

# 第四次上机题解

---

## Published by *login*

本次上机共9道题，以下题目按通过人数排序，代码主要为出题人代码，也有部分同学的代码。

- [第四次上机题解](#)
  - [C login不会梦到银行利息 \(login\) \(810/925\)](#)
  - [A 高速公路 \(prime21\) \(655/986\)](#)
  - [B 三角函数\(prime21\) \(329/653\)](#)
  - [D 酸奶织方帕 \(酸奶\) \(113/192\)](#)
  - [E 零 \(prime21\) \(100/527\)](#)
  - [G lx的最短正整数序列 \(lx\) \(82/237\)](#)
  - [F Max的DDL \(Max.D\) \(67/164\)](#)
  - [H 这道题没有题面描述 \(MountVoom\) \(31/213\)](#)
  - [I HugeGun学姐的猜想 \(HugeGun学姐\) \(13/27\)](#)

## C login不会梦到银行利息 (login) (810/925)

**题意：**利息计算，过了n年，年利率是a%，本金是x元，每年计算完利息后，收取b元的年费。

即 $a_n = a_{n-1} \times (1 + a\%) - b$ ，求 $a_n$

**解题思路：**按照题意简单的使用循环计算即可。

注意输入的a为一个百分比，需要除以100；以及a的后面接上了%，需要使用格式化读入（注意'%'的转义为"%%"）。

代码如下

```
//Provided by login
#include<stdio.h>

int main()
{
    int n;
    double a,x,b;
    scanf("%d%lf%lf%lf",&n,&a,&x,&b);
    a/=(double)100;
    for(int i=1; i<=n; i++)
    {
        x=x*(1+a)-b;
    }
    printf("%.3lf",x);
    return 0;
}
```

## A 高速公路 (prime21) (655/986)

**题意：**给出当前速度 $v_{now}$ 和限速 $v_{lim}$ ，在 $v_{now}/v_{max} < 110\%$ ， $110\% \leq v_{now}/v_{max} < 150\%$ ， $150 \leq v_{now}/v_{max}$ 三种情况分别输出不同的结果。

**解题思路：**使用if语句判断三种情况即可

**关于这题的“坑”：**由于涉及小数，需要考虑浮点数误差的问题。同时，OJ上和dev的编译器版本较老，似乎在浮点数处理上和新版不同，导致如果直接使用 $<$  和  $\leq$ 比较会出现问题（如 $200*1.1$ 可能会出现220.000000003的结果）。因此，在比较浮点数的时候，如果是 $a < b$ 这种情况，可以写成 $a < b - eps$ ，同理， $a \leq b$ 可写成 $a \leq b + eps$ ，这里的eps是一个很小的正数，一般可以取 $1e-10$  ( $10^{-10}$ )~ $1e-5$  ( $10^{-5}$ )。

**出题人的吐槽（转述）：**这题本来是没有打算卡大家的，写标称、造数据的时候出题人也没有打算需要eps。但是由于OJ编译器版本问题和系统问题，会出现一定的浮点误差的问题，因此使用乘法判断的话，在相等的情况会出现问题。（如果使用除法来判断的话，此题可以通过。）

最后此题删减了一些数据，重新评测了两次，让一些做法过了（其实本来出题人也是不想卡什么做法的）。但是真正正确的写法是需要eps或者直接使用整数来判断的，建议大家即使通过了这道题，也尝试一下正确的写法。

以下是原代码

```

//Provided by prime21
#include <stdio.h>

int main(){
    int now,lim;
    double ratio;
    scanf("%d%d",&now,&lim);
    ratio = now *1. /lim * 100;
    if (ratio < 110)
        printf("OK");
    else
    {
        printf("Exceed %.0lf%c. ",ratio-100,'%');
        if (ratio<150)
            printf("Ticket 200");
        else
            printf("License Revoked");
    }
    return 0;
}

```

以下是使用eps的代码

```

//Provided by prime21
//Modified by login
#include <stdio.h>
const double eps = 1e-6;
int main(){
    int now,lim;
    double ratio;
    scanf("%d%d",&now,&lim);
    ratio = now *1. /lim * 100;
    if (now < lim * 1.1 - eps)
        printf("OK");
    else
    {
        printf("Exceed %.0lf%c. ",ratio-100,'%');
        if (now < lim*1.5-eps)
            printf("Ticket 200");
        else
            printf("License Revoked");
    }
    return 0;
}

```

以下是使用整数判断的代码

```

//Provided by login
#include <stdio.h>
const double eps = 1e-6;
int main(){
    int now,lim;
    double ratio;
    scanf("%d%d",&now,&lim);
    ratio = now *1. /lim * 100;
    int can = 0;
    if(lim%10 == 0) //也可写成lim*10%100
        can = now < lim+lim/10;           //lim*1.1刚好为整数时
    else
        can = now <= lim+lim/10;          //lim*1.1不为整数时
    if (can)
    {
        printf("OK");
    }
    else
    {
        printf("Exceed %.0lf%c. ",ratio-100,'%');
        if(lim%2 == 0) //也可写成lim*50%100
            can = now < lim+lim/2;        //lim*1.5刚好为整数时
        else
            can = now <= lim+lim/2;       //lim*1.5不为整数时
        if (can)
            printf("Ticket 200");
        else
            printf("License Revoked");
    }
    return 0;
}

```

## B 三角函数(prime21) (329/653)

**题意：**使用泰勒级数计算 $\cos(x)$ ,只需要计算到第m项。

$$\text{即 } \cos x \approx \sum_{n=0}^m \frac{(-1)^n}{(2n)!} x^{2n}$$

**解题思路：**使用for运算即可。计算阶乘和次幂的朴素的方法是每次使用for来计算。

经观察可发现，每次的阶乘 $(2i)!$ ，次幂 $(-1)^n$ 、 $x^{2i}$ ，其实可以是 $(2(i-1))! \times (2i-1) \times (2i)$ 、 $(-1)^{n-1} \times (-1)$ 、 $x^{2(i-1)} \times x^2$ ，因此可以使用变量记下当前的三个值，每次先计算出这次的这三个值，再运算。（这里蕴含了一定的递推思想）其中 $(-1)^n$ 还直接与奇偶有关，也可直接通过判断奇偶得到。因此直接使用一层for循环即可。

注意当阶乘和次幂足够大的时候，我们可以忽略精度问题，于是使用double储存。

代码如下

```
//Provided by prime21
#include <stdio.h>

double now=1.;
double sum=0;

double x;
int m;

int main()
{
    int i=0;
    scanf("%lf%d",&x,&m);
    for(i=0;i<=m;i++)
    {
        if (i&1)
            sum-=now;
        else
            sum+=now;
        now=now*x*x;
        now=now/((i+i+1)*(i+i+2));
    }
    printf("%.6lf\n",sum);
    return 0;
}
```

---

## D 酸奶织方帕 (酸奶) (113/192)

**题意：**输出一个图形，一个多圈的正方形，由外圈向内依次是从a到z的大写字母。

**解题思路：**观察符号位置和行号列号的关系，每层可以分为左（依次变大）中（全部一样）右（依次缩小）三部分，使用for语句输出即可。

注意字母的ASCII码是连续的，可以直接通过for、%c（或者putchar()）和数字来输出。

**（这里建议大家认真思考一下字符和数字的对应性，即为什么可以用数字输出字符）**

以下为代码

```

//Provided by 酸奶
#include <stdio.h>
#include <stdlib.h>

int x,n;

int main()
{
    scanf("%d",&x);
    n=2*x-1;
    for(int i=1;i<=x;i++)
    {
        for(int j=i;j>1;j--)
            printf("%c",'A'+i-j);
        for(int j=1;j<=n-2*(i-1);j++)
            printf("%c",'A'+i-1);
        for(int j=2;j<=i;j++)
            printf("%c",'A'+i-j);
        printf("\n");
    }
    for(int i=x-1;i>=1;i--)
    {
        for(int j=i;j>1;j--)
            printf("%c",'A'+i-j);
        for(int j=1;j<=n-2*(i-1);j++)
            printf("%c",'A'+i-1);
        for(int j=2;j<=i;j++)
            printf("%c",'A'+i-j);
        printf("\n");
    }
    return 0;
}

```

## E 零 (prime21) (100/527)

**题意：**求 $n!$ 的末尾有多少个0。

**解题思路：**出现零情况的话，我们考虑把 $n!$ 这个数分解为质因数后，只有 $2*5$ 这种情况会有贡献。显然 $1-n$ 这些数中，质因数的2的个数是大于5的个数的。因此只需统计 $1-n$ 中质因数5的个数。注意25即其倍数的质因数有2个5，125及其倍数的质因数有3个5。

代码如下

```
//From 许圣林
#include<stdio.h>

int main(){
    long long a,c=0;
    scanf("%lld",&a);
    while(a!=0){
        a/=5;
        c+=a;
    }
    printf("%lld",c);
    return 0;
}
```

## G lx的最短正整数序列 (lx) (82/237)

**题意：**对于一个数 $m$ ，找出最短的，形如“2,3,4,5”这样的连续的正整数数列 $\{a_n\}$ （注意长度 $>1$ ），使得 $\sum_1^n a_n = m$ 。不存在则输出-1。

**解题思路：**枚举每一个数列的起始数，然后判断以这个为起始的数列是否满足条件，可以的话，记下起始和长度。注意如果找到新的，需要比较大小后覆盖掉这个。

当然，这题有一定的方法优化：

- 最短的肯定是起始值最大的那个，从大向小枚举，找到第一个即可。（或者先枚举长度亦可）
- 以一个起点开始的数列可以使用等差数列求和公式直接解出，即枚举 $i$ ，查看方程 $\frac{(i+i+x-1) \times x}{2} = n$ 是否有大于1的整数解。（当然枚举长度 $x$ ，解出初始位置 $i$ 亦可）

代码如下

```

//Provided by lx
#include <stdio.h>
int main()
{
    int s, i, j, k, st, sum, got = 0, min_st = 0, min_end = 1e9;
    scanf("%d", &s);
    for (i = 1; i <= s/2; i++) {
        sum = i;
        for (j = i+1; sum < s; j++) {
            sum += j;
            if(sum == s && j-i < min_end-min_st) {
                min_st = i;
                min_end = j;
                got++;
            }
        }
    }
    if (got) {
        printf("%d = %d", s, min_st++);
        for (i = min_st; i <= min_end; i++) {
            printf(" + %d", i);
        }
        printf("\n");
    } else {
        printf("-1\n");
    }
    return 0;
}

```

优化的代码如下(枚举长度)



```

//Provided by mogg
#include <stdio.h>
int main()
{
    int n,k;

    scanf("%d", &n);

    for (k = 2; k <= n / 2; k++)
    {
        if ((2 * n) % k == 0)
        {
            int tmp = 2 * n / k + 1 - k;
            if ((tmp & 1) == 0 && tmp != 0)
            {
                tmp /= 2;
                printf("%d = ", n);
                for(int i = 0; i < k; i++)
                {
                    printf("%d", tmp + i);
                    if (i != k - 1) printf(" + ");
                }
                break;
            }
        }
    }
    if (k == n / 2 + 1) printf("-1");
    return 0;
}

```

## F Max的DDL (Max.D) (67/164)

**题意：**判断某一天是否为周六周日。可以参考wikipedia：[星期的计算](#)

**解题思路：**本题方法有很多，主要是为了算出今天是星期几：

- 方法1：求出公元1年1月1日到现如今的天数，循环年月
- 方法2：[基姆拉尔森公式](#)
- 方法3：[蔡勒公式](#)
- .....

代码如下：

```

//Provide by Max.D
//Modified by login
//方法1: 求出公元1年1月1日到现如今的天数, 循环年月
#include<stdio.h>

int y,m,d;

int main()
{
    while(scanf("%d%d%d",&y,&m,&d)==3)
    {
        int tot=0;
        for(int i=1;i<y;i++)
            tot+=i%4==0&&i%100| |i%400==0?366:365;
        for(int i=1;i<m;i++)
        {
            if(i==2)
                tot+=28;
            else if(i==4| |i==6| |i==9| |i==11)
                tot+=30;
            else
                tot+=31;
            if((y%4==0&&y%100| |y%400==0)&&i==2)
                tot++;
        }
        tot+=d;
        puts((tot-1)%7+1>5?"DDL is the primary productive force.":"All work and no play");
    }
    return 0;
}

```

```

//Provide by Max.D
//方法2: 基姆拉尔森公式
#include<stdio.h>

int y,m,d;

int main()
{
    while(scanf("%d%d%d",&y,&m,&d)==3)
    {
        if(m<3)
            m+=12,y--;
        puts((d+2*m+3*(m+1)/5+y+y/4-y/100+y/400)%7+1>5?"DDL is the primary productive force.":"All work and no play");
    }
    return 0;
}

```

```

//Provide by Max.D
//方法3: 蔡勒公式
#include<stdio.h>

int y,m,d;

int zeller(int year,int month,int day)//根据日期判断出星期几
{
    if(month==1||month==2)
    {
        month+=12;
        year--;
    }
    int c=year/100;
    int y=year%100;
    int m=month;
    int d=day;
    int W=c/4-2*c+y+y/4+26*(m+1)/10+d-1;
    if(W<0)
        return (W+(-W/7+1)*7)%7;
    return W%7;
}

int main()
{
    while(scanf("%d%d%d",&y,&m,&d)==3)
    {
        if(m<3)
            m+=12,y--;
        puts(zeller(y,m,d)==0||zeller(y,m,d)==6?"DDL is the primary productive");
    }
    return 0;
}

```

## H 这道题没有题面描述 (MountVoom) (31/213)

**题意:** 求 $(\sum_{i=1}^n i!) \bmod (10^9 + 7)$ , 多组输入

**解题思路:** 简单的递推, 使用for, 并记下所有值即可。

以下为代码

```

//From xsy学姐
#include<stdio.h>
#include<stdlib.h>
#define maxn 100010
#define mod 1000000007
int q, i, n, fac[maxn], sum[maxn];

int main()
{
    fac[0] = 1;
    for (i = 1; i <= 100000; ++i)
    {
        fac[i] = 1ll * fac[i - 1] * i % mod;
        sum[i] = (sum[i - 1] + fac[i]) % mod;
    }

    scanf("%d", &q);
    while (q--) {
        scanf("%d", &n);
        printf("%d\n", sum[n]);
    }
    return 0;
}

```

## I HugeGun学姐的猜想 (HugeGun学姐) (13/27)

**题意：**

给定 $n(1 \leq n \leq 10)$ 个正整数 $x_1, x_2 \dots x_n (1 \leq x_1, x_2, \dots, x_n \leq 10^8)$ ，然后进行类似角谷猜想中的变换：

1. 如果它们全都是奇数，则将他们分别变为 $3 \times x_1 + 1, 3 \times x_2 + 1, \dots, 3 \times x_n + 1$ ;
2. 如果它们中至少有一个偶数，则将它们分别变为 $\lceil \frac{x_1}{2} \rceil, \lceil \frac{x_2}{2} \rceil, \dots, \lceil \frac{x_n}{2} \rceil$ ; 其中,  $\lceil x \rceil$ 表示对 $x$ 向上取整。

一直重复以上过程，在猜想中，这 $n$ 个数最后必将同时等于1或者陷入一个循环节长度为3的循环。

按照以上方法变化，输出每次变换后的序列，直到它们都等于1或已变换66666次为止。

**解题思路：**按照题面意思模拟即可。

```
//Provided by HugeGun
#include<stdio.h>
int n,a[11],bo,i,sum;
int main()
{
    scanf("%d",&n);
    for(i=1;i<=n;i++)scanf("%d",&a[i]);
    while(1)
    {
        bo=1;
        for(i=1;i<=n;i++)if(a[i]%2==0)bo=0;
        for(i=1;i<=n;i++)if(!bo)a[i]=(a[i]+1)/2;
        else a[i]=3*a[i]+1;
        sum++;
        for(i=1;i<=n;i++)printf("%d ",a[i]);
        puts("");
        bo=1;
        for(i=1;i<=n;i++)if(a[i]!=1)bo=0;
        if(bo||sum==66666)break;
    }
    return 0;
}
```