

[报告]C-John

[Source]

<http://202.114.18.202:8080/judge/problem/viewProblem.action?id=15083>

[Description]

N 堆石子，两个玩家轮流取，每次可取某一堆中任意数量的石子，不可不取，取到最后一粒石子的为败者

[Solution]

本题是在经典的 NIM 取石子游戏上做了变化，修改了胜利的条件，即由经典的取完石子者胜变为取完石子者败。在我们解决经典 NIM 问题时，我们将初始状态每堆石子的数量放在一起做异或运算，若结果大于 0，则认为当前玩家（即先手）处于必胜态。

分析经典 NIM 的取胜原理，处于必胜态的玩家总能通过合理的操作使自己处于只有一个富裕堆的状态，即只有一个堆石子数大于 1，其余堆石子数为孤单堆（只有一颗石子），他可以取走整堆石子或使该堆剩下一枚石子从而使局势变为只含**奇数**个孤单堆的情况，从而保证自己取得胜利。

对于本题取完石子者败的情况，处于异或值大于 0 状态的玩家同样可以控制含一个富裕堆与若干个孤单堆的状态，他可以取走整堆石子或使该堆剩下一枚石子从而使局势变为只含**偶数**个孤单堆的情况，从而保证自己取得胜利。所以初态异或为 0 则先手必败，初态异或为 1 则先手必胜，这里的**特殊情况**是：若初态中全为孤单堆，则 N 为偶数时先手必胜，N 为奇数是先手必败。

[Code]

```
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
using namespace std;
int n,win,x,t,one;
int main()
{
    scanf("%d",&t);
    while (t--)
    {
        scanf("%d",&n);
        one=1;        //用于标记初态是否全为 1
        win=0;
        for (int i=1; i<=n; i++)
        {
            scanf("%d",&x);
            if (x>1) one=0;
            win=win^x;
        }
    }
}
```

```
    if (one)
    {
        if (n%2==0) printf("John\n");
        else printf("Brother\n");
    }
    else if (!win)
        printf("Brother\n");
    else
        printf("John\n");
}
return 0;
}
```