巴什博弈：从一堆石子中拿石子，一次拿1~m个。

威佐夫博弈：从两堆石子中拿石子，方法①任选一堆石子拿k个石子（k≥1），方法②从两堆石子中拿相同数量的石子（当然所拿的数要≥1）

尼姆博弈：从三堆石子中拿石子，每次任选一堆拿任意数目（≥1）的石子。这些当然是谁先那光，谁获胜。

# 2509 Be the Winner

**Problem Description**

Let's consider m apples divided into n groups. Each group contains no more than 100 apples, arranged in a line. You can take any number of consecutive（连续的） apples at one time.  
For example "@@@" can be turned into "@@" or "@" or "@ @"(two piles). two people get apples one after another and the one who takes the last is the loser. Fra wants to know in which situations he can win by playing strategies（策略） (that is, no matter what action the rival（竞争对手） takes, fra will win).

**Input**

You will be given several cases. Each test case begins with a single number n (1 <= n <= 100), followed by a line with n numbers, the number of apples in each pile（一堆）. There is a blank line between cases.

**Output**

If a winning strategies can be found, print a single line with "Yes", otherwise print "No".

**Sample Input**

2

2 2

1

3

**Sample Output**

No

Yes

1. #include <stdio.h>
2. #include <string.h>
3. #include <algorithm>
4. **using** **namespace** std;
6. **int** main()
7. {
8. **int** n,i;
9. **int** sum1,sum2,ans;
10. **int** a[55];
11. **while**(~scanf("%d",&n))
12. {
13. sum1 = sum2 = ans = 0;
14. **for**(i = 0; i<n; i++)
15. {
16. scanf("%d",&a[i]);
17. ans^=a[i];
18. **if**(a[i]>=2)
19. sum2++;
20. **else**
21. sum1++;
22. }
23. **if**((ans && sum2) || (!ans && !sum2))
24. printf("Yes\n");
25. **if**((ans && sum1%2 && !sum2) || (!ans && sum2>=2))
26. printf("No\n");
27. }
29. **return** 0;
30. }

# 1907 John

**Problem Description**

Little John is playing very funny game with his younger brother. There is one big box filled with M&Ms of different colors. At first John has to eat several M&Ms of the same color. Then his opponent（对手） has to make a turn. And so on. Please note that each player has to eat at least one M&M during his turn. If John (or his brother) will eat the last M&M from the box he will be considered as a looser and he will have to buy a new candy box.  
Both of players are using optimal game strategy（最优博弈策略）. John starts first always. You will be given information about M&Ms and your task is to determine a winner of such a beautiful game.

**Input**

The first line of input will contain a single integer T – the number of test cases. Next T pairs of lines will describe tests in a following format（格式）. The first line of each test will contain an integer N – the amount of different M&M colors in a box. Next line will contain N integers Ai, separated by spaces – amount of M&Ms of i-th color.  
Constraints（约束，限制）:  
1 <= T <= 474,  
1 <= N <= 47,  
1 <= Ai <= 4747

**Output**

Output T lines each of them containing information about game winner. Print “John” if John will win the game or “Brother” in other case.

**Sample Input**

2

3

3 5 1

1

1

**Sample Output**

John

Brother

这道题分为两种情况讨论：

①若所有堆的数量都为1。则根据奇偶来判断谁胜。

②其他情况，将所有数据异或起来，判断是否为奇异态。

#include <stdio.h>

#include <string.h>

#include <algorithm>

using namespace std;

int main()

{

int t,n,i;

int sum1,sum2,ans;

int a[55];

scanf("%d",&t);

while(t--)

{

scanf("%d",&n);

sum1 = sum2 = ans = 0;

for(i = 0;i<n;i++)

{

scanf("%d",&a[i]);

ans^=a[i];

if(a[i]>=2)

sum2++;

else

sum1++;

}

if((ans && sum2) || (!ans && !sum2))

printf("John\n");

if((ans && sum1%2 && !sum2) || (!ans && sum2>=2))

printf("Brother\n");

}

return 0;

}

# 1850 Being a Good Boy in Spring Festival

**Problem Description**  
  
下面是一个二人小游戏：桌子上有M堆扑克牌；每堆牌的数量分别为Ni(i=1…M)；两人轮流进行；每走一步可以任意选择一堆并取走其中的任意张牌；桌子上的扑克全部取光，则游戏结束；最后一次取牌的人为胜者。  
现在我们不想研究到底先手为胜还是为负，我只想问大家：  
——“先手的人如果想赢，第一步有几种选择呢？”

**Input**

输入数据包含多个测试用例，每个测试用例占2行，首先一行包含一个整数M(1<M<=100)，表示扑克牌的堆数，紧接着一行包含M个整数Ni(1<=Ni<=1000000，i=1…M)，分别表示M堆扑克的数量。M为0则表示输入数据的结束。

**Output**

如果先手的人能赢，请输出他第一步可行的方案数，否则请输出0，每个实例的输出占一行。

**Sample Input**

3

5 7 9

0

**Sample Output**

1

Nim博弈

题意：有m堆牌，两个人先后取某堆中的任意（不少于一）张牌，最后取完者胜；问先手取胜第一次取牌有多少种取法。

思路：1）如若给出的是必败状态：a1^a2^......^an=0,则先手不会有任何可能获得胜利；

        2）若给出的是必胜状态：a1^a2^.......^an=k,(其中k不为零)，那么我们的目的是要把必胜状态转化为必败状态从 而使得先手胜利。若a1^a2^...^an!=0，一定存在某个合法的移动，将ai改变成ai'后满足a1^a2^...^ai'^...^an=0。若a1^a2^...^an=k，则一定存在某个ai，它的二进制 表示在k的最高位上是1（否则k的最高位那个1是怎么得到的）。这时ai^k<ai一定成立。则我们可以将ai改变成ai'=ai^k，此a1^a2^...^ai'^...^an=a1^a2^...^an^k=0。

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int a[102],m,i,sum,s,count;

while(scanf("%d",&m)!=EOF&&m)

{

sum=count=0;

for(i=0;i<m;i++)

{

scanf("%d",&a[i]);

sum=sum^a[i];

}

for(i=0;i<m;i++)

{

s=sum^a[i];

if(s<a[i])

count++;

}

printf("%d\n",count);

}

return 0;

}