

## Задача А. Два капитана

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

*Корабли без капитанов, капитан без корабля...*

Несколько приятелей решили поиграть в футбол. Известно, что не все они играют одинаково хорошо, и у каждого есть сила, которую можно выразить некоторым числом (чем больше число, тем сильнее игрок). Поэтому было решено выделить двух капитанов, которые будут набирать людей в команды. Сначала решили, что капитаны будут по очереди выбирать по одному человеку из группы нераспределенных игроков. При этом изначально капитаны тянут жребий, и тот, кто его выигрывает, выбирает первый. Однако получается, что команда выигравшего жребий капитана получается гораздо сильнее другой команды, так как в каждой паре выбранных игроков игрок первой команды окажется сильнее (по крайней мере, не слабее) игрока второй команды. Поэтому было решено поступить более справедливо. Капитаны также по очереди выбирают игроков, но при выборе первой пары (и каждой нечётной) выбирать начинает выигравший жребий капитан, а при выборе второй (и далее каждой чётной) — начинает другой, проигравший жребий, капитан. Однако в таком случае может получиться, что при оптимальном выборе игроков капитанами команда проигравшего жребий капитана окажется сильнее команды выигравшего жребий капитана. Вам необходимо определить, что выгодно — выиграть или проиграть жребий, чтобы набрать более сильную команду, при условии, что оба капитана будут действовать оптимально, то есть выберут стратегию, которая позволит им набрать настолько сильную команду, насколько это возможно, независимо от выбора капитана команды соперника.

### Формат входного файла

В первой строке входного файла дано количество игроков  $n$  ( $0 < n \leq 100$ ,  $n$  — чётное). Во второй строке перечислены  $n$  чисел  $a_i$  — силы игроков ( $a_i \geq 0$ ). Капитанов среди них нет. Силы капитанов равны нулю. Все числа во входном файле целые и их сумма не превышает  $10^9$ .

### Формат выходного файла

В первой строке выходного файла выведите, получится ли команда выигравшего жребий сильнее (YES — если получится, и NO — если не получится). Во второй строке выведите через пробел, какая сила команды получится у выигравшего жребий капитана, и какая сила команды получится у проигравшего жребий капитана. Сила команды — это сумма сил всех игроков этой команды.

### Примеры

<code>captains.in</code>	<code>captains.out</code>
4 10 7 6 1	NO 11 13
2 100 99	YES 100 99

## Задача В. Игра

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

Однажды два математика придумали математическую игру и решили в неё сыграть. Они попросили постороннего человека написать на разных листах бумаги два натуральных числа, причём одно должно быть больше другого ровно в 2 раза. Затем один математик взял один лист, а второй взял оставшийся. В чужие листы они не заглядывали и единственное, что знал каждый из них — на другом листе записано число либо в 2 раза больше, либо в 2 раза меньше. Затем началась игра. Глядя на свой лист, первый игрок сказал, знает ли он, какое число записано на листе другого игрока. Затем на аналогичный вопрос ответил второй игрок, затем снова первый и так далее по очереди. При этом игроки всегда учитывали всю информацию, которую им давал ответ соперника, и говорили только правду. Например, могло быть так: первому игроку достался лист с числом 1, а второму — с числом 2. Тогда игра развивалась бы следующим образом:

*1 игрок: Я знаю, твоё число — 2.*

А вот другая ситуация: первому игроку достался лист с числом 8, а второму — с числом 16.

*1 игрок: Я не знаю, какое число записано у тебя на листе.*

*2 игрок: Я не знаю, какое число записано у тебя на листе.*

*1 игрок: Я не знаю, какое число записано у тебя на листе.*

*2 игрок: Я не знаю, какое число записано у тебя на листе.*

*1 игрок: Я знаю, твоё число — 16.*

Ваша задача — определить, с какого хода игроки смогут угадать, какое число записано на листе у соперника. Если они так и не смогут догадаться — выведите 0.

### Формат входного файла

В первой строке входного файла через пробел записаны два целых числа  $a$  и  $b$  — числа первого и второго игроков соответственно. ( $0 < a, b \leq 10^6$ ).

### Формат выходного файла

В выходной файл выведите номер хода, на котором игрок сможет с уверенностью сказать, какое число записано на листе у соперника, либо 0, если оба игрока не смогут этого сделать при любом количестве ходов.

### Примеры

<code>game.in</code>	<code>game.out</code>
1 2	1
8 16	5

## Задача С. Футбольные парадоксы

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

После того, как сборная Бразилии стала Чемпионом Мира по футболу, она проиграла сборной Италии, которая, в свою очередь, проиграла сборной Болгарии, а та — сборной Люксембурга. Так что же, получается, что Люксембург сильнее Бразилии? Такие парадоксы происходят довольно часто, и удивляться этому не стоит, так как победа в матче не обладает свойством транзитивности, то есть описанное выше не означает, что при встрече сборных Бразилии и Люксембурга обязательно выиграет Люксембург. Однако это заинтересовало Петю, который решил проанализировать результаты всех матчей, которые он знает, и найти самую длинную серию игр, чтобы, если в какой-то игре серии сборная выигрывает, то в следующей игре серии эта же сборная проигрывала. При этом одна и та же сборная может встречаться в серии несколько раз. Заметьте, что игры в серии упорядочены по времени, то есть для любой игры серии, следующая за ней должна была проходить позже. Пете не важно, заканчивается ли серия игр той же командой, с которой начинается, или нет. Прежде всего, его волнует количество игр в серии.

### Формат входного файла

Во входном файле содержится отсортированная хронологически последовательность игр, то есть каждая следующая игра в этой последовательности была сыграна позже предыдущей. В первой строке входного файла записано целое число  $n$  — количество сыгранных игр ( $0 < n \leq 10000$ ). В каждой из следующих  $n$  строк содержится описание одной игры. Каждая игра описывается строкой из семи символов. Первые три символа — идентификатор выигравшей команды, четвёртый символ — тире, символы с пятого по седьмой — идентификатор проигравшей команды. Идентификатор команды всегда является трёхбуквенным, и состоит только из заглавных латинских букв. Количество различных идентификаторов команд во входном файле не превышает 200.

### Формат выходного файла

В первую строку выходного файла выведите максимальное количество матчей в искомой серии. Во вторую строку выведите цепочку команд, начиная от команды, победившей в последнем матче серии и заканчивая командой, проигравшей в первом матче серии. Если вариантов наиболее длинных цепочек несколько, выведите любую из них.

### Пример

paradox.in	paradox.out
5 FRA-ITA GER-ITA ITA-GER BRA-FRA RUS-ITA	3 RUS-ITA-GER-ITA

**Пояснение:** серия в примере состоит из матчей GER-ITA, ITA-GER, RUS-ITA.

## Задача D. Шахматные снайперы

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

В связи с обострением ситуации на первом и третьем фронтах шахматной доски  $n \times m$  главнокомандующим белой армии было решено ввести новую боевую единицу — шахматного снайпера — фигуру, которая бьет в каком-то определенном направлении (вверх, вниз, влево или вправо), которое фиксируется при установке его на клетку поля. Однако перед непосредственным вступлением снайперов в бой необходимо выяснить: какое максимальное количество таких фигур можно разместить на пустом поле  $n \times m$  так, чтобы они не били друг друга.

### Формат входного файла

В первой строке входного файла записаны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 1000$ ).

### Формат выходного файла

В первую строку выходного файла выведите одно число — максимальное количество не бьющих друг друга снайперов.

### Пример

<code>snipers.in</code>	<code>snipers.out</code>
2 2	4

## Задача Е. Крестики-нолики

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

Крестики-нолики на бесконечном поле — это игра для двух человек. Поле представляет собой бесконечный клетчатый лист бумаги. Игроки ходят по очереди. За один ход игрок ставит в какую-нибудь свободную клетку игрового поля свой символ. Символ первого игрока — крестик, а символ второго игрока — нолик.

Если после очередного хода игрока на поле появляются **пять** стоящих в ряд (по горизонтали, вертикали или диагонали) символов этого игрока, то он объявляется победителем партии, и игра заканчивается.

Однажды одаренный школьник Антон обнаружил у себя в тетради некоторую картинку из крестиков и ноликов. Он не может вспомнить, что это такое: поле его поединка с другом Лешей или просто произвольная картинка из крестов и нулей.

Напишите программу, которая по заданной картине из крестиков и ноликов, определяет, могла ли такая позиция возникнуть на поле в результате игры — либо законченной, либо незаконченной. Предполагается, что игроки всегда соблюдают все правила.

### Формат входного файла

Во входном файле находится картина из тетради Антона. Пустые клетки обозначаются символом `'.'` (точка). Символы игроков обозначаются символами `'X'` и `'O'` (заглавные латинские буквы 'икс' и 'о').

Количество строчек во входном файле не превосходит 100. Количество символов в каждой строке также не превосходит 100. Пустых строчек в файле нет. Гарантируется, что во входном файле будет хотя бы один крестик или нолик.

### Формат выходного файла

В выходной файл выведите слово `CORRECT`, если данная позиция могла возникнуть в результате игры в крестики-нолики на бесконечном поле (в конце игры или в середине). В противном случае выведите слово `INCORRECT`.

### Примеры

<code>xo.in</code>	<code>xo.out</code>
<code>.X ..X ..OXO ....X .....O</code>	<code>CORRECT</code>
<code>.... .XO.. .XO. .XO.. .XO. .XO.. ....</code>	<code>INCORRECT</code>