## 巴什博弈(Bash Game))

## 取牌游戏1

(1) 玩家：2人；

(2) 道具：23张扑克牌；

(3) 规则：

游戏双方轮流取牌；

每人每次仅限于取1张、2张或3张牌；

扑克牌取光，则游戏结束；

最后取牌的一方为胜者。

## kiki’s game

Recently kiki has nothing to do. While she is bored, an idea appears in his mind, she just playes the checkerboard game.The size of the chesserboard is n\*m.First of all, a coin is placed in the top right corner(1,m). Each time one people can move the coin into the left, the underneath or the left-underneath blank space.The person who can't make a move will lose the game. kiki plays it with ZZ.The game always starts with kiki. If both play perfectly, who will win the game?

Input

Input contains multiple test cases. Each line contains two integer n, m (0<n,m<=2000). The input is terminated when n=0 and m=0.

Output

If kiki wins the game printf "Wonderful!", else "What a pity!".

Sample Input

5 3

5 4

6 6

0 0

Sample Output

What a pity!

Wonderful!

Wonderful!

## 取牌游戏2

现有n张扑克牌，两个人轮流取牌，每个人只能取1、3、7、8张牌，取到最后一张者获胜。请判断先取者是胜还是败。

分析：模拟得到规律

## 取石子游戏

两名参与者交替从一堆石子中取出若干数目,其个数由参与者自已决定.但是要求参与者每次至少取出一个,至多取出一半,然后另一名参与者继续.拿到最后一个石子的参与者将输掉该游戏.

## Ferguson游戏

有两个盒子，一开始其中一个有m颗糖，另一个有n颗糖，把这样的状态记为(n, m)。每次移动是将一个盒子清空，而把另一个盒子的一些糖拿到被清空的盒子中，使得两个盒子至少各有一颗糖，显然唯一的终止状态为(1,1)。如果最后移动的游戏者获胜，那么状态(m, n)的先手是胜还是败。

## Chomp游戏

有一个m\*n棋盘，每次可以取走一个方格并取走它上边和右边的所有方格，拿到左下角的格子(1, 1)者输。问先手必胜还是必败。

引申：约数游戏。有1~n个共n个数字，两个人轮流选择一个数，并把它和它的所有约数擦去，擦去最后一个数的人赢，问谁会获胜。

## Nim游戏

Nim游戏是博弈论中最经典的模型之一，它又有着十分简单的规则和无比优美的结论，Nim游戏是组合游戏(Combinatorial Games)的一种，准确来说，属于“Impartial Combinatorial Games”

有两个玩家；

有三堆扑克牌（比如：分别是 5，7，9张）；

游戏双方轮流操作；

玩家的每次操作是选择其中某一堆牌，然后从中取走任意张；

最后一次取牌的一方为获胜方；

## Be the Winner

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 apples at one time.

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 (that is, no matter what action the rival 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. 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

## Wythoff's Game

来源：POJ1067

有两堆石子，数量任意，可以不同。游戏开始由两个人轮流取石子。游戏规定，每次有两种不同的取法，一是可以在任意的一堆中取走任意多的石子；二是可以在两堆中同时取走相同数量的石子。最后把石子全部取完者为胜者。现在给出初始的两堆石子的数目，如果轮到你先取，假设双方都采取最好的策略，问最后你是胜者还是败者。

分析：

## 高僧斗法

问题描述

　　古时丧葬活动中经常请高僧做法事。仪式结束后，有时会有“高僧斗法”的趣味节目，以舒缓压抑的气氛。  
　　节目大略步骤为：先用粮食（一般是稻米）在地上“画”出若干级台阶（表示N级浮屠）。又有若干小和尚随机地“站”在某个台阶上。最高一级台阶必须站人，其它任意。(如图1所示)  
　　两位参加游戏的法师分别指挥某个小和尚向上走任意多级的台阶，但会被站在高级台阶上的小和尚阻挡，不能越过。两个小和尚也不能站在同一台阶，也不能向低级台阶移动。  
　　两法师轮流发出指令，最后所有小和尚必然会都挤在高段台阶，再也不能向上移动。轮到哪个法师指挥时无法继续移动，则游戏结束，该法师认输。  
　　对于已知的台阶数和小和尚的分布位置，请你计算先发指令的法师该如何决策才能保证胜出。

输入格式

　　输入数据为一行用空格分开的N个整数，表示小和尚的位置。台阶序号从1算起，所以最后一个小和尚的位置即是台阶的总数。（N<100, 台阶总数<1000）

输出格式

　　输出为一行用空格分开的两个整数: A B, 表示把A位置的小和尚移动到B位置。若有多个解，输出A值较小的解，若无解则输出-1。

样例输入

1 5 9

样例输出

1 4

样例输入

1 5 8 10

样例输出

1 3

分析：将小和尚的位置从小到大进行排序。然后从最底层的小和尚开始，两两分组(如果N为奇数，则忽略最高处的小和尚)，计算每组小和尚之间的间隔数b[i]，对b[i]采用nim-sum判断是否获胜。原理：如果nim-sum为非0，则第一次走时可以达到必败点，此时必定移动某对和尚中的下面的和尚，如果对手移动的是某对和尚的下面的和尚，则根据nim游戏结论，只要移动某对下面的和尚是nim-sum为0即可；如果对手移动的是某对上面的和尚，则只要将该对下面的和尚移动相同的台阶数即可，此时nim-sum还是0。

## 除法游戏

有一个n\*m(1<=n, m<=50)矩阵，每个元素均为2~10000之间的正整数。两个游戏者轮流操作。每次可以选一行中的1个或多个大于1的整数，把它们中的每个数都变成它的某个真因子，比如12可以变成1,2,3,4,6，不能操作的输(换句话说，如果在谁操作之前，矩阵中的所有数都是1，则他输)。

Input

The \_rst line has a positive integer T, T \_ 50, denoting the number of test cases. This is followed by

each test case per line.

Each test case starts with a line containing 2 integers N and M representing the number of rows

and columns respectively. Both N and M are between 1 and 50 inclusive. Each of the next N line each

contains M integers. All these integers are between 2 and 10000 inclusive.

Output

For each test case, the output contains a line in the format `Case #x: M', where x is the case number

(starting from 1) and M is `YES' when the \_rst player has a winning strategy and `NO' otherwise.

Sample Input

5

2 2

2 3

2 3

2 2

4 9

8 5

3 3

2 3 5

3 9 2

8 8 3

3 3

3 4 5

4 5 6

5 6 7

2 3

4 5 6

7 8 9

Sample Output

Case #1: NO

Case #2: NO

Case #3: NO

Case #4: YES

Case #5: YES

## Godsend

来源：Codeforces 841B

time limit per test 2 seconds

memory limit per test 256 megabytes

Leha somehow found an array consisting of *n* integers. Looking at it, he came up with a task. Two players play the game on the array. Players move one by one. The first player can choose for his move a subsegment of non-zero length with an odd sum of numbers and remove it from the array, after that the remaining parts are glued together into one array and the game continues. The second player can choose a subsegment of non-zero length with an even sum and remove it. Loses the one who can not make a move. Who will win if both play optimally?

**Input**

First line of input data contains single integer *n* (1 ≤ *n* ≤ 106) — length of the array.

Next line contains *n* integers *a*1, *a*2, ..., *an* (0 ≤ *ai* ≤ 109).

**Output**

Output answer in single line. "First", if first player wins, and "Second" otherwise (without quotes).

**Examples**

**input**

4  
1 3 2 3

**output**

First

**input**

2  
2 2

**output**

Second

**Note**

In first sample first player remove whole array in one move and win.

In second sample first player can't make a move and lose.

## 石头游戏

有n堆石子，每堆分别有a1, a2, …, an个石子，每次可以选1堆，拿走至少一个一个石子，但不能拿走超过一半石子。比如，若有3堆石子，每堆分别有5,1,2个石子，则在下一轮中，游戏者可以从第一堆中拿1个或2个，第二堆中不能拿，第三堆中只能拿1个。谁不能拿石子，就算输。

输入：

输入第一行为石子堆数n(1<=n<=100)，第二行包含n个整数a1, a2, …, an(1<=ai<=2\*1018)。

输出：

如果先手胜，则输出YES，否则输出NO。

## Treblecross游戏

来源：uva 10561

有n个格子排成一排，其中一些格子里面有字符X，两个游戏者轮流操作，每次可以选择一个空格，在里面放上字符X。如果此时有3个连续的X出现，则该游戏者赢得比赛，初始的情况不会出现3个连续的X。判断先手必胜还是必败，如果必胜，输出所有的必胜策略(第一次放置的位置)。

**Input**

The input begins with an integer *N* (*N <* 100), the number of states that will follow. Each state is represented by a string on a line by itself. The string will only contain the characters `.' and `X'. The length of the string (the size of the board) will be between 3 and 200 characters, inclusive. No state will contain three `X' in a row.

**Output**

For each case, \_rst output `WINNING' or `LOSING' depending on if the player to move will win or lose the game. On the next line, output in increasing order all positions on the board where the player to move may put an X and win the game. The positions should be separated by a blank, and be in increasing order. The leftmost position on the board is 1.

**Sample Input**

4

.....

X.....X..X.............X....X..X

.X.X...X

...............................................

**Sample Output**

WINNING

3

LOSING

WINNING

3

WINNING

1 12 15 17 20 24 28 31 33 36 47