

## Задача А. Жестокая задача

Имя входного файла: `cruel.in`  
Имя выходного файла: `cruel.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Штирлиц и Мюллер стреляют по очереди. В очереди  $n$  человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

### Формат входных данных

Входной файл содержит единственное число  $n$  ( $2 \leq n \leq 5\,000$ ) — количество человек в очереди.

### Формат выходных данных

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова `Mueller`. Иначе в первой строке необходимо вывести слово `Schtirlitz`, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

### Примеры

<code>cruel.in</code>	<code>cruel.out</code>
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

## Задача В. Ретроанализ для маленьких

Имя входного файла: `retro.in`  
Имя выходного файла: `retro.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан ориентированный весёлый граф из  $n$  вершин и  $m$  ребер. Оля и Коля играют в игру. Изначально фишка стоит в вершине  $i$ . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины  $i$  определить, кто выиграет при оптимальной игре обоих.

### Формат входных данных

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой два целых числа  $n$  ( $1 \leq n \leq 300\,000$ ) и  $m$  ( $1 \leq m \leq 300\,000$ ). Следующие  $m$  строк содержат ребра графа, каждое описывается парой целых чисел от 1 до  $n$ . Пара  $a\ b$  обозначает, что ребро ведет из вершины  $a$  в вершину  $b$ . В графе могут быть петли, могут быть кратные ребра. Сумма  $n$  по всем тестам не превосходит 300 000, сумма  $m$  по всем тестам также не превосходит 300 000.

### Формат выходных данных

Для каждого теста выведите для каждой вершины `FIRST`, `SECOND` или `DRAW` в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

### Примеры

<code>retro.in</code>	<code>retro.out</code>
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	FIRST
2 1	SECOND
1 2	FIRST
4 4	FIRST
1 2	SECOND
2 3	SECOND
3 1	
1 4	

## Задача С. Малыш и Карлсон

Имя входного файла: `karlsson.in`  
Имя выходного файла: `karlsson.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

На свой День рождения Малыш позвал своего лучшего друга Карлсона. Мама испекла его любимый пирог прямоугольной формы  $a \times b \times c$  сантиметров. Карлсон знает, что у Малыша еще есть килограмм колбасы. Чтобы заполучить ее, он предложил поиграть следующим образом: они по очереди разрезают пирог на две ненулевые по объему прямоугольные части с целыми измерениями и съедают меньшую часть (в случае, когда части равные, можно съесть любую). Проигрывает тот, кто не может сделать хода (то есть когда размеры будут  $1 \times 1 \times 1$ ). Естественно, победителю достается колбаса.

Малыш настаивает на том, чтобы он ходил вторым.

Помогите Карлсону выяснить, сможет ли он выиграть, и если сможет — какой должен быть его первый ход для этого.

Считается, что Малыш всегда ходит оптимально.

### Формат входных данных

Во входном файле содержится 3 целых числа  $a, b, c$  ( $1 \leq a, b, c \leq 5000$ ) — размеры пирога.

### Формат выходных данных

В случае, если Карлсон не сможет выиграть в Малыша, выведите NO. В противном случае в первой строке выведите YES, во второй — размеры пирога после первого хода Карлсона в том же порядке, что и во входном файле.

### Примеры

<code>karlsson.in</code>	<code>karlsson.out</code>
1 1 1	NO
1 2 1	YES 1 1 1
1 1 10	YES 1 1 7

## Задача D. Конфетки

Имя входного файла:	sweets.in
Имя выходного файла:	sweets.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

После разгромной победы Директора ЛКШ руководитель физической школы сдался и отпустил Деда Мороза к нам. Но ради интереса предложил сразиться ЛФШатам и ЛКШатам в еще одной непростой игре.

В каждой игре участвует один из вас и один ЛФШонок, а ходите вы по очереди. В кучку перед вами кладется  $N$  вкусных конфеток. На каждом ходе игрок может съесть от 1 до  $K$  конфеток (больше нельзя — много сладкого вредно даже в Новый Год), но при этом не больше, чем взял его противник на предыдущем ходе (не будем жадничать, мы же добрые). Второго ограничения нет лишь для первого хода каждой игры. Проигрывает тот, кому не осталось конфеток.

У нас возникли подозрения, что директор ЛФШ специально подобрал такие  $N$  и  $K$ , чтобы ЛКШата никогда не смогли выиграть. Мы надеемся, что это не так, и очень просим вас проверить это.

### Формат входных данных

Во входном файле записаны через пробел два целых числа —  $N$  ( $1 \leq N \leq 500$ ) и  $K$  ( $1 \leq K \leq 100$ ).

### Формат выходных данных

В выходной файл выведите минимальное число конфет, которое должен съесть ЛКШонок первым ходом, чтобы выиграть при оптимальной игре ЛФШонка, либо 0, если даже самый умный из нас не сможет одолеть идеального играющего противника.

### Примеры

sweets.in	sweets.out
7 3	1

## Задача Е. Огромный ним

Имя входного файла: `big-nim.in`  
Имя выходного файла: `big-nim.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Петя и Вася играют в ним, но не простой, а просто огромный. У них есть очень много кучек камней. Кучки разделены на  $n$  групп. Группа  $i$  состоит из кучек размеров от  $l_i$  до  $r_i$  включительно. Помогите ребятам понять, кто выиграет при оптимальной игре

### Формат входных данных

Первая строка входного файла содержит число  $n$  ( $1 \leq n \leq 10^5$ ), следующие  $n$  строк содержат пары чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq 10^{18}$ ).

### Формат выходных данных

Если первый игрок проигрывает, выведите `Lose`, если выигрывает — выведите в первой строке `Win`, а во второй строке — любой выигрышный ход для первого игрока. Ход задается размером кучки до хода и после него.

### Примеры

<code>big-nim.in</code>	<code>big-nim.out</code>
1	Win
1 10	8 3
2	Lose
2 5	
2 5	

## Задача F. Странная игра

Имя входного файла: `strange-game.in`  
Имя выходного файла: `strange-game.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Двое играют в простую игру на доске  $n \times n$ . У первого игрока есть одна белая фишка, а у второго — одна чёрная. Игроки ходят по очереди, первым ходит первый игрок (белые).

Первый игрок имеет право двигать свою фишку на одну клетку в одном из четырёх основных направлений (влево, вправо, вверх, вниз). Второй игрок при своем ходе также выбирает одно из этих четырёх направлений, но может передвинуть свою фишку как на одну клетку в этом направлении, так и на две. Выигрывает тот, кто первым съедает фишку соперника.

Определите победителя и число ходов, требуемое для победы, при оптимальной игре сторон.

### Формат входных данных

Во входном файле даны пять чисел —  $n$  ( $2 \leq n \leq 20$ ), а также координаты белой и чёрной фишек. Фишки стоят в разных клетках доски.

### Формат выходных данных

Выведите `WHITE`  $x$ , если выигрывают белые, `BLACK`  $x$ , если выигрывают чёрные, `DRAW`, если игра закончится вничью. Здесь  $x$  — число ходов обеих сторон (полуходов) до момента окончания игры.

### Примеры

<code>strange-game.in</code>	<code>strange-game.out</code>
2 1 1 2 2	BLACK 2
2 2 2 1 2	WHITE 1
3 1 1 3 3	BLACK 6