

Задача А. Вычти подстроку

Имя входного файла: `substring.in`
 Имя выходного файла: `substring.out`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

На доске написано натуральное число n . Аким Сергеевич и Маша из A' делают ходы по очереди.

Каждым ходом игрок выбирает натуральное число m , являющееся собственной подстрокой числа, написанного в данный момент на доске, и из числа на доске вычитается m .

Например, если на доске написано 2309, игрок может выбрать $m = 2, 3, 9, 23, 30, 230$ или 309. Таким образом, после этого хода на доске окажется одно из чисел 2000, 2079, 2279, 2286, 2300, 2306 и 2307.

Игрок, который не может сделать ход, проигрывает.

Первым ходит Аким Сергеевич. Помогите ему обыграть Машу! Найдите минимальное число m , которое ему следует вычесть своим первым ходом, чтобы после этого выиграть игру (при оптимальной игре Маши).

Формат входного файла

Во входном файле число n ($1 \leq n \leq 1000\,000$).

Формат выходного файла

Выведите минимальное m , которое следует вычесть Аким Сергеевичу, чтобы выиграть. Если Аким Сергеевич проигрывает при оптимальной игре Маши, выведите -1 .

Пример

substring.in	substring.out
5	-1
10	1
17	-1
239	9
566	66

Задача В. Две кучки

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

Два игрока играют в игру. На столе лежат две кучки: в первой a камней, во второй — b . Игроки ходят по очереди. Каждый ход игрок выбирает одну кучку и берет какое-то количество камней из нее. Первый игрок может брать a_1, a_2, \dots, a_k камней, второй — b_1, b_2, \dots, b_l . Проигрывает тот, кто не может сделать ход. Определите победителя!

Формат входного файла

В первой строке записаны a и b . Во второй строке записаны k и последовательность a_i , на третьей — l и b_i . $1 \leq a, b \leq 1\,000$, $1 \leq k, l \leq 10$, $1 \leq a_i, b_j \leq 1\,000$.

Формат выходного файла

Если выигрывает первый игрок, выведите `First`. Иначе выведите `Second`.

Пример

heaps.in	heaps.out
2 2 2 1 2 1 1	First
2 2 1 1 2 1 2	Second

Задача С. 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. Каждый 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 входного файла

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

Формат выходного файла

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.

Пример

cruel.in	cruel.out
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

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 започучить 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 входного файла

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

Формат выходного файла

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

karlsson.in	karlsson.out
1 1 1	NO
2 1 1	YES 1 1 1

Задача E. 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, 8×8 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 входного файла

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — циф-

рой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

Формат выходного файла

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

Примеры

terminator.in	terminator.out
01000000	-1
10100000	
31100000	
00020000	
00000000	
00000000	
00000000	
00000000	