

## Задача А. Два поросенка

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

Два поросенка — Хрюк и Хряк — хотят поделить желуди, которые им подарил их дедушка. Первоначально желуди находятся в  $N$  горшочках. Все горшочки выглядят как свиные копилки. В них можно класть любое количество желудей, но чтобы достать желуди из горшочка, нужно его разбить. Чтобы донести желуди до своих домиков, поросятам нужно переложить желуди в два горшочка — один горшочек для Хрюка, а другой — для Хряка (поросенок может нести лишь один горшочек).

К сожалению, поросята не умеют считать. Поэтому они решили делить желуди по следующему алгоритму, выполняющемуся, пока количество горшочков больше двух:

- Выбрать и разбить горшочек.
- Разделить желуди, выпавшие из разбитого горшочка, на две равные кучки, если число желудей четно, или на две кучки, отличающиеся по размеру на один желудь, если оно нечетно.
- Выбрать два разных горшочка и засыпать в каждый из них по кучке.

Найдите последовательность ходов, которая разделит желуди наиболее бесчестно, то есть такую, после которой разница между количествами желудей в горшочках у поросят максимальна.

### Формат входного файла

В первой строке файла находится целое положительное число  $N$  — количество горшочков. Вторая строка содержит  $N$  целых положительных чисел  $a_i$  — количество желудей в  $i$ -ом горшочке.  $2 \leq N \leq 100$ ,  $1 \leq a_i \leq 10\,000$ .

### Формат выходного файла

Выведите  $N$  строк — последовательность ходов, описывающих оптимальное решение. Каждая строка должна содержать три номера горшочков: номер разбиваемого горшочка, номер горшочка, куда перекладывается меньшая по размеру кучка, и номер горшочка, куда перекладывается большая по размеру кучка. Горшочки пронумерованы числами от 1 до  $N$ . Если оптимальных решений несколько, выведите любое из них.

### Пример

<code>acorns.in</code>	<code>acorns.out</code>
3	2 1 3
2 3 4	

## Задача В. Шахматы

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

На одном из полей громадной шахматной доски стояли фигуры и спорили. «Я дойду до той клетки за семь ходов!», — кричал Конь. «А я — за пять», — ответил ему Слон. Впрочем, спорить можно было долго, а в планах флажка часов уже маячило падение. Поэтому Королю нужно принять решение: какую фигуру следует направить из клетки 1 в клетку 2, чтобы количество ходов было минимальным.

Размеры доски —  $10^9$  на  $10^9$  клеток.

### Формат входного файла

Во входном файле записаны две пары целых чисел  $x_1 y_1$  и  $x_2 y_2$  — координаты первой и второй клеток, соответственно ( $1 \leq x_i, y_i \leq 10^9$ ). Клетки не совпадают.

### Формат выходного файла

В первой строке выведите название фигуры, которую следует направить на ответственное задание: “Queen” (ферзь), “Knight” (конь), “Bishop” (слон) или “Rook” (ладья). Во второй строке выведите одно целое число — количество необходимых ходов. Затем выведите координаты посещаемых клеток, по одной паре в каждой строке, в порядке их посещения. Выводить координаты начальной клетки не требуется. Если оптимальных ответов несколько, можно выводить любой из них.

### Пример

<code>chess.in</code>	<code>chess.out</code>
5 6 3 8	Bishop 1 3 8

## Задача С. Приземление

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

Космический корабль приземляется на прямоугольную площадку космодрома. Правда, пилот не слишком опытен, поэтому возможны некоторые неточности...

Площадка представляет собой прямоугольник со сторонами, параллельными осям координат. Корабль (вид сверху) — выпуклый многоугольник.

Точка находится в безопасности, если расстояние от нее до корабля не меньше  $d$ . Автоматические системы не дадут посадить корабль так, чтобы хоть одна точка вне площадки не оказалась в безопасности. Также эти системы не разрешат вращение корабля в плоскости площадки, однако разрешат его параллельный перенос. Гарантируется, что корабль можно посадить.

Вычислите площадь той части площадки космодрома, все точки которой будут в безопасности при любом случае посадки, не нарушающем указанные выше правила.

### Формат входного файла

В первой строке входного файла находятся 4 целых числа:  $N$  (количество вершин многоугольника, описывающего корабль),  $W$ ,  $H$  (размеры площадки) и  $d$  (безопасное расстояние).  $3 \leq N \leq 100$ ,  $0 \leq W, H \leq 10^4$ . Затем следует  $N$  пар целых чисел  $X_i Y_i$  через пробел, по одной паре в каждой строке ( $0 \leq X_i \leq W$ ,  $0 \leq Y_i \leq H$ ) — координаты планируемого положения корабля, заданные в порядке обхода. Последовательные точки в порядке обхода границы многоугольника могут лежать на одной прямой и даже совпадать.

### Формат выходного файла

Выходной файл должен содержать одно вещественное число с не менее, чем 6 точными знаками после запятой — площадь космодрома, на которой находиться будет безопасно.

### Пример

landing.in	landing.out
3 100 100 1 10 10 20 10 10 20	56.7162717227

## Задача D. Таблица

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

Сгенерируйте такую квадратную таблицу размера  $N \times N$  из натуральных чисел, что для каждой строки сумма всех чисел в ней делится на  $P$ , а для каждого столбца сумма всех чисел в нем не делится на  $P$ , или выясните, что этого сделать нельзя.

### Формат входного файла

В первой строке входного файла записаны два целых числа  $N$  и  $P$  через пробел ( $1 \leq N \leq 100$ ,  $2 \leq P \leq 10^9$ ).

### Формат выходного файла

Если требуемую таблицу построить невозможно, выведите “No solution” (без кавычек) в первой строке выходного файла. В противном случае выведите в первой строке “Solution exists” (без кавычек), а в следующих  $N$  строках выведите по  $N$  натуральных чисел в каждой через пробел — искомую таблицу. Элементы таблицы не должны превышать  $10^9$ .

### Пример

<code>matrix.in</code>	<code>matrix.out</code>
4 7	Solution exists 4 2 3 5 7 3 6 5 5 2 1 6 3 2 5 4

## Задача Е. Многоугольники

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

Требуется определить, лежит ли один выпуклый многоугольник строго внутри другого выпуклого многоугольника, то есть верно ли, что все точки одного многоугольника находятся строго во внутренности другого.

### Формат входного файла

Во входном файле заданы один за другим два многоугольника. Многоугольник описывается так: в первой строке описания задано количество вершин  $N$  ( $3 \leq N \leq 10^5$ ), а в следующих  $N$  строках — координаты его вершин  $x_i y_i$  через пробел в порядке обхода. Последовательные точки в порядке обхода границы многоугольника могут лежать на одной прямой и даже совпадать. Координаты не превышают  $10^6$  по абсолютной величине. Все числа во входном файле целые.

### Формат выходного файла

Выведите “Yes”, если второй многоугольник лежит строго внутри первого, и “No” в противном случае.

### Пример

<code>polygons.in</code>	<code>polygons.out</code>
4 -2 -2 -2 2 2 2 2 -2 3 -1 0 0 0 0 1	Yes

## Задача F. Палочки

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

В некоторых квадратах сетки размера  $m$  на  $n$  нарисованы палочки. Палочки нарисованы либо между левым верхним и правым нижним углами ячейки, либо между правым верхним и левым нижним. В каждой ячейке не более одной палочки. За один ход можно взять произвольную палочку и повернуть ее на  $90$  градусов. За какое минимальное количество ходов можно добиться того, чтобы все палочки были связаны? Связанность — это возможность зарисовать расположение всех палочек, не отрывая перо от бумаги, при необходимости проводя перо по палочке несколько раз.

### Формат входного файла

В первой строке находятся два натуральных числа  $m$  и  $n$  ( $1 \leq m, n \leq 100$ ). В следующих  $m$  строках, каждая из которых содержит по  $n$  символов, задается начальное положение палочек. Символ `/` соответствует палочке из правого верхнего угла в левый нижний, а символ `\` — палочке из левого верхнего угла в правый нижний. Клетки, на которых нет палочек, задаются точками.

### Формат выходного файла

Вывести минимальное количество ходов. Если решения не существует, выведите `-1`.

### Примеры

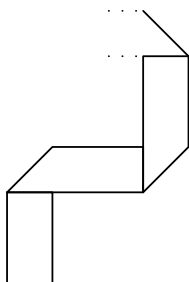
<code>sticks.in</code>	<code>sticks.out</code>
<pre>4 4 .... //\\ \..\ ....</pre>	<pre>3</pre>
<pre>2 3 \.. ../</pre>	<pre>-1</pre>

## Задача G. Цветная полоска

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

Есть полоска бумаги шириной 1 сантиметр, выкрашенная с одной стороны в зеленый, а с другой стороны — в фиолетовый цвет. Ее постепенно выкладывают на стол, иногда складывая и поворачивая на 90 градусов. В результате этих действий полоска частично оказывается на столе фиолетовой стороной вверх, частично — зеленой. Требуется вычислить площади видимых частей полоски обоих цветов.

Более точно: полоска условно делится на квадраты размера  $1 \times 1$  сантиметр. В начале центр первого квадрата фиксируется на столе. На  $i$ -м шаге от центра последнего зафиксированного квадрата отступают  $K_i$  сантиметров и фиксируют там центр соответствующего квадрата. Затем полоску сгибают в этом месте так, чтобы на месте квадрата, где находится сгиб, образовался прямоугольный треугольник. Полоска сгибается через верх, то есть цвет  $i$ -го треугольника не совпадает с цветом  $i$ -го отрезка. Затем переходят к следующему шагу.



Сначала полоска выкладывается зеленой стороной вверх.

### Формат входного файла

В первой строке входного файла задано число  $N$  ( $1 \leq N \leq 100$ ) — количество отрезков, на которых полоску выкладывают прямо. Затем следует  $N$  строк, в каждой из которых содержится по два числа  $K_i$  и  $T_i$  — длина соответствующего отрезка в сантиметрах и сторона, в которую совершается поворот после этого отрезка (1 соответствует повороту вправо, 2 — повороту влево). Все  $K_i$  целые,  $1 \leq K_i \leq 10^7$ .

### Формат выходного файла

В выходной файл следует вывести два числа, разделенных пробелом или переводом строки: площади зеленого и фиолетового цветов, соответственно, в квадратных сантиметрах. Необходимо вывести как можно более точный ответ.

### Примеры

<code>stripe.in</code>	<code>stripe.out</code>
1	10.00
10 1	0.50
2	10.50
10 1	19.50
20 2	

## Задача Н. Система

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

План разрабатываемой в настоящее время тестирующей системы «Дельтаплан» представляет собой набор модулей и зависимостей между ними. В процессе отладки, совмещенной с бета-тестированием, разработчики пришли к следующему выводу: зависимостей между модулями слишком много. Было принято решение удалить некоторое (возможно, пустое) множество зависимостей так, чтобы каждый модуль оказался зависим напрямую не более чем от двух других. Ситуация осложняется еще и тем, что разработчики не могут прийти к единственному решению насчет множества зависимостей, которые будут устранены. Отношение зависимости между модулями считается симметричным.

Вас очень интересует предполагаемое время окончания бета-тестирования системы, а оно напрямую зависит от количества способов устранить некоторые зависимости. Найдите это количество.

### Формат входного файла

В первой строке входного файла содержится одно число  $N$  ( $1 \leq N \leq 16$ ) — количество модулей в системе. В следующих  $N$  строках по  $N$  чисел в строке записана матрица, задающая зависимости.  $j$ -е число в  $i$ -й строке матрицы равно 1, если между модулями  $i$  и  $j$  существует зависимость, 0 в противном случае. Матрица всегда симметрична относительно главной диагонали. Ни один модуль не зависит от самого себя.

### Формат выходного файла

Выведите единственное число — искомое количество способов.

### Примеры

<code>system.in</code>	<code>system.out</code>
3 0 1 1 1 0 1 1 1 0	8
4 0 1 1 1 1 0 0 0 1 0 0 0 1 0 0 0	7
5 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0	16



## Задача I. XOR

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

*Симметрическая разность* двух бит равна 0, если их значения равны, и 1, если они различны.

Симметрическая разность двух двоичных чисел вычисляется побитово. Обозначим симметрическую разность чисел  $X$  и  $Y$  как  $X \oplus Y$ . Тогда, если  $X = x_0 + 2x_1 + 4x_2 + \dots + 2^{k-1}x_{k-1}$ , а  $Y = y_0 + 2y_1 + 4y_2 + \dots + 2^{k-1}y_{k-1}$ , то  $X \oplus Y = z_0 + 2z_1 + 4z_2 + \dots + 2^{k-1}z_{k-1}$ , где  $z_i$  — это симметрическая разность  $x_i$  и  $y_i$ .

Дан набор  $A$  из  $N$  целых чисел. Нужно найти в нем какой-нибудь поднабор  $B$  такой, что  $B_1 + \dots + B_k = B_1 \oplus \dots \oplus B_k$  ( $k \geq 2$ ).

### Формат входного файла

В первой строке входного файла задано число  $N$  ( $1 \leq N \leq 2000$ ), а в нескольких следующих строках — числа  $A_1, A_2, \dots, A_N$ , разделенные пробелами или переводами строк ( $0 \leq A_i \leq 2^{31} - 1$ ).

### Формат выходного файла

Если решение существует, в первой строке выходного файла выведите  $k$  — количество чисел в искомом поднаборе, а во второй строке — сами числа  $B_1, B_2, \dots, B_k$  через пробел. В противном случае выведите “No solution” (без кавычек) в первой строке выходного файла.

### Пример

<code>xor.in</code>	<code>xor.out</code>
3	2
1 2	1 2
3	