第五次上机题解

A. Ausar的劣质复读机 (solution by wjyi)

本题考察同学们读入多组数据的能力，已经考察过很多次，所以是必须熟练掌握的基础题。

本题很多同学使用数组，来记录前一个和前两个数，这样做存在爆空间

这里放出题人的代码给大家参考。

```c
/*
 Author: 陆俊峰(15130)
 Result: AC Submission_id: 1531436
 Created at: Sun Mar 31 2019 11:16:52 GMT+0800 (CST)
 Problem: 1988  Time: 8 Memory: 1704
*/

#include <stdio.h>

int main(){
    int a,b,c,x;
    scanf("%d",&b);
    while(~scanf("%d",&x)){
        c=(x==1)?b+1:a+1;
        printf("%d\n",c);
        a=b;b=c;
    }
    return 0;
}
```

B(hanoi) (solution by prime21)

hanoi塔，经典递归问题。

抽象问题模型，目标问题将n个满足要求的圆盘，从`src`柱子，移动到`tar`柱子上面去，可以借助`tmp`，不妨记作`f(n,src,tar,tmp)`

抽象一次操作:将第i大的圆盘从`src`移动到，`tar`上面去，不妨记作`move(i,src,tar)`

那么考虑分解问题`f(n,src,tar,tmp)`，可以看作:

先将在上面的$n-1$个圆盘按照要求移动到可以借的柱子`tmp`去，然后将执行`move(n,src,tar)`，再把$n-1$个圆盘移动到`tar`上面去。

用函数来描述就是

`f(n,src,tar,tmp) = f(n-1,src,tmp,tar) + move(n,src,tar) + f(n-1,tmp,tar,src)`

注意递归边界

轻松的发现问题是递归的。

代码如下

```c
#include <stdio.h>

void move(int n,char src,char tar) {
    printf("move %d from %c to %c\n",n,src,tar);
}
void f(int n,char src,char tar,char tmp) {
    /*no disk need to move*/
    if (n==0)
        return ;

    f(n-1,src,tmp,tar);
    move(n,src,tar);
    f(n-1,tmp,tar,src);
}

int main() {
    int n;
    scanf("%d",&n);
    /*
     * src = 'A' , tar = 'C' , tmp = 'B'
     * disk = n
     */
    f(n,'A','C','B');
    return 0;
}
```

C. 比 (solution by wjyi)

本题考察double的计算和比较，是基础题目，需要熟练掌握。

```
/*
 Author: 任泉(27995)
 Result: AC Submission_id: 1540739
 Created at: Tue Apr 02 2019 19:21:32 GMT+0800 (CST)
 Problem: 1980  Time: 8 Memory: 1688
*/

#include<stdio.h>
int main()
{
    int i,a,b,x,y;
    double max,m;
    scanf("%d%d",&a,&b);
    max=(double)a/b;
    x=a;y=b;
    for(i=0;i<3;i++)
    {
        scanf("%d%d",&a,&b);
        m=(double)a/b;
        if(m>max)
        {
            max=m;
            x=a;y=b;
        }
    }
    printf("%d %d %.5lf",x,y,max);
    return 0;
}
```

D. 酸奶想成为魔法少女4 (solution by wjyi)

这道题目帮助大家理解最简单的递归，递归是很多题目的基本思想，需要大家熟练掌握。

一个递归函数由固定的几部分组成，其中最重要的是递归运行的终点和调用子程序的逻辑。

```
/*
 Author: 董翰元(28325)
 Result: AC Submission_id: 1540368
 Created at: Tue Apr 02 2019 19:17:40 GMT+0800 (CST)
 Problem: 1989  Time: 199   Memory: 1492
*/

#include <stdio.h>
int f(int x)
{
    switch(x)
    {
        case 1:case 2:case 3:
            return x;
        default:
            return f(x-1) + f(x-3);
    }
}
int main()
{
    int x;
    scanf("%d",&x);
    printf("%d\n",f(x));
    return 0;
}
```

E (solution by prime21)

提交人数581，通过率0.67

歌德巴赫猜想，意图：枚举两个质数，判断是否等于输入，注意到要求a最小，所以枚举a.

```
#include <stdio.h>
int is_prime(int x) {
    if (x == 1) return 0;
    for (int now = 2 ; now * now <= x; now ++)
        if (x%now == 0)
            return 0;
    return 1;
}
int main(){
    int n = 0, x;
    scanf("%d",&n);
    for (int i = 0; i < n; i++) {
        scanf("%d",&x);
        for (int a = 2; a<x; a++) // test each number , one by one.
            if (is_prime(a) && is_prime(x-a)) {
                printf("%d %d\n",a,x-a);
                break;
            }
    }
    return 0;
}
```

F (方程) (solution by prime21)

提交人数514，通过率0.07 通过率有点低，这道题是我从D题挪到F的。

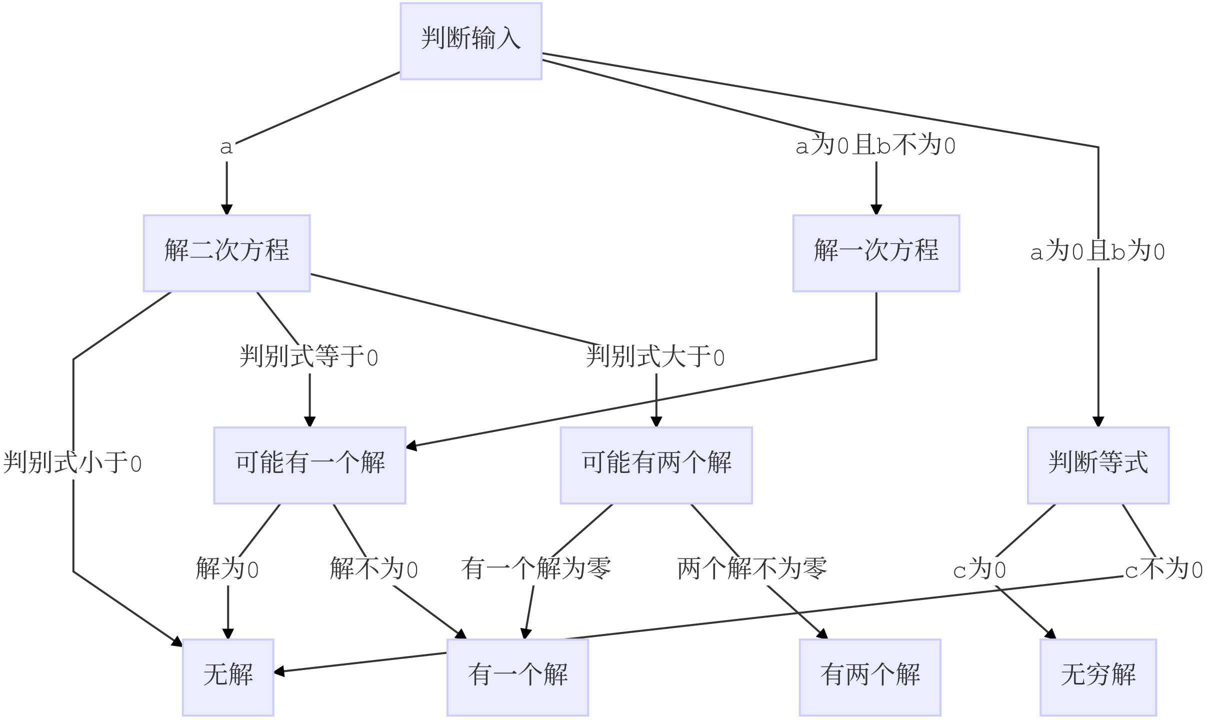
问题可能稍有些难分析，如果能试着把需要统一解决的问题写成一个函数，会轻松很多。

合理的考虑代码的结构，可以让本来显得*很多很杂的情况更加的合理*

注意的问题有如下:

1. 如何判断解出现了0这种不合法的。
2. 根据a, b, c的值，做三种方程处理
3. 二次方程忽略只有一个解的情况

程序思路:



代码里面有细致的注释，和一些处理问题的技巧，有空可以稍微看看。

```c
#include <stdio.h>
#include <math.h>

/*
 * First of all, x can not equal to zero
 * Then , solve ax^2 - bx + c = 0
 * Step 1, check is second-order equation,
 * Step 2, check is Linear equation,
 * Step 3, check is const.
 * */

char no_sol[] = "No solution.";
char inf_sol[] = "Infinite solutions.";
```

```

int is_zero(double now) {
    return fabs(now) < 1e-6;
}

void solve_2(int a,int b,int c) { // solve ax^2+bx+c = 0
    long long delta = b*1ll*b - 4ll*a*c;
    double tmp, x1, x2;

    if (delta < 0) {
        printf("%s",no_sol);
    }
    else if (delta == 0) {
        x1 = -1. * b / (2ll*a);
        if (is_zero(x1))
            printf("%s",no_sol);
        else
            printf("%.2lf",x1);
    }
    else {
        x1 =( -b + sqrt(delta) ) / (2ll * a);
        x2 =( -b - sqrt(delta) ) / (2ll * a);
        if (x1 > x2) {
            tmp = x1;
            x1 = x2;
            x2 = tmp;
        }
        if (!is_zero(x1)) printf("%.2lf ",x1);
        if (!is_zero(x2)) printf("%.2lf ",x2);
    }
}

void solve_1(int a,int b) { // solve ax+b = 0
    double x = -1.*b / a;
    if ( is_zero(x) )
        printf("%s",no_sol);
    else
        printf("%.2lf\n",x);
}

int main(){
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    if (a != 0) {
        solve_2(a,-b,c);
    }
    else if (b != 0) {
        solve_1(b,-c);
    }
    else
        printf("%s", (c==0) ? inf_sol : no_sol );
    return 0;
}
```

G (solution by prime21)

提交人数71，通过率0.15

可能是一道不太难的题，没想到大家被F题拦住了，

依次判断就号，可以作为**课后的字符串练习题目!!**

```c
#include<stdio.h>
#include<string.h>
#define GG return printf("Chawuciren\n"), 0
#define isDigit(x) (x >= '0' && x <= '9')
#define upper(x) (isDigit(x) ? x : x >= 'a' ? x - 'a' + 'A' : x)
const int w[] = { 0, 7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2 };
const char chk[] = { '1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2' };
const int days[2][13] = { {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}, {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31} };
char s[300];

int Int(int l, int r)
{
    int ret = 0, i;
    for (i = l; i <= r; ++i)
        ret = ret * 10 + (s[i] - '0');
    return ret;
}

int isSp(int year)
{
    return ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0);
}

int check(int y, int m, int d)
{
    int sp = isSp(y);
    if (m < 1 || m > 12) return 0;
    if (d < 1 || d > days[sp][m]) return 0;
    return 1;
}

int main()
{
    int i, sum = 0;
    scanf("%s", s + 1);
    if (strlen(s + 1) != 18) GG;
    for (i = 1; i <= 17; ++i) {
        if (isDigit(s[i]))
            sum += (s[i] - '0') * w[i];
        else GG;
    }
    sum %= 11;
    if (chk[sum] != upper(s[18])) GG;
    int y, d, m, ans = 0, sp;
    y = Int(7, 10);
    m = Int(11, 12);
    d = Int(13, 14);
    if (!check(y, m, d)) GG;
    while (1) {
        ++ans; sp = isSp(y);
        if (y == 2019 && m == 4 && d == 2) break;
        if (++d > days[sp][m]) ++m, d = 1;
        if (m > 12) ++y, m = 1;
    }
    printf("%d\n", ans - 1);
    return 0;
}
```