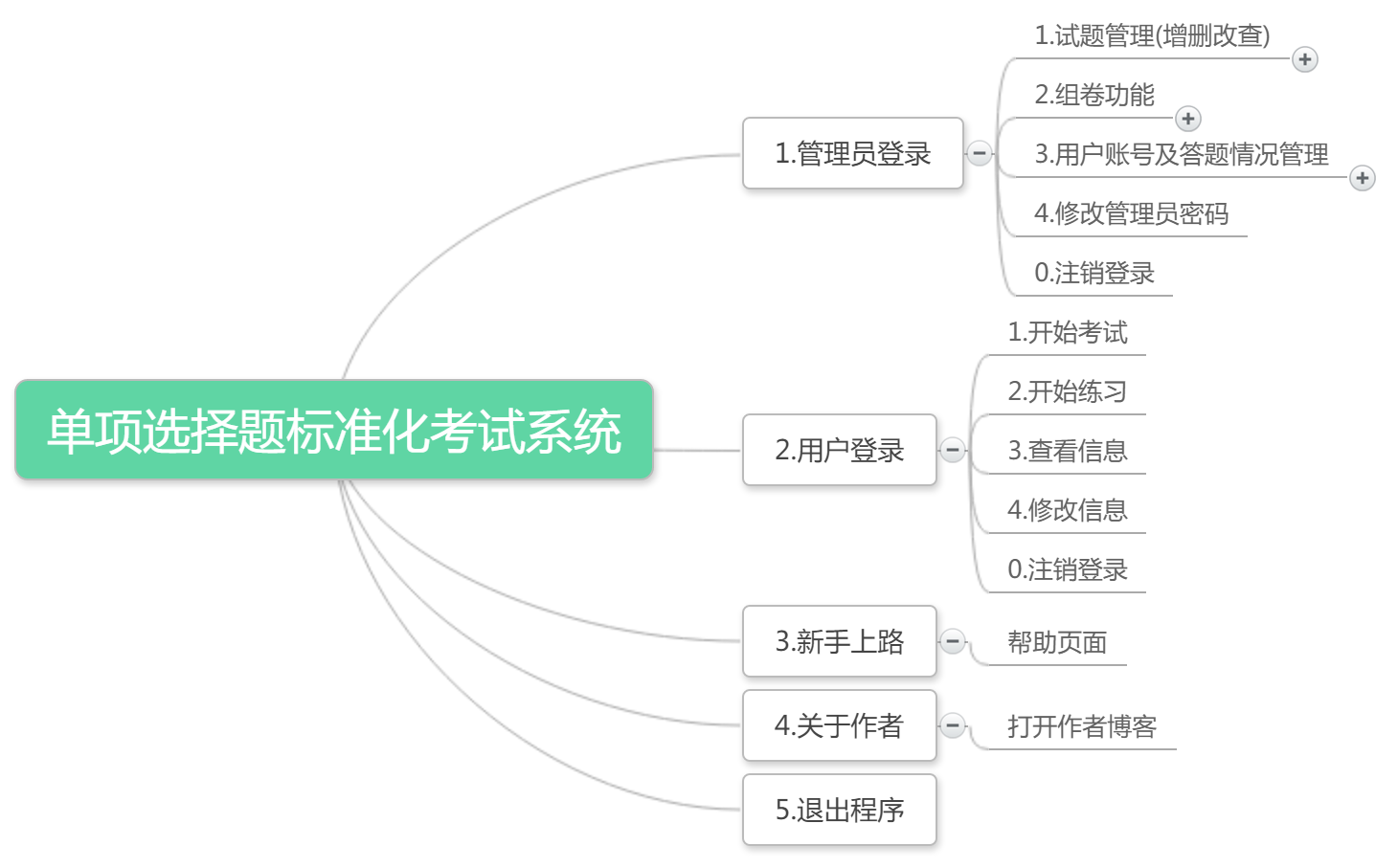
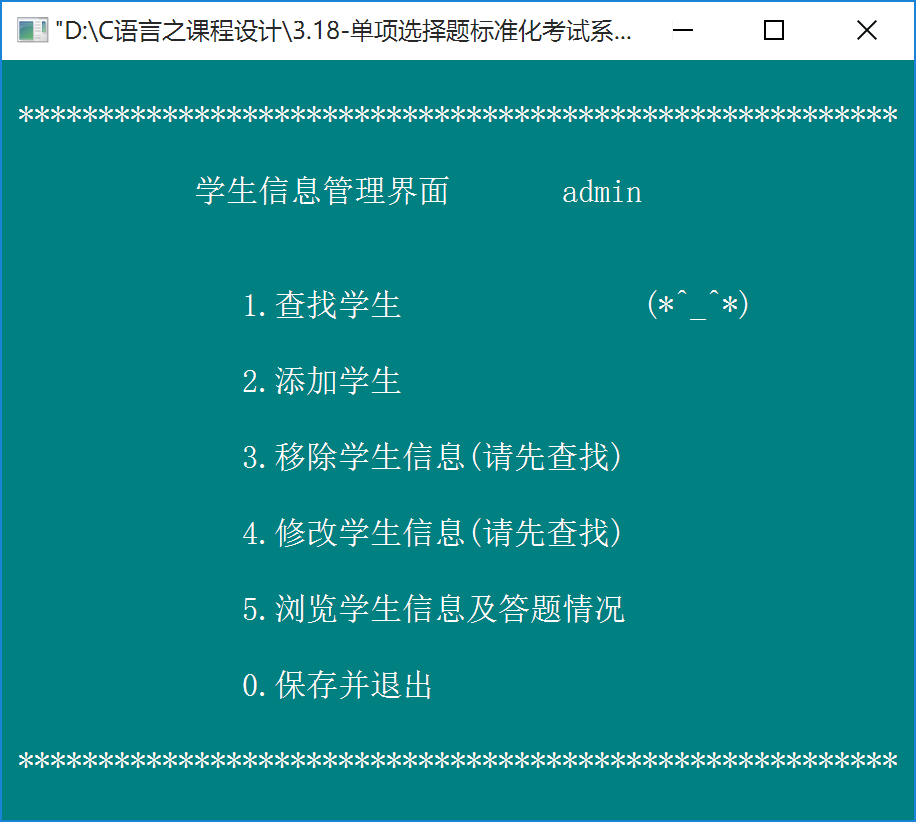
练习功能：用户指定题目数，随机选题。对用户的答案与标准答案进行对比，并最终给出成绩。对错误题目，要能给出正确答案。

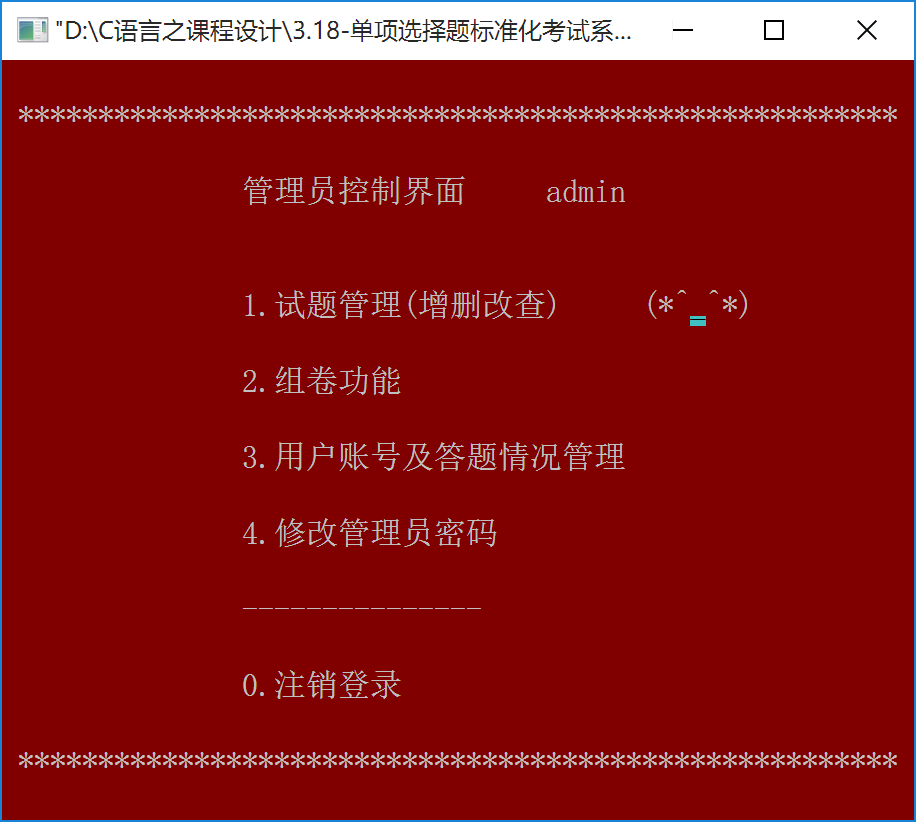
考试功能：用户指定题目数，随机生成总分100分的试卷，系统可根据用户答案与标准答案的对比实现判卷，并给出成绩。并将用户所答试卷、用户的答案、用户所得总分，输出到磁盘文件保存。进行基本试题分析。

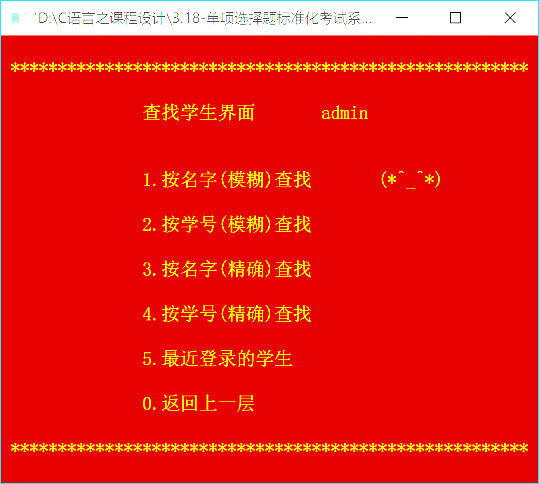
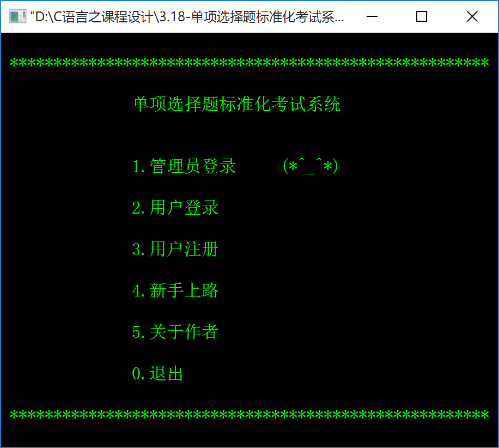
（3）设计提示

管理员和用户分别通过密码登录，进行题目维护和答题操作。用户产生的答题文件，应以用户名+系统时间.txt的形式存储，以便于进行管理。

# 2 程序流程（框图结构）







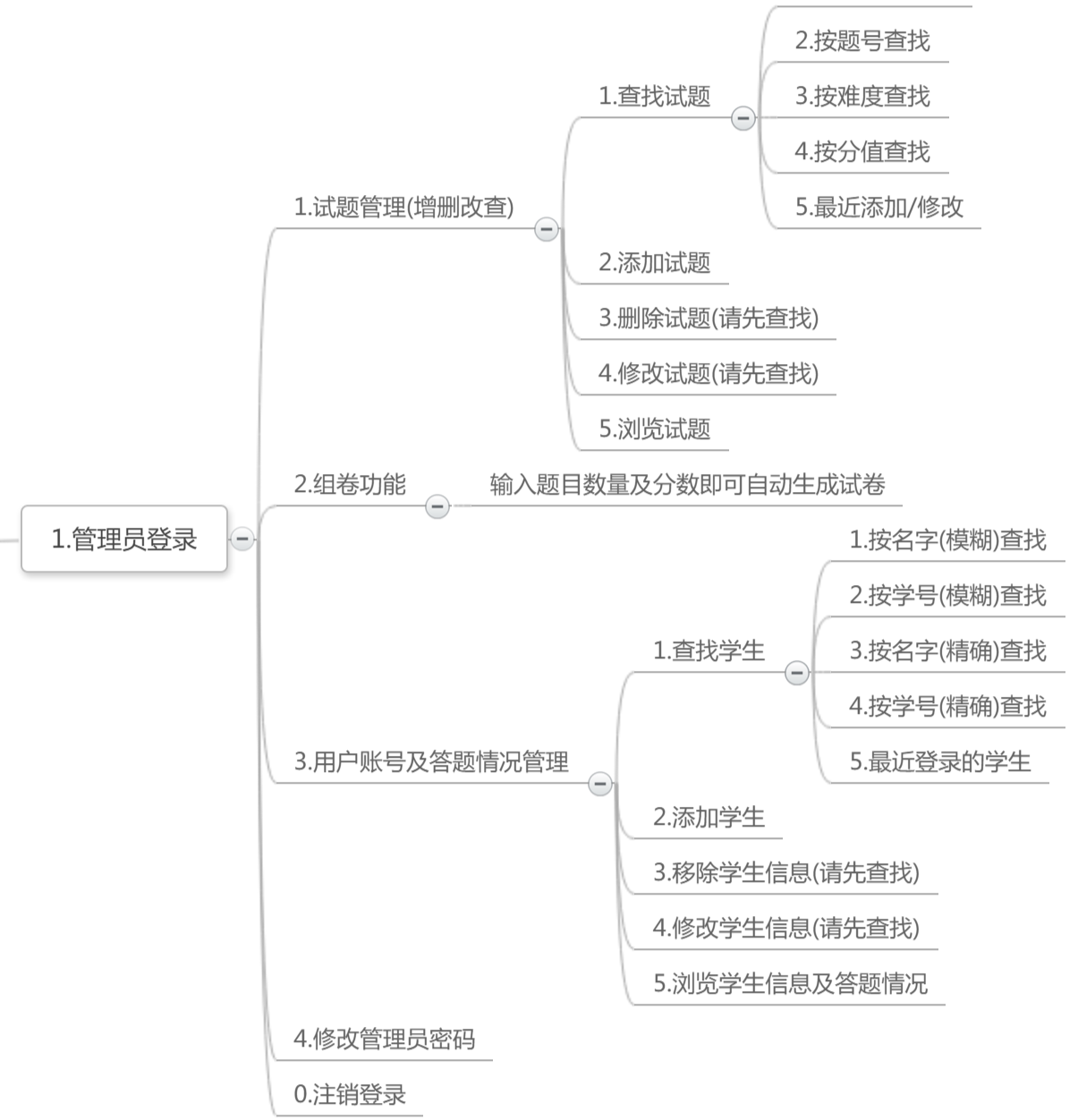


图 x-x

# 3 函数详解

//main.cpp

void main(void);//程序入口

int move\_1(int num,void (\*p)(int) );//集大成的move\_1函数 传入菜单选项的个数已经相应的void menuPrint\_x(int a); 即可打印各种菜单

void menuPrint(int a);//菜单显示函数

void printLaugh(int x, int y);//打印(\*^\_^\*)并把光标移动至笑脸的中央

void goto\_pos(int x, int y);//光标移动函数

int fuzzy\_search(char str[],char str2[]);//模糊查找函数,char str[]为源字符串,char str2[]为查找的关键字

//admin.cpp

int sign\_in();//管理员登录界面

void admin();//管理员控制界面 1.试题管理(增删改查) 2.组卷功能 3.用户账号及答题情况管理 4.修改管理员密码 0.注销登录

void menuPrint\_2(int a);//显示菜单~~管理员登录界面

void question\_admin();//question() -> question.cpp

void auto\_paper();//自动出卷

void user\_admin();//进入函数admin\_user()

void change\_admin\_password();//修改管理员密码

struct info \* find\_info\_num(struct info \* head,int num);//根据题号查找试题

struct info \* find\_info\_score\_num(struct info \* head,int lev);//根据分数查找试题 多半是废了

//question.cpp

void question();//试题管理界面

void menuPrint\_3(int a);//显示菜单~~试题管理界面

struct info \* solo\_info();//申请malloc单个题目

void input\_info(struct info \* head);//新建一个题目并初始化题目信息带头结点的尾插法

void write\_info(struct info \* head);//把题库写入到文件info.txt中

struct info \* read\_info();//从本地文件info.txt读入题库

void free\_info(struct info \* head);//释放链表

void print\_info(struct info \* node);//输出所有题目的信息

void print\_info\_solo(struct info \* node);//输出一个题目全部信息到屏幕

void fprint\_info\_solo(FILE \*fp,struct info \* node);//输出一个题目全部信息到文件fp

//time.cpp

void print\_time(long lt);//输出当前时间到屏幕

void sprint\_time(char \* buf);//输出当前时间到字符数组

//find\_info.cpp

struct info \* find\_info(struct info \* head);//查找试题界面

void menuPrint\_4(int a);//显示菜单~~查找试题界面

struct info \* find\_info\_key(struct info \* head);//按照关键字查找题目,返回题目的上一个指针

void delete\_info\_no(struct info \* find);//删除题目信息

struct info \* find\_info\_no(struct info \* head);//按照题目编号查找题目,返回题目的上一个指针

void change\_info(struct info \* node);//修改题目详细信息

void find\_info\_next(struct info \* node);//选择修改题目信息或删除题目,传来的是题目的上一个节点

struct info \* find\_info\_lever(struct info \* head);//按题目难度查找

struct info \* find\_info\_score(struct info \* head);//按题目分值查找

struct info \* find\_info\_time(struct info \* head);//按最近添加/修改题目

//admin\_user.cpp

void admin\_user();//学生信息管理界面

void menuPrint\_5(int a);//显示菜单~~学生信息管理界面

struct user \* solo\_user();//申请内存

void input\_user(struct user \* head);//学生注册不~带头结点的尾插法

void write\_user(struct user \* head);//把学生信息写入文件中

struct user \* read\_user();//从文件中读入学生信息

void free\_user(struct user \* head);//释放内存

void print\_user(struct user \* node);//输出所有学生信息

void print\_user\_solo(struct user \* node);//单个输出学生信息

//find\_user.cpp

struct user \* find\_user(struct user \* head);//查找学生界面

void menuPrint\_6(int a);//显示菜单~~查找学生界面

struct user \* find\_user\_name\_key(struct user \* head);//按学生姓名(模糊)查找

struct user \* find\_user\_no\_key(struct user \* head);//按学生学号(模糊)查找

void delete\_user\_no(struct user \* find);//按照学号删除用户信息

struct user \* find\_user\_no(struct user \* head);//按照学号查找用户

struct user \* find\_user\_name(struct user \* head);//按照姓名查找用户

void change\_user(struct user \* node);//修改用户学号和密码

void find\_user\_next(struct user \* node);//修改用户信息或删除用户

struct user \* find\_user\_time(struct user \* head);//最近登录的用户

//user.cpp

void user();//学生登录界面

void user\_login(struct user \* head,struct user \* node);//用户登录后

void menuPrint\_7(int a);//显示菜单~~单项选择题标准化考试系统

void printRectangle(int x, int y);//在指定位置输出黄色矩形

void color(const unsigned short color1);//改变颜色---

void exam(struct user \* node);//考试

int isQusetion(struct info \* node,char \*p);//题目是否满足要求

int errorsNum(struct user \* node);//计算用户错了多少题

void answerWrong(struct user \* node,struct info \* temp);//用户答题错误之后...

void execrise(struct user \* node);//练习