以下是我从网上收集的关于组合博弈的资料汇总：

有一种很有意思的游戏，就是有物体若干堆，可以是火柴棍或是围棋子等等均可。两个  
人轮流从堆中取物体若干，规定最后取光物体者取胜。这是我国民间很古老的一个游戏  
，别看这游戏极其简单，却蕴含着深刻的数学原理。下面我们来分析一下要如何才能够  
取胜。

（一）**巴什博奕（Bash Game）：**只有一堆n个物品，两个人轮流从这堆物品中取物，规  
定每次至少取一个，最多取m个。最后取光者得胜。

    显然，如果n=m+1，那么由于一次最多只能取m个，所以，无论先取者拿走多少个，  
后取者都能够一次拿走剩余的物品，后者取胜。因此我们发现了如何取胜的法则：如果  
n=（m+1）r+s，（r为任意自然数，s≤m),那么先取者要拿走s个物品，如果后取者拿走  
k（≤m)个，那么先取者再拿走m+1-k个，结果剩下（m+1）（r-1）个，以后保持这样的  
取法，那么先取者肯定获胜。总之，要保持给对手留下（m+1）的倍数，就能最后获胜。  
    这个游戏还可以有一种变相的玩法：两个人轮流报数，每次至少报一个，最多报十  
个，谁能报到100者胜。  
（二）**威佐夫博奕（Wythoff Game）：**有两堆各若干个物品，两个人轮流从某一堆或同  
时从两堆中取同样多的物品，规定每次至少取一个，多者不限，最后取光者得胜。  
  
    这种情况下是颇为复杂的。我们用（ak，bk）（ak ≤ bk ,k=0，1，2，…,n)表示  
两堆物品的数量并称其为**局势**，如果甲面对（0，0），那么甲已经输了，这种局势我们  
称为**奇异局势**。前几个奇异局势是：（0，0）、（1，2）、（3，5）、（4，7）、（6，  
10）、（8，13）、（9，15）、（11，18）、（12，20）。

    可以看出,a0=b0=0,ak是未在前面出现过的最小自然数,而 bk= ak + k，奇异局势有  
如下三条性质：

    1。任何自然数都包含在一个且仅有一个奇异局势中。  
    由于ak是未在前面出现过的最小自然数，所以有ak > ak-1 ，而 bk= ak + k > ak  
-1 + k-1 = bk-1 > ak-1 。所以性质1。成立。  
    2。任意操作都可将奇异局势变为非奇异局势。  
    事实上，若只改变奇异局势（ak，bk）的某一个分量，那么另一个分量不可能在其  
他奇异局势中，所以必然是非奇异局势。如果使（ak，bk）的两个分量同时减少，则由  
于其差不变，且不可能是其他奇异局势的差，因此也是非奇异局势。  
    3。采用适当的方法，可以将非奇异局势变为奇异局势。

    假设面对的局势是（a,b），若 b = a，则同时从两堆中取走 a 个物体，就变为了  
奇异局势（0，0）；如果a = ak ，b > bk，那么，取走b  – bk个物体，即变为奇异局  
势；如果 a = ak ，  b < bk ,则同时从两堆中拿走 ak – ab – ak个物体,变为奇异局  
势（ ab – ak , ab – ak+ b – ak）；如果a > ak ，b= ak + k,则从第一堆中拿走多余  
的数量a – ak 即可；如果a < ak ，b= ak + k,分两种情况，第一种，a=aj （j < k）  
,从第二堆里面拿走 b – bj 即可；第二种，a=bj （j < k）,从第二堆里面拿走 b – a  
j 即可。

    从如上性质可知，两个人如果都采用正确操作，那么面对非奇异局势，先拿者必胜  
；反之，则后拿者取胜。

    那么任给一个局势（a，b），怎样判断它是不是奇异局势呢？我们有如下公式：

    ak =[k（1+√5）/2]，bk= ak + k  （k=0，1，2，…,n 方括号表示取整函数)  
  
奇妙的是其中出现了黄金分割数（1+√5）/2 = 1。618…,因此,由ak，bk组成的矩形近  
似为黄金矩形，由于2/（1+√5）=（√5-1）/2，可以先求出j=[a（√5-1）/2]，若a=[  
j（1+√5）/2]，那么a = aj，bj = aj + j，若不等于，那么a = aj+1，bj+1 = aj+1  
+ j + 1，若都不是，那么就不是奇异局势。然后再按照上述法则进行，一定会遇到奇异  
局势。

（三）**尼姆博奕（Nimm Game）：**有三堆各若干个物品，两个人轮流从某一堆取任意多的  
物品，规定每次至少取一个，多者不限，最后取光者得胜。

    这种情况最有意思，它与二进制有密切关系，我们用（a，b，c）表示某种局势，首  
先（0，0，0）显然是奇异局势，无论谁面对奇异局势，都必然失败。第二种奇异局势是  
（0，n，n），只要与对手拿走一样多的物品，最后都将导致（0，0，0）。仔细分析一  
下，（1，2，3）也是奇异局势，无论对手如何拿，接下来都可以变为（0，n，n）的情  
形。

    计算机算法里面有一种叫做按位模2加，也叫做异或的运算，我们用符号（+）表示  
这种运算。这种运算和一般加法不同的一点是1+1=0。先看（1，2，3）的按位模2加的结  
果：

1 =二进制01  
2 =二进制10  
3 =二进制11 （+）  
———————  
0 =二进制00 （注意不进位）

    对于奇异局势（0，n，n）也一样，结果也是0。

    任何奇异局势（a，b，c）都有a（+）b（+）c =0。

如果我们面对的是一个非奇异局势（a，b，c），要如何变为奇异局势呢？假设 a < b  
< c,我们只要将 c 变为 a（+）b,即可,因为有如下的运算结果: a（+）b（+）(a（+）  
b)=(a（+）a)（+）(b（+）b)=0（+）0=0。要将c 变为a（+）b，只要从 c中减去 c-（  
a（+）b）即可。  
  
    例1。（14，21，39），14（+）21=27，39-27=12，所以从39中拿走12个物体即可达  
到奇异局势（14，21，27）。

    例2。（55，81，121），55（+）81=102，121-102=19，所以从121中拿走19个物品  
就形成了奇异局势（55，81，102）。

    例3。（29，45，58），29（+）45=48，58-48=10，从58中拿走10个，变为（29，4  
5，48）。

    例4。我们来实际进行一盘比赛看看：  
        甲:(7,8,9)->(1,8,9)奇异局势  
        乙:(1,8,9)->(1,8,4)  
        甲:(1,8,4)->(1,5,4)奇异局势  
        乙:(1,5,4)->(1,4,4)  
        甲:(1,4,4)->(0,4,4)奇异局势  
        乙:(0,4,4)->(0,4,2)  
        甲:(0.4,2)->(0,2,2)奇异局势  
        乙:(0,2,2)->(0,2,1)  
        甲:(0,2,1)->(0,1,1)奇异局势  
        乙:(0,1,1)->(0,1,0)  
        甲:(0,1,0)->(0,0,0)奇异局势  
        甲胜。

取火柴的游戏  
题目1：今有若干堆火柴，两人依次从中拿取，规定每次只能从一堆中取若干根，   
可将一堆全取走，但不可不取，最后取完者为胜，求必胜的方法。   
题目2：今有若干堆火柴，两人依次从中拿取，规定每次只能从一堆中取若干根，   
可将一堆全取走，但不可不取，最后取完者为负，求必胜的方法。  
嘿嘿，这个游戏我早就见识过了。小时候用珠算玩这个游戏：第一档拨一个，第二档拨两个，依次直到第五档拨五个。然后两个人就轮流再把棋子拨下来，谁要是最后一个拨谁就赢。有一次暑假看见两个小孩子在玩这个游戏，我就在想有没有一个定论呢。下面就来试着证明一下吧  
先解决第一个问题吧。  
定义：若所有火柴数异或为0，则该状态被称为利他态，用字母T表示；否则，   
为利己态，用S表示。  
[定理1]：对于任何一个S态，总能从一堆火柴中取出若干个使之成为T态。  
证明：  
    若有n堆火柴，每堆火柴有A(i)根火柴数，那么既然现在处于S态，  
      c = A(1) xor A(2) xor … xor A(n) > 0;  
    把c表示成二进制，记它的二进制数的最高位为第p位，则必然存在一个A(t),它二进制的第p位也是1。（否则，若所有的A(i)的第p位都是0，这与c的第p位就也为0矛盾）。  
    那么我们把x = A(t) xor c,则得到x < A(t).这是因为既然A(t)的第p位与c的第p位同为1,那么x的第p位变为0,而高于p的位并没有改变。所以x < A(t).而  
    A(1) xor A(2) xor … xor x xor … xor A(n)  
  = A(1) xor A(2) xor … xor A(t) xor c xor … xor A(n)  
  = A(1) xor A(2) xor… xor A(n) xor A(1) xor A(2) xor … xor A(n)  
  = 0  
这就是说从A(t)堆中取出 A(t) – x 根火柴后状态就会从S态变为T态。证毕  
[定理2]：T态，取任何一堆的若干根，都将成为S态。  
证明：用反证法试试。  
      若  
      c = A(1) xor A(2) xor … xor A(i) xor … xor A(n) = 0；  
      c’ = A(1) xor A(2) xor … xor A(i’) xor c xor … xor A(n) = 0;  
      则有  
c xor c’ = A(1) xor A(2) xor … xor A(i) xor … xor A(n) xor A(1) xor A(2) xor … xor A(i’) xor c xor … xor A(n) = A(i) xor A(i’) =0  
      进而推出A(i) = A(i’)，这与已知矛盾。所以命题得证。  
[定理 3]：S态，只要方法正确，必赢。   
  最终胜利即由S态转变为T态，任何一个S态，只要把它变为T态，（由定理1，可以把它变成T态。）对方只能把T态转变为S态(定理2)。这样，所有S态向T态的转变都可以有己方控制，对方只能被动地实现由T态转变为S态。故S态必赢。  
[定理4]：T态，只要对方法正确，必败。   
  由定理3易得。   
接着来解决第二个问题。  
定义：若一堆中仅有1根火柴，则被称为孤单堆。若大于1根，则称为充裕堆。  
定义：T态中，若充裕堆的堆数大于等于2，则称为完全利他态，用T2表示；若充裕堆的堆数等于0，则称为部分利他态，用T0表示。  
   
孤单堆的根数异或只会影响二进制的最后一位，但充裕堆会影响高位（非最后一位）。一个充裕堆，高位必有一位不为0，则所有根数异或不为0。故不会是T态。  
[定理5]：S0态，即仅有奇数个孤单堆，必败。T0态必胜。   
证明：  
S0态，其实就是每次只能取一根。每次第奇数根都由己取，第偶数根都由对   
方取，所以最后一根必己取。败。同理,  T0态必胜#  
[定理6]：S1态，只要方法正确，必胜。   
证明：  
若此时孤单堆堆数为奇数，把充裕堆取完；否则，取成一根。这样，就变成奇数个孤单堆，由对方取。由定理5，对方必输。己必胜。  #   
[定理7]：S2态不可转一次变为T0态。   
证明：  
充裕堆数不可能一次由2变为0。得证。  #

[定理8]：S2态可一次转变为T2态。   
证明：  
由定理1，S态可转变为T态，态可一次转变为T态，又由定理6，S2态不可转一次变为T0态，所以转变的T态为T2态。  #   
[定理9]：T2态，只能转变为S2态或S1态。   
证明：  
由定理2，T态必然变为S态。由于充裕堆数不可能一次由2变为0，所以此时的S态不可能为S0态。命题得证。   
[定理10]：S2态，只要方法正确，必胜.   
证明：  
方法如下：   
      1）  S2态，就把它变为T2态。（由定理8）   
      2）  对方只能T2转变成S2态或S1态（定理9）  
    若转变为S2,  转向1）   
    若转变为S1,  这己必胜。（定理5）   
[定理11]：T2态必输。   
证明：同10。   
综上所述，必输态有：  T2,S0   
          必胜态：    S2,S1,T0.   
两题比较：   
第一题的全过程其实如下：   
S2->T2->S2->T2->  ……  ->T2->S1->T0->S0->T0->……->S0->T0(全0)   
第二题的全过程其实如下：   
S2->T2->S2->T2->  ……  ->T2->S1->S0->T0->S0->……->S0->T0(全0)   
下划线表示胜利一方的取法。  是否发现了他们的惊人相似之处。   
我们不难发现(见加黑部分)，S1态可以转变为S0态（第二题做法），也可以转变为   
T0（第一题做法）。哪一方控制了S1态，他即可以有办法使自己得到最后一根（转变为   
T0）,也可以使对方得到最后一根（转变为S0）。   
  所以，抢夺S1是制胜的关键！   
  为此，始终把T2态让给对方，将使对方处于被动状态，他早晚将把状态变为S1.

推荐HDOJ题目  
<http://acm.hdu.edu.cn/showproblem.php?pid=1907>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2509>  
看完上面的结论，就能顺利解决上面2道了

S-Nim  
<http://acm.hdu.edu.cn/showproblem.php?pid=1536>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1944>

博弈算法入门小节 1536 1517 1907  
小子最近迷途于博弈之中。。。感触颇深。  
为了让大家能够在学习博弈的时候少走弯路，最重要的也是为了加深自己的影响，温故而知新，特发此贴与大家共勉。  
学博弈先从概念开始：  
特别推荐LCY老师的课件：博弈入门。  
下载地址：<http://acm.hdu.edu.cn/forum/read.php?tid=6875>  
这个课件个人认为从博弈的基本思想，一直到解博弈的中心算法做了很好的诠释。但是特别要注意的是。课件后面一部分英语写的讲义是重中之重。小子英语很弱，在这困扰很久。现在为大家大概介绍一下。  
主要是后继点和SG值的问题:  
SG值：一个点的SG值就是一个不等于它的后继点的SG的且大于等于零的最小整数。  
后继点：也就是按照题目要求的走法（比如取石子可以取的数量，方法）能够走一步达到的那个点。  
具体的有关SG值是怎么运用的希望大家自己多想想。  
课件后面有一个1536的代码。可以放在后面做做  
看到这里推荐大家做几道题：1846（最简单的博弈水题）  
1847（求SG值）

有了上面的知识接下来我们来看看组合博弈（n堆石子）  
推荐大家看个资料：  
博弈-取石子游戏(推荐等级五星级)  
<http://acm.hdu.edu.cn/forum/read.php?fid=20&tid=5748>  
<http://hi.baidu.com/netnode/blog/item/30932c2edc7384514fc226ea.html>  
这里提出了一个奇异状态的问题。看了这篇文章你会发现异或运算在博弈中使用的妙处。当然这里指出的只是组合博弈中一种特殊情况。  
王道还是对SG值的求解，但是知道这么一种思路无疑对思维的广度和深度扩展是很有帮助的。  
ZZ博弈  
<http://acm.hdu.edu.cn/forum/read.php?fid=9&tid=10617>  
这里介绍了组和博弈的两种大的类型，一种是最后取的是N状态一种是最后取的是P状态，两个状态的解题方法能看懂很有帮助。当然，能够把推导过程理解，吃透无疑是大牛级的做法~小子也佩服的紧~     
    1536题推荐做做这题，这题前面提醒大家是一个求SG值的题目，题目前面是对异或运算运用在组合博弈问题中的很好的解释。当然题目本身是有所不同的。因为在这里面对取法有所要求。那么这样就回归到了解决博弈问题的王道算法——求SG值上。  
    有关运用求SG值的博弈题目有： 1850（也可基于奇异状态异或）  
1848（中和的大斐波那契数列的典型求SG值题）  
1517（个人认为有点猥琐的题目。。。。在此题上困扰很久。当然搞出来很开心。小子是用比较规矩的求SG值的方法求出来的，但是论坛有人对其推出来了规律，这里佩服一下，大家可以学习一下）  
1079（更猥琐的题目，对新手要求较高，因为按传统方法需要比较细致的模拟加对边角状态的考虑，同样有人推出来了公式）  
当你全部看完以上的东西。做完以上的题目的话。。。小子恭喜你~你博弈入门了~~~~  
    这里小子告诉大家。博弈很强大。学习要耐心~谢谢  
Current System Time : 2008-12-11 19:16:03

ACM课作业：  
1001 Brave Game  
1002 Good Luck in CET-4 Everybody!  
1003 Fibonacci again and again  
1004 Rabbit and Grass  
1005 Being a Good Boy in Spring Festival  
1006 Public Sale   
1007 悼念512汶川大地震遇难同胞——选拔志愿者   
1008 kiki’s game   
1009 Calendar Game   
1010 A Multiplication Game   
1011 Digital Deletions   
1012 S-Nim  
<http://acm.hdu.edu.cn/forum/read.php?tid=11339&fpage=0&toread=&page=1>

1536的参考代码  
本部分设定了隐藏,您已回复过了,以下是隐藏的内容  
Copy code  
//博弈-基于求SG值  
//Accepted 1536 578MS 416K 904 B  
#include”iostream”  
using namespace std;  
int f[101],sg[10001],k;  
int mex(int b)  
{  
    int a[101]={0},i;  
    for(i=0;i<k;i++)  
    {  
        if(b-f*<0)//b-f后继点  
            break;  
        if(sg[b-f]==-1)  
        {  
            sg[b-f]=mex(b-f);  
        }  
        a[sg[b-f]]=1;  
    }  
    for(i=0;i<k;i++)  
        if(!a)  
        {  
            return i;  
        }  
}  
int main()  
{  
    int i,t,n,s,bead,j;  
    while(cin >> k,k)  
    {  
        for(i=0;i<k;i++)  
        {  
            cin >> f;  
        }  
        memset(sg,-1,sizeof(sg));  
        for(i=0;i<k;i++)  
            for(j=i+1;j<k;j++)  
                if(f>f[j])  
                {  
                    f+=f[j];  
                    f[j]=f-f[j];  
                    f-=f[j];  
                }  
        sg[0]=0;  
        cin >> t;  
        while(t–)  
        {  
            cin >> n;  
            s=0;  
            while(n–)  
            {  
                cin >> bead;//该堆的成员个数  
                if(sg[bead]==-1)  
                    sg[bead]=mex(bead);  
                s=s^sg[bead];  
            }  
            if(s==0)  
                cout << “L”;  
            else  
                cout << “W”;  
        }  
        cout << endl;  
    }  
    return 0;  
}*

1517参考代码  
本部分设定了隐藏,您已回复过了,以下是隐藏的内容  
Copy code  
//博弈-基于求SG值  
//Accepted 1517 234MS 0K 837 B  
#include”iostream”  
using namespace std;  
int main()  
{  
    \_\_int64 a[7000]={1},min,n;  
    int p[10],sg[7000],i,j,k;  
    for(i=2;i<10;p*=0,i++);  
    for(i=1;i<7000;i++)  
    {  
        for(j=2,min=-1;j<10;j++)  
            if(min==-1||a[p[j]]\*j<a[p[min]]\*min)  
                min=j;  
        a=a[p[min]]\*min;  
        min=a[p[min]]\*min;  
        if(a>=5000000000)  
            break;  
        for(j=2;j<10;j++)  
            if(a[p[j]]\*j==min)  
                p[j]++;  
    }//从小到大求出所有乘积  
    while(scanf(“%I64d”,&n)!=EOF)  
    {  
        for(i=0;i<7000;i++)  
        {  
            sg=0;  
            if(a>=n)  
                break;  
        }  
        for(j=i-1;a[j]\*9>=n&&j>=0;j–)  
            sg[j]=1;  
        while(j>=0)  
        {  
            for(k=j+1;k<i&&a[j]\*9>=a[k];k++)  
                if(a[k]%a[j]==0&&sg[k]==0)  
                {  
                    sg[j]=1;  
                    break;  
                }  
            j–;  
        }  
        puts(sg[0]?”Stan wins.”:”Ollie wins.”);  
    }  
    return 0;  
}  
这里感谢shǎ崽同学的一段代码让小子学会了puts的妙用*

1907参考代码  
本部分设定了隐藏,您已回复过了,以下是隐藏的内容  
#include”iostream”  
using namespace std;  
int main()  
{  
    int temp,t,n,s,x,i;  
    cin >> t;  
    while(t–)  
    {  
        cin >> n;  
        for(i=s=temp=0;i<n;i++)  
        {  
            cin >> x;  
            if(x>1)    temp=1;  
            s^=x;  
        }  
        if((s&&temp)||(!s&&!temp))  
            cout << “John” << endl;  
        else  
            cout << “Brother” << endl;  
    }  
    return 0;  
}