

Задача А. Камни

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

На столе лежат N камней. Играют двое, ходят по очереди. За ход игрок может взять:

- 1 или 2 камня, если текущее количество камней на столе делится на 3;
- 1 или 3 камня, если оно дает остаток 1;
- 1, 2 или 3 камня, если оно дает остаток 2.

Каждый ход можно сделать только при наличии достаточного количества камней. Проигрывает тот, кто хода сделать не может.

Формат входных данных

Во входном файле находится одно целое число N ($0 < N \leq 100$).

Формат выходных данных

Выведите в выходной файл одно число 1 или 2 — номер игрока, который выигрывает при правильной игре.

Пример

<code>stones.in</code>	<code>stones.out</code>
3	2

Задача В. Вычти подстроку

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

На доске написано натуральное число n . Аким Сергеевич и Маша 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 входных данных

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

Формат выходных данных

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

Пример

<code>substr.in</code>	<code>substr.out</code>
5	-1
10	1
17	-1
239	9
566	66

Задача С. 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. Каждым ходом 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 входных данных

В первой строке записаны a и b . Во второй строке записаны k и последовательность a_i , на третьей — l и b_i . $1 \leq a, b \leq 1000$, $1 \leq k, l \leq 10$, $1 \leq a_i, b_j \leq 1000$.

Формат выходных данных

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

Пример

<code>heaps.in</code>	<code>heaps.out</code>
2 2 2 1 2 1 1	<code>First</code>
2 2 1 1 2 1 2	<code>Second</code>

Задача D. Ферзя в угол

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

В левом нижнем углу доски $M \times N$ стоит ферзь. Двое игроков по очереди ходят ферзем, перемещая его на любое число клеток по вертикали вверх, по горизонтали вправо, или по диагонали вправо-вверх. Выигрывает тот, кто поставит ферзя в правый верхний угол доски. Определите, какой из игроков имеет выигрышную стратегию.

Формат входных данных

На вход программе подается два натуральных числа M и N , не превосходящих 100.

Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

Пример

<code>queen.in</code>	<code>queen.out</code>
3 4	1

Задача Е. Терминатор

Имя входного файла: `terminat.in`
Имя выходного файла: `terminat.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 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>terminat.in</code>	<code>terminat.out</code>
01000000 10100000 31100000 00020000 00000000 00000000 00000000 00000000 00000000	-1