Problem A. 计算幂次

题目描述

对于给定的 a,b,p 某只蒟蒻想知道以下式子的值是多少：

$$求：a^b \ mod \ p$$

超大声：快去看提示！！

输入格式

一行，三个整数a,b,p $(1 < a,b,p \leq 10^9)$。

输出格式

一行，一个整数表示$a^b \ mod \ p$。

样例输入

```
2 3 998244353
```

样例输出

```
8
```

Hint

调用以下函数可以得到 $a^b \ mod \ p$ 的值：

```
int qpow(int a,int b,int p)
{
    int ret=1;
    while(b)
    {
        if(b&1) ret=1ll*ret*a%p;
        a=1ll*a*a%p,b>>=1;
    }
    return ret;
}
```

Problem B. 错误的斐波那契数列

题目描述

某只蒟蒻在**程设课的ppt**上学习到了如何计算斐波那契数列的第 n 项，可是ta太弱了把公式记错成了以下的样子：

$$f(n) = f(n-3) + f(n-1)$$

请你预计一下这只蒟蒻算出的“错误的斐波那契数列”的第 n 项是多少：

输入格式

一行，一个整数 $n, 1 \leq n \leq 20$。

输出格式

一行，一个整数$f(n)$。

样例输入

```
7
```

样例输出

```
6
```

样例解释

$$f(1)=f(2)=f(3)=1$$
$$f(4)=f(3)+f(1)=1+1=2$$
$$f(5)=f(4)+f(2)=2+1=3$$
$$f(6)=f(5)+f(3)=3+1=4$$
$$f(7)=f(6)+f(4)=4+2=6$$

Hint

如果不知道从何下手的话不如去翻一翻ppt~

可以使用递归解决这个问题~

Problem C. 输出距离

题目背景

自定义一个函数，计算两个三维坐标点的距离。请大家体会“利用函数实现模块化编程，可以提高程序开发效率”。

题目描述

输入 4 个三维坐标点，按要求顺序输出它们两两之间的距离。

输入格式

共 4 行。第 i 行三个整数 x_i, y_i, z_i ，表示第 i 个三维坐标点，数字之间用一个空格隔开。 $-100 \leq x_i, y_i, z_i \leq 100$。

输出格式

共 6 行，每行分别输出：坐标点 2 与坐标点 4 、坐标点 1 与坐标点 3 、坐标点 2 与坐标点 3 、坐标点 3 与坐标点 4 、坐标点 1 与坐标点 2 、坐标点 1 与坐标点 4 的距离。结果保留 2 位小数。

输入样例

```
20 21 10
21 10 17
10 17 20
17 20 21
```

输出样例

```
11.49
14.70
13.38
7.68
13.08
11.45
```

样例说明

无

Hint

提示：使用数组而非 12 个单独的变量存储坐标，访问坐标以及调用函数时更方便。

或许你可以编写这样一个函数：

```
//计算点(x1,y1,z1)到点(x2,y2,z2)的距离
double dis(double x1,double y1,double z1,double x2,double y2,double z2)
{
    //body
}
```

那么如何计算距离呢：

$$dis = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Problem D. xf学组合数学

题目描述

“打表出奇迹，暴力得榜一。”——沃兹基硕德

xf遇到一个组合数学的题目：从 m 个数中取 n 个无序的数字一共有多少种取法？ 你能帮帮他解决这个问题吗？

输入格式

输入一个正整数t，代表数据组数$(t \leq 10)$，接下来t行每行输入两个正整数，先输入 m ,再输入$n, 0 < n \leq m \leq 20$。

输出格式

一共 t 行，每行输出一个数，代表从 m 个数里选 n 个数有多少种选法。

样例输入

```
3
2 1
4 2
20 2
```

样例输出

```
2
6
190
```

Hint

还记得高中学过的组合数公式嘛?

$$设C(m,n)表示从m个数中取n个无序的数字一共有多少种取法$$
$$则：C(m,n)=C(m-1,n-1)+C(m-1,n)$$
$$即：从m个数中取n个数取法数=从m-1个数中取n-1个数取法数+从m-1个数中取n个数取法数$$

或者可以试试递归QAQ

边界条件是什么样的呢:

$$若m<n,显然无法从m个数中选出n个数！因此C(m,n)=0(当m<n时)$$
$$从m个数中选取0个数，显然只有一种选法就是什么都不选！因此C(m,0)=1$$
$$从m个数中选取1个数，显然有m种选法！因此C(m,1)=m$$

如果还是不会：不如去看看ppt~

Problem E. GPA的计算

题目背景

GPA，或称平均学分绩点，是用来衡量学生学习成果的重要指标。 我们这里的 GPA 采用 4 分制（即满分为 4 分）。

算法如下： 设某门课程的百分制成绩为 x ，则相应的GPA为:

$$GPA=4-\frac{3\times(100-x)^2}{1600}(60\le x\le 100)$$

当分数为 60 分时 GPA 为 1，60 **分以下** GPA **为** 0。

现输入$N(1\le N\le 10^6)$ 门课的百分制成绩 $x_1,x_2,\ldots\ldots,x_N$ 和每门课对应的学分 $h_1,h_2,\ldots\ldots,h_N$。 通过各门课 GPA 计算**总 GPA** 的公式为:

$$总GPA=\frac{(GPA_1\times h_1+GPA_2\times h_2+\cdots+GPA_N\times h_N)}{(h_1+h_2+\cdots+h_N)}$$

题目描述

现在告诉你某只蒟蒻 N 门课程的**百分制**分数，请你算出ta的**总GPA**。

输入格式

第一行一个整数 N 表示课程的数量。

接下来 N 行，每行两个整数x_i,h_i，分别表示第 i 门课程的**百分制**分数和学分。

输出格式

一行一个浮点数表示蒟蒻的总GPA，保留两位小数。

样例输入

```
2
85 3
60 2
```

样例输出

```
2.55
```

Problem F. 素数日期

题目描述

有一只蒟蒻喜欢素数，尤其喜欢**素数日期。**

如果使用八位数字表示一个日期其中前 4 位代表年，接下来 2 位代表月，最后 2 位代表日（即形如`yyyymmdd`的格式），当这个8位数是**素数**的时候这只蒟蒻就会非常开心。

如：20020103、20101219 都是**素数日期**，而20210928、20210926 等都不是**素数日期。**

现在，请你帮这位蒟蒻计算一下，再经过多少天ta可以再经历一个令人开心的**素数日期**，并且这个**素数日期**是星期几呢。

注：快去看提示！！

输入格式

本题存在多组数据（数据组数$\leq 10^3$），对于每组数据：

一行，一个8位整数n表示今天的日期，且$20000101 \leq n \leq 99991231$。

输出格式

对于每组数据分别输出答案：

`下一个素数日期 经过的天数 星期几`

若直至 99991231 都没有再经历一个令人开心的素数日期，输出 -1 。

样例输入

```
20211021
20211224
20210526
99991216
```

样例输出

```
20211031 10 Sunday
20220103 10 Monday
20210609 14 Wednesday
-1
```

Hint

- 调用以下isprime函数可以判断一个数是否为素数：(请把qpow函数也复制进代码~)
    - 以下算法是采用根据费马小定理的逆定理判断素数。算法的核心思想是使用随机数反复验证费马小定理的逆定理是否成立。如果经过多次（如20次）验证后，该定理均成立，则以大概率认为这个数是一个素数。
    - 采用费马小定理的逆定理判断一个数是否是素数，尤其当数值很大时，算法效率远远优于传统算法。

```
int qpow(int a,int b,int p)
{
    //求(a^b) mod p 在isprime函数中会用到
    int ret=1;
    while(b)
    {
        if(b&1) ret=1ll*ret*a%p;
        a=1ll*a*a%p,b>>=1;
    }
    return ret;
}
```

```
int isprime(int n)//返回1表示n为素数，0表示n不是素数
{
    if(n==2||n==3) return 1;
    if(!(n&1)) return 0;
    int m=n-1,a,tmp,ans,cnt=0;
    while(!(m&1))
        m>>=1,cnt++;
    int rd=20011224,seed=998244353,seed2=20217371;
    for(int i=0;i<20;i++)
    {
        rd=rd*seed+seed2;
        if(rd<0) rd=-rd;
        a=rd%(n-1)+1;
        //验证费马小定理的逆定理
        tmp=qpow(a,m,n);
        for(int j=0;j<cnt;j++)
        {
            ans=1ll*tmp*tmp%n;
            if(ans==1)
            {
                if(tmp!=1 && tmp!=n-1)
                    return 0;
                break ;
            }
            tmp=ans;
        }
        if(ans!=1) return 0;
    }
    return 1;
}
```

- 调用以下函数可以计算一个日期为星期几：

  返回值的含义：0-星期日；1-星期一；2-星期二；3-星期三；4-星期四；5-星期五；6-星期六

```
int Zeller(int y,int m,int d)
{
    if(m==1 || m==2) y--,m+=12;
    int c=y/100;y%=100;
    return ((y+y/4+c/4-2*c+26*(m+1)/10+d-1)%7+7)%7;
}
```

- 调用以下函数可以判断某个八位数是否是合法的日期：

```
int day[]={0,31,28,31,30,31,30,31,31,30,31,30,31};//平年每个月的天数
int isday(int y)
{
    int d=y%100;y/=100;//日
    int m=y%100;y/=100;//月
    if(m<1 || m>12) return 0;//月必须在1-12之间
    else if(d==29&&m==2)//若是2月29日
        return (y%100&&y%4==0)||y%400==0;//合法日期的年份必须为闰年
    return d>=1 && d<=day[m];//日的区间
}
```

Problem G. 分数加法

题目描述

`XIAO7`得到了两个分数，请你帮她完成两个分数的加法运算。

输入格式

本题存在多组数据（数据组数≤ 10），对于每组数据：

输入两个合法分数，形如

```
a/b c/d
```

保证输入的**整数**a,b,c,d的绝对值 $\leq 1\times10^{16}$，约分后，$|a|,|b|,|c|,|d| \leq 1\times10^{7}$。

输出格式

输出一个最简分数，形如

```
x/y
```

输入输出分数的符号都在分数前面，保证最终结果不为 0 或 1。

样例输入

```
1/4 5/6
1/2 1/10
-1/3 1/2
-1/3 -1/2
```

样例输出

```
13/12
3/5
1/6
-5/6
```

Hint

不如把已经练习过很多次的最大公约数/最小公倍数写成一个函数~

```
 ∧    ∧  ❀ ❀
( ˘ ˬ ˘❀祝你AC❀
 ヽ  つ \     /
    ∪∪   /  \
```

author XIAO7

Problem H. 可爱的数列

题目描述

对于任意一个长度为偶数的由 0 和 1 组成的数列，某只蒟蒻都赋予了它一个可爱程度：

- 对于数列$[0]$，其可爱程度为$f([0])=0$。
- 对于数列$[1]$，其可爱程度为$f([1])=1$。
- 对于长度为 $2k$ 的数列$[a_1,a_2,\cdots,a_k,a_{k+1},\cdots,a_{2k}]$，其可爱程度
  $f([a_1,a_2,\cdots,a_k,a_{k+1},\cdots,a_{2k}])=(f([a_1,a_2,\cdots,a_k])+f([a_{k+1},\cdots,a_{2k}]))\ \&\ (f([a_1,a_2,\cdots,a_k])-f([a_{k+1},\cdots,a_{2k}]))$

现在请你告诉这只蒟蒻，这个数列到底有多可爱！

即给定一个 01 数列$[a_1,a_2,\cdots,a_n],n=2^k,k\in Z^+$，求$f([a_1,a_2,\cdots,a_n])$。

输入格式

第一行一个整数 T，$1 \leq T \leq 50$，表示数据的组数，接下来若干行对于每组数据：

两行，第一行一个整数 n，$n \leq 10^3$ 且 $n = 2^k, k \in Z^+$，表示数列的长度；

第二行 n 个整数分别表示 $a_i, 1 \leq i \leq n,$ 且 $a_i \in \{0, 1\}$。

输出格式

一行，一个整数表示 $f([a_1, a_2, \cdots, a_n])$。

样例输入

```
2
8
1 0 1 1 1 0 1 0
8
0 0 1 0 0 0 1 0
```

样例输出

```
1
0
```

Hint

使用递归模拟一下吧~

考虑一下**递归的三要素**（递归基本情况、递归约束条件和递归表达式）在这道题里是什么呢~

Problem I. $Verilog$ 匹配

背景描述

$Verilog$ 是一种硬件描述语言，以文本形式来描述数字系统硬件的结构和行为的语言，用它可以表示逻辑电路图、逻辑表达式，还可以表示数字逻辑系统所完成的逻辑功能，它是计算机学院的同学们在大二上的计算机组成原理课程中需要学习的一门语言。

以下是一段 $Verilog$ 代码示例:

```
    always @(posedge clk) begin
        if(reset) begin
            for(i=0;i<4096;i=i+1) begin
                RAM[i]<=0;
            end
        end
        else begin
            if(MEM_WRITE_ENABLED) begin
                case(S_OP)
                    1: begin//sw
                        RAM[MEM_WRITE_A[13:2]] <= MEM_WRITE_DATA;
                        $display("%d@%h: *%h <= %h",$time,PC,{MEM_WRITE_A[31:2],2'b0},MEM_WRITE_DATA);
                    end
                    2: begin //sh
                        case (MEM_WRITE_A[1])
                            0: begin
                                RAM[MEM_WRITE_A[13:2]][15:0] <= MEM_WRITE_DATA[15:0];
                                $display("%d@%h: *%h <= %h",$time,PC,{MEM_WRITE_A[31:2],2'b0},
{RAM[MEM_WRITE_A[13:2]][31:16],MEM_WRITE_DATA[15:0]});
```

```
                            end
                            1: begin
                                RAM[MEM_WRITE_A[13:2]][31:16] <= MEM_WRITE_DATA[15:0];
                                $display("%d@%h: *%h <= %h",$time,PC,{MEM_WRITE_A[31:2],2'b0},
{MEM_WRITE_DATA[15:0],RAM[MEM_WRITE_A[13:2]][15:0]});
                            end
                        endcase
                    end
                endcase
            end
        end
    end
```

题目描述

$Verilog$和C语言一样存在代码块，与C语言以`{`作为开头，以`}`作为结束不同的是，它的代码块是由`begin`作为开头，`end`作为结束，可以进行代码块的嵌套。现在给你一段$Verilog$代码，请你判断它是否符合$Verilog$的代码块规则。规则如下：

1、每个`begin`均有一个`end`与之对应，且`end`一定在`begin`的后面。

2、至少出现一次`begin`和`end`。

注意：**区分大小写。**

为简化题目，每组数据的所有代码均被压缩在一行中且没有空格。

输入格式

多组数据输入，保证数据组数小于20

每行为一个长度小于1000的字符串，保证字符串中只包含大小写字母和数字。

输出格式

对于每组数据，输出一行字符串。

如果符合规则，输出`"yes"`（不包含双引号），否则，输出`"no"`（不包含双引号）。

输入样例

```
beginen
beginAbeginDendEendF
```

输出样例

```
no
yes
```

Problem J. 异或

题目描述

注意：本题中的$\bigoplus$均表示位运算中的异或运算。

输入对于任意两个非负整数m, n可构造以下数列：

$m \bigoplus 0, m \bigoplus 1, m \bigoplus 2, \ldots, m \bigoplus n$

求在以上数列中未出现的最小非负整数。

输入描述

注意本题有多组数据，对于每组数据：

一行，两个非负整数m, n，$0 \leq m \leq 10^9$ 且 $0 \leq n \leq 10^9$。

输出描述

对于每组数据输出一行一个整数表示 $m \bigoplus 0, m \bigoplus 1, m \bigoplus 2, \ldots, m \bigoplus n$ 中未出现的最小非负整数。

样例输入

```
3 5
1 5
```

样例输出

```
4
6
```

样例解释

$3 \bigoplus 0 = 3 \quad 3 \bigoplus 1 = 2 \quad 3 \bigoplus 2 = 1 \quad 3 \bigoplus 3 = 0 \quad 3 \bigoplus 4 = 7 \quad 3 \bigoplus 5 = 6$

$\{3, 2, 1, 0, 7, 6\}$中未出现的最小自然数为4。

$1 \bigoplus 0 = 1 \quad 1 \bigoplus 1 = 0 \quad 1 \bigoplus 2 = 1 \quad 1 \bigoplus 3 = 2 \quad 1 \bigoplus 4 = 5 \quad 1 \bigoplus 5 = 4$

$\{3, 2, 1, 0, 7, 6\}$中未出现的最小自然数为4。

数据范围

对于10%的数据，$0 \leq m \leq 10^3$ 且 $0 \leq n \leq 10^3$；

对于30%的数据，$0 \leq m \leq 10^5$ 且 $0 \leq n \leq 10^5$；

对于40%的数据，$0 \leq m \leq 10^5$ 且 $0 \leq n \leq 10^9$；

对于100%的数据，$0 \leq m \leq 10^9$ 且 $0 \leq n \leq 10^9$；

对于100%的数据，数据组数不超过1000组 。

Hint

AUTHOR YUKILSY