

## Задача А. Одна кучка

Имя входного файла:            `heaps2.in`  
Имя выходного файла:        `heaps2.out`  
Ограничение по времени:    2 секунды  
Ограничение по памяти:      64 мегабайта

Два игрока играют в игру. На столе лежит кучка из  $N$  камней. Двое ходят по очереди. За ход можно взять  $a_1, a_2, \dots, a_k$  камней. Проигрывает тот, кто не может сделать ход. Определите победителя!

### Формат входных данных

В первой строке записано число  $k$ . Во второй строке  $k$  чисел —  $a_1, a_2, \dots, a_k$ . В третьей строке идет число  $m$  — количество различных  $N$ , для каждого из которых требуется определить победителя. В четвертой строке  $m$  чисел —  $N_1, N_2, \dots, N_m$ .

Ограничения:  $1 \leq k \leq 20, m \leq 10^4, 1 \leq N_i, a_i \leq 10^6$ .

### Формат выходных данных

Выведите  $m$  строк, в каждой ответ на вопрос “кто выиграет” — **First** или **Second**.

### Пример

heaps2.in	heaps2.out
3	First
1 2 3	First
8	First
1 2 3 4 5 6 7 8	Second
	First
	First
	First
	Second

## Задача В. Комфортабельная рассадка

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

В парке стоит скамейка на  $n$  мест. Изначально она пуста. Две команды играют в игру, по очереди *комфортабельно* сажая на скамейку по одному человеку. Человек садится *комфортабельно*, если на соседних местах ни справа, ни слева от него непосредственно никого нет. Проигрывает та команда, которая не может *комфортабельно* посадить очередного человека на своём ходу.

Дано  $n$  — количество мест на скамейке. Какая команда выигрывает при правильной игре?

### Формат входных данных

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 100$ ).

### Формат выходных данных

В первой строке выходного файла выведите “FIRST”, если при правильной игре выигрывает первая команда, и “SECOND”, если вторая.

### Примеры

<code>comfort.in</code>	<code>comfort.out</code>
2	FIRST
4	SECOND

## Задача С. 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 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 строке 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

retro.in	retro.out
5 5 1 2 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 2 2 3 3 1 1 4	?

## Задача D. Малыш и Карлсон

Имя входного файла: `karlsson.in`  
Имя выходного файла: `karlsson.out`  
Ограничение по времени: 1 секунда  
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  $a, b, c$  ( $1 \leq a, b, c \leq 5\,000$ ) — размеры пирога.

### Формат выходных данных

В случае, если Карлсон не сможет выиграть у 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
2 1 1	YES 1 1 1

## Задача Е. Терминатор

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

Два игрока играют в настольную игру. Игровое поле представляет собой квадратный лабиринт,  $8 \times 8$  клеток. В некоторых клетках располагаются стенки. Один игрок управляет фишкой-терминатором, а второй — фишкой-беглецом. Игроки ходят по очереди, ходы пропускать нельзя (гарантируется, что ход всегда возможен). За один ход игрок может переместить свою фишку в любую из свободных клеток, расположенных рядом с исходной по горизонтали, вертикали или по диагонали (то есть ходом короля). Терминатор, кроме того, может стрелять в беглеца ракетами. Выстрел идет по прямой в любом направлении по горизонтали, вертикали или диагонали. Если беглец оказывается на линии выстрела терминатора и не прикрыт стенками, то терминатор незамедлительно делает выстрел (вне зависимости от того, чей ход), и беглец проигрывает. Начальное положение фишек задано. Первый ход делает беглец. Он выигрывает, если сделает ход с восьмой строки за пределы игрового поля, так как остальные границы поля окружены стенками.

Вопрос задачи: может ли беглец выиграть при оптимальной игре обеих сторон?

### Формат входных данных

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — цифрой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

### Формат выходных данных

В выходной файл выведите число 1, если беглец выигрывает, и  $-1$  — в противном случае.

### Примеры

<code>terminator.in</code>	<code>terminator.out</code>
01000000 10100000 31100000 00020000 00000000 00000000 00000000 00000000	-1