

**Задача А. Максимальная разность (высшая лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Вася выбирает два натуральных числа, стандартная запись каждого из которых в системе счисления с основанием  $b$  состоит из  $N$  цифр, и сумма цифр в этих записях одинаковы. Вася хочет сделать выбор таким образом, чтобы разность между числами была максимально возможной. Помогите Васе найти эту разность.

**Ограничения**

$$2 \leq b \leq 36, 1 \leq N \leq 10.$$

**Формат входного файла**

Единственная строка содержит два целых числа  $b$  и  $N$ .

**Формат выходного файла**

Выведите единственное целое число — максимальную разность между  $N$ -значными числами в системе счисления с основанием  $b$  с одинаковой суммой цифр. Результат должен быть представлен в стандартной записи в системе с основанием  $b$ . Для цифр, больших 9, используйте большие латинские буквы:  $A, B, C, \dots, Z$ .

**Пример**

<code>stdin</code>	<code>stdout</code>
10 2	72
16 3	E01

## Задача В. Максимальная разность (юниорская лига)

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Вася выбирает два натуральных  $N$ -значных числа с одинаковой суммой цифр. Он хочет сделать это таким образом, чтобы разность между ними была максимально возможной. Помогите Васе сделать такой выбор.

### Ограничения

$1 \leq N \leq 10$ .

### Формат входного файла

Единственная строка содержит целое число  $N$ .

### Формат выходного файла

Выведите единственное целое число — максимальную разность между  $N$ -значными числами с одинаковой суммой цифр.

### Пример

<code>stdin</code>	<code>stdout</code>
2	72

## Задача С. Построение многоугольника (высшая лига)

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Дана последовательность из  $N$  целых чисел  $a_1, a_2, \dots, a_N$ . Требуется построить выпуклый многоугольник с указанными длинами сторон, никакие три вершины которого не лежат на одной прямой.

### Ограничения

Все заданные значения являются целыми и не превосходят  $10^5$  по абсолютной величине,  $N \geq 3$ .

### Формат входного файла

Первая строка содержит целое число  $N$ . Во второй строке записаны числа  $a_1, a_2, \dots, a_N$ .

### Формат выходного файла

Выведите  $N$  строк. В каждой строке должны находиться декартовы координаты соответствующей вершины треугольника. Расстояние между вершинами с номерами  $i$  и  $i + 1$  должно быть равно  $a_i$ , между вершинами с номерами  $N$  и  $1$  – значению  $a_N$ . Все равенства должны выполняться с точностью не менее  $10^{-5}$ . Все координаты не должны превышать  $2 \cdot 10^9$  по абсолютной величине. Если построить многоугольник с требуемыми свойствами невозможно, выведите одну строку “Impossible”.

### Пример

stdin	stdout
3 3 4 5	0.0000000 3.0000000 0.0000000 0.0000000 4.0000000 0.0000000
3 1 2 3	Impossible
5 2 5 13 10 10	0.0000000 0.0000000 2.0000000 0.0000000 5.0000000 4.0000000 0.0000000 16.0000000 -6.0000000 8.0000000

**Задача D. Построение треугольника (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Даны три целых числа  $a$ ,  $b$  и  $c$ . Требуется построить невырожденный треугольник с указанными длинами сторон.

**Ограничения**

Значение  $a$ ,  $b$  и  $c$  не превосходят  $10^5$  по абсолютной величине.

**Формат входного файла**

Единственная строка содержит три целых числа  $a$ ,  $b$  и  $c$ .

**Формат выходного файла**

Выведите 3 строки. В каждой строке должны находиться декартовы координаты соответствующей вершины треугольника. Расстояние между первой и второй вершиной должно быть равно  $a$ , между второй и третьей –  $b$ , между первой и третьей –  $c$ . Все равенства должны выполняться с точностью до  $10^{-5}$ . Все координаты не должны превышать  $10^6$  по абсолютной величине. Если построить невырожденный треугольник с указанными сторонами невозможно, выведите одну строку “Impossible”.

**Пример**

stdin	stdout
3 4 5	0.0000000 3.0000000 0.0000000 0.0000000 4.0000000 0.0000000
1 2 3	Impossible

**Задача Е. Красивые узоры (высшая лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Компания BrokenTiles планирует заняться выкладыванием в некотором дворе узора из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Двор имеет форму прямоугольника  $N \times M$  метров. Однако оказалось, что  $K$  плиток уже расположены в некоторых клетках двора. Требуется определить, сколько вариантов узора может быть выполнено на этом дворе, при условии, что узор должен быть красивым.

Узор считается красивым, если в любом квадрате  $2 \times 2$  есть либо три черных плитки и одна белая или, наоборот, одна черных и три белых.

**Ограничения**

$$1 \leq N, M \leq 10^5, 0 \leq K \leq 10^5.$$

**Формат входного файла**

В первой строке даны три целых числа  $N$ ,  $M$  и  $K$ . В каждой из последующих  $K$  строк записаны по три целых числа  $x$ ,  $y$ ,  $c$  ( $1 \leq x \leq N$ ,  $1 \leq y \leq M$ ), обозначающих, что в клетке с координатами  $(x, y)$  находится плитка. Если она черная, то  $c = 0$ , а если белая, то  $c = 1$ . Все клетки — различны.

**Формат выходного файла**

Выведите одно число — остаток от деления на  $10^9 + 7$  количества различных красивых узоров, которые могут быть выполнены на данном дворе. Узоры считаются различными, если есть хотя бы одна клетка, в которой в одном узоре лежит белая плитка, а в другом — черная.

**Пример**

<b>stdin</b>	<b>stdout</b>
5 3 5 2 1 0 5 1 1 1 2 1 4 2 0 3 3 0	4

**Задача F. Красивые узоры (юниорская лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Компания BrokenTiles планирует заняться выкладыванием в некотором дворе узора из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Двор имеет форму прямоугольника  $N \times M$  метров. Однако оказалось, что  $K$  плиток уже расположены в некоторых клетках двора. Требуется определить, сколько вариантов узора может быть выполнено на этом дворе, при условии, что узор должен быть красивым.

Узор считается красивым, если в любом квадрате  $2 \times 2$  есть либо три черных плитки и одна белая или, наоборот, одна черных и три белых.

**Ограничения**

$$1 \leq N, M \leq 10, 0 \leq K \leq NM.$$

**Формат входного файла**

В первой строке даны три целых числа  $N$ ,  $M$  и  $K$ . В каждой из последующих  $K$  строк записаны по три целых числа  $x$ ,  $y$ ,  $c$  ( $1 \leq x \leq N$ ,  $1 \leq y \leq M$ ), обозначающих, что в клетке с координатами  $(x, y)$  находится плитка. Если она черная, то  $c = 0$ , а если белая, то  $c = 1$ . Все клетки — различны.

**Формат выходного файла**

Выведите одно число — количество различных красивых узоров, которые могут быть выполнены на данном дворе. Узоры считаются различными, если есть хотя бы одна клетка, в которой в одном узоре лежит белая плитка, а в другом — черная.

**Пример**

<b>stdin</b>	<b>stdout</b>
5 3 5 2 1 0 5 1 1 1 2 1 4 2 0 3 3 0	4

## Задача G. Строка без повторений (высшая лига)

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Рассмотрим строку символов. Будем говорить, что в строке  $s_1s_2\dots s_n$  есть повторение, если в ней есть две совпадающие подстроки, следующие непосредственно одна за другой. То есть, если для некоторых  $i$  и  $k$  ( $i, k > 0, i + 2k - 1 \leq n$ ) выполняется  $s_i = s_{i+k}, s_{i+1} = s_{i+k+1}, \dots, s_{i+k-1} = s_{i+2k-1}$ .

Найдите строку длины  $n$  без повторений с минимальным количеством использованных букв.

### Ограничения

$$1 \leq n \leq 4 \cdot 10^6.$$

### Формат входного файла

В единственной строке задается одно целое число  $n$  — длина искомой строки.

### Формат выходного файла

В первой строке выведите минимальное количество различных символов, которые нужно использовать для построения строки без повторений, а во второй строке — искомую строку. Разрешается использовать лишь маленькие латинские буквы. Гарантируется, что при всех входных данных 26 символов будет достаточно для построения строки без повторений (возможно, что не оптимальной).

### Пример

stdin	stdout
5	3 abaca

**Задача Н. Строка без повторений (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Рассмотрим строку символов. Будем говорить, что в строке  $s_1s_2\dots s_n$  есть повторение, если в ней есть две совпадающие подстроки, следующие непосредственно одна за другой. То есть, если для некоторых  $i$  и  $k$  ( $i, k > 0, i + 2k - 1 \leq n$ ) выполняется  $s_i = s_{i+k}, s_{i+1} = s_{i+k+1}, \dots, s_{i+k-1} = s_{i+2k-1}$ .

Найдите лексикографически минимальную строку длины  $n$  без повторений.

**Ограничения**

$$1 \leq n \leq 4 \cdot 10^6.$$

**Формат входного файла**

В единственной строке задается одно целое число  $n$  – длина искомой строки.

**Формат выходного файла**

Выведите одну искомую строку. Разрешается использовать лишь маленькие латинские буквы. Гарантируется, что при всех входных данных 26 символов будет достаточно для построения строки без повторений (возможно, что не оптимальной).

**Пример**

stdin	stdout
5	abaca



**Задача I. Сбор бобов (высшая лига)**

Вход: `stdin`  
 Выход: `stdout`  
 Ограничение по времени: 1 с  
 Ограничение по памяти: 64 Мб

Есть бесконечная в одну сторону полоса, состоящая из клеток, пронумерованных числами  $0, 1, 2, \dots$ . В некоторых клетках лежит некоторое количество бобов. Каждый ход разрешается выполнять следующее действие. Если в клетке с номером  $i$  ( $i > 0$ ) есть не менее  $i$  горошин, то  $i$  из них забираются из этой клетки и раскладываются по одной в клетки с номерами  $i - 1, i - 2, \dots, 0$ . Если есть несколько таких клеток, то можно делать ход из любой из них. Ваша задача — собрать в клетке с номером 0 все имеющиеся на полосе бобы.

**Ограничения**

$$0 \leq N \leq 2 \cdot 10^5, 0 \leq i_k \leq 10^9, 1 \leq a_k \leq 10^{18}.$$

**Формат входного файла**

В первой строке задано одно целое число  $N$ . В каждой из последующих  $N$  строк задаются по два целых числа  $i_k$  и  $a_k$ , означающих, что в клетке с номером  $i_k$  находится в начальном состоянии  $a_k$  горошин. Гарантируется, что все  $i_k$  различны.

**Формат выходного файла**

Выведите “Yes”, если существует такая последовательность действий, которая приведет к тому, что все горошины окажутся в клетке 0. В противном случае нужно вывести “No”.

**Пример**

<b>stdin</b>	<b>stdout</b>
2 1 1 2 2	Yes
3 1 1 2 2 3 3	No
4 0 3 1 3 2 3 3 3	Yes

**Задача J. Сбор бобов (юниорская лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Есть полоса, состоящая из  $N + 1$  клеток, пронумерованных числами  $0, 1, 2, \dots, N$ . В каждой из клеток лежит некоторое количество бобов. Каждый ход разрешается выполнять следующее действие. Если в клетке с номером  $i$  ( $i > 0$ ) есть не менее  $i$  горошин, то  $i$  из них забираются из этой клетки и раскладываются по одной в клетки с номерами  $i - 1, i - 2, \dots, 0$ . Если есть несколько таких клеток, то можно делать ход из любой из них. Ваша задача — собрать в клетке с номером 0 все имеющиеся на полосе бобы.

**Ограничения**

$$0 \leq N \leq 10^5, 0 \leq a_i \leq 10^9.$$

**Формат входного файла**

В первой строке задано одно целое число  $N$ . Во второй строке записаны  $N + 1$  целых чисел  $a_0, a_1, \dots, a_N$ , где  $a_i$  — количество горошин в клетке с номером  $i$ .

**Формат выходного файла**

Выведите “Yes”, если существует такая последовательность действий, которая приведет к тому, что все горошины окажутся в клетке 0. В противном случае нужно вывести “No”.

**Пример**

<b>stdin</b>	<b>stdout</b>
2 0 1 2	Yes
3 0 1 2 3	No
3 3 3 3 3	Yes

## Задача К. Обратный сбор бобов (высшая лига)

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Есть бесконечная в одну сторону полоса, состоящая из клеток, пронумерованных числами  $0, 1, 2, \dots$ . У вас есть  $K$  горошин, которые вы должны разместить в клетках этой полосы. Необходимо сделать это таким образом, чтобы существовала последовательность действий, которая приведет к тому, что все  $K$  бобов окажутся в клетке с номером 0. Напомним, что разрешенным является следующее действие. Если в клетке с номером  $i$  ( $i > 0$ ) есть не менее  $i$  горошин, то  $i$  из них забираются из этой клетки и раскладываются по одной в клетки с номерами  $i - 1, i - 2, \dots, 0$ . Если есть несколько таких клеток, то можно делать ход из любой из них.

### Ограничения

$$0 \leq K \leq 10^{11}.$$

### Формат входного файла

В единственной строке записано одно целое число  $K$ .

### Формат выходного файла

Выведите  $N$  строк, где  $N$  – количество ячеек, в которых будет ненулевое количество горошин. В каждой строке должны быть указаны два целых числа  $i_k$  и  $a_k$ , означающих, что в клетке с номером  $i_k$  находится в начальном состоянии  $a_k$  горошин. Строки должны выводиться в порядке увеличения  $i_k$ .

Если существует несколько вариантов расположения, выберите антилексикографически минимальный. Напомним, что набор  $(x_0, x_1, \dots, x_i, \dots)$  антилексикографически предшествует  $(y_0, y_1, \dots, y_i, \dots)$ , если существует такое  $i$ , что  $x_i > y_i$  и  $x_j = y_j$  для всех  $j > i$ .

### Пример

<code>stdin</code>	<code>stdout</code>
3	1 1 2 2
7	1 1 3 2 4 4

**Задача L. Обратный сбор бобов (юниорская лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Есть бесконечная в одну сторону полоса, состоящая из клеток, пронумерованных числами  $0, 1, 2, \dots$ . У вас есть  $K$  горошин, которые вы должны разместить в клетках этой полосы. Необходимо сделать это таким образом, чтобы существовала последовательность действий, которая приведет к тому, что все  $K$  бобов окажутся в клетке с номером 0. Напомним, что разрешенным является следующее действие. Если в клетке с номером  $i$  ( $i > 0$ ) есть не менее  $i$  горошин, то  $i$  из них забираются из этой клетки и раскладываются по одной в клетки с номерами  $i - 1, i - 2, \dots, 0$ . Если есть несколько таких клеток, то можно делать ход из любой из них.

**Ограничения**

$$0 \leq K \leq 10^5.$$

**Формат входного файла**

В единственной строке записано одно целое число  $K$ .

**Формат выходного файла**

Выведите  $N$  строк, где  $N$  — количество ячеек, в которых будет ненулевое количество горошин. В каждой строке должны быть указаны два целых числа  $i_k$  и  $a_k$ , означающих, что в клетке с номером  $i_k$  находится в начальном состоянии  $a_k$  горошин. Строки должны выводиться в порядке увеличения  $i_k$ .

Если существует несколько вариантов расположения, выберите антилексикографически минимальный. Напомним, что набор  $(x_0, x_1, \dots, x_i, \dots)$  антилексикографически предшествует  $(y_0, y_1, \dots, y_i, \dots)$ , если существует такое  $i$ , что  $x_i > y_i$  и  $x_j = y_j$  для всех  $j > i$ .

**Пример**

<code>stdin</code>	<code>stdout</code>
3	1 1 2 2
7	1 1 3 2 4 4

**Задача М. Уравнение (высшая лига)**

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Имеется уравнение

$$a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 = y p^2,$$

в котором  $a_i$ ,  $N$  и  $p$  — известные целые числа, а  $x$  и  $y$  — неизвестные целочисленные переменные, причем  $x \geq 0$ . Требуется проверить существует ли у данного уравнения решение в целых числах.

**Ограничения**

$0 \leq N \leq 20$ ,  $|a_i| \leq 10^9$ ,  $2 \leq p \leq 10^6$ ,  $p$  — простое число.

**Формат входного файла**

В первой строке записаны два целых числа  $p$  и  $N$ . Вторая строка содержит  $N + 1$  чисел  $a_N, \dots, a_0$ .

**Формат выходного файла**

Если уравнение не имеет решений в целых числах, выведите число  $-1$ . В противном случае выведите значение  $x$  из пары  $(x, y)$ , удовлетворяющей уравнению. Если таких решений несколько, выберите такое, в котором  $x$  принимает наименьшее неотрицательное значение.

**Пример**

<b>stdin</b>	<b>stdout</b>
3 2 1 2 3	3
2 2 1 1 1	-1

**Задача N. Уравнение (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 2 с  
Ограничение по памяти: 64 Мб

Имеется уравнение

$$a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 = yp^2,$$

в котором  $a_i$ ,  $N$  и  $p$  – известные целые числа, а  $x$  и  $y$  – неизвестные целочисленные переменные, причем  $x \geq 0$ . Требуется проверить существует ли у данного уравнения решение в целых числах.

**Ограничения**

$0 \leq N \leq 20$ ,  $|a_i| \leq 10^9$ ,  $1 \leq p \leq 2014$ .

**Формат входного файла**

В первой строке записаны два целых числа  $p$  и  $N$ . Вторая строка содержит  $N + 1$  чисел  $a_N, \dots, a_0$ .

**Формат выходного файла**

Если уравнение не имеет решений в целых числах, выведите число  $-1$ . В противном случае выведите значение  $x$  из пары  $(x, y)$ , удовлетворяющей уравнению. Если таких решений несколько, выберите такое, в котором  $x$  принимает наименьшее неотрицательное значение.

**Пример**

stdin	stdout
3 2 1 2 3	3
2 2 1 1 1	-1

**Задача О. Соревнование (высшая лига)**

Вход: `stdin`  
 Выход: `stdout`  
 Ограничение по времени: 1 с  
 Ограничение по памяти: 64 Мб

На соревновании по карате участвуют две команды  $A$  и  $B$ . В каждой из них есть по  $N$  каратистов с известным уровнем мастерства. Соревнование состоит из  $N$  раундов-поединков, в каждом из которых участвует по одному каратисту от каждой команды. Согласно правилам, каждый каратист может поучаствовать только в одном поединке. Каждый раунд заканчивается победой одного из участников (того, у которого выше уровень мастерства), либо ничьей (если уровни мастерства участников одинаковы). Если зафиксирована ничья в поединке, обе команды за раунд получают по 1 очку, в противном случае победитель приносит своей команде 2 очка, проигравший – 0. В начале соревнования тренеры обеих команд дают организаторам списки, где указывают, в каком порядке будут выходить на ринг их подопечные. Однако тренеру команды  $B$  стало известно, в каком порядке будут выходить участники из команды  $A$ . Обладая этой информацией, он хочет составить свой список таким образом, чтобы набрать как можно больше очков, и просит вас помочь ему.

**Ограничения**

$$1 \leq N \leq 2 \cdot 10^5, 1 \leq a_i, b_i \leq 10^9.$$

**Формат входного файла**

В первой строке задается одно целое число  $N$ . Во второй строке задаются  $N$  чисел  $a_i$ , каждое из которых определяет уровень мастерства соответствующего каратиста команды  $A$ . Значения задаются в том порядке, в котором будут выходить на ринг участники этой команды. В третьей строке задаются  $N$  чисел  $b_i$ , определяющих уровни мастерства каратистов команды  $B$ .

**Формат выходного файла**

Выведите одно число — максимальное количество очков, которое может набрать команда  $B$  за счет выбора тренером этой команды оптимального порядка.

**Пример**

<code>stdin</code>	<code>stdout</code>
3 3 4 5 1 2 6	2
4 4 5 6 2 1 7 3 8	6

**Пояснение**

Во втором примере в первом раунде тренер  $B$  выпустит каратиста с уровнем 1 (который наберет 0 очков в поединке с каратистом уровня 4), а в последующих – каратистов с уровнями 7, 8 и 3 соответственно, которые выиграют свои поединки против 5, 6 и 2 и принесут по два очка.

**Задача Р. Соревнование (юниорская лига)**

Вход: `stdin`  
 Выход: `stdout`  
 Ограничение по времени: 1 с  
 Ограничение по памяти: 64 Мб

На соревновании по карате участвуют две команды  $A$  и  $B$ . В каждой из них есть по  $N$  каратистов с известным уровнем мастерства. Соревнование состоит из  $N$  раундов-поединков, в каждом из которых участвует по одному каратисту от каждой команды. Согласно правилам, каждый каратист может поучаствовать только в одном поединке. Каждый раунд заканчивается победой того из участников, у которого выше уровень мастерства. Победитель приносит своей команде 1 очко, проигравший – 0. В начале соревнования тренеры обеих команд дают организаторам списки, где указывают, в каком порядке будут выходить на ринг их подопечные. Однако тренеру команды  $B$  стало известно, в каком порядке будут выходить участники из команды  $A$ . Обладая этой информацией, он хочет составить свой список таким образом, чтобы набрать как можно больше очков, и просит вас помочь ему.

**Ограничения**

$$1 \leq N \leq 2 \cdot 10^5, 1 \leq a_i, b_i \leq 2N.$$

**Формат входного файла**

В первой строке задается одно целое число  $N$ . Во второй строке задаются  $N$  чисел  $a_i$ , каждое из которых определяет уровень мастерства соответствующего каратиста команды  $A$ . Значения задаются в том порядке, в котором будут выходить на ринг участники этой команды. В третьей строке задаются  $N$  чисел  $b_i$ , определяющих уровни мастерства каратистов команды  $B$ . Гарантируется, что уровни мастерства у всех участников различны.

**Формат выходного файла**

Выведите одно число — максимальное количество очков, которое может набрать команда  $B$  за счет выбора тренером этой команды оптимального порядка.

**Пример**

<code>stdin</code>	<code>stdout</code>
3 3 4 5 1 2 6	1
4 4 5 6 2 1 7 3 8	3

**Пояснение**

Во втором примере в первом раунде тренер  $B$  выпустит каратиста с уровнем 1 (который наберет 0 очков в поединке с каратистом уровня 4), а в последующих – каратистов с уровнями 7, 8 и 3 соответственно, которые выиграют свои поединки против 5, 6 и 2 и принесут по одному очку.



## Задача Q. Вписанная окружность (высшая лига)

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Задан выпуклый многоугольник. Будем говорить, что некоторая окружность является вписанной в данный многоугольник, если все ее точки содержатся внутри многоугольника или на его границе. Требуется найти вписанную окружность с наибольшим радиусом.

### Ограничения

$$3 \leq N \leq 50000, -10^7 \leq x_i, y_i \leq 10^7.$$

### Формат входного файла

В первой строке задается одно целое число  $N$ . В каждой из последующих  $N$  строк содержатся по два вещественных числа  $x_i, y_i$ , имеющих не более 6 знаков после десятичной точки и определяющие координаты соответствующей вершины многоугольника. Вершины многоугольника задаются в порядке обхода. Никакие три точки не лежат на одной прямой.

### Формат выходного файла

Выведите одно число – радиус максимальной вписанной окружности с точностью не менее  $10^{-5}$ .

### Пример

stdin	stdout
3 2.0 0.0 0.0 0.0 1.0 1.0	0.414214
4 -1 0 0 1 2 -1 1 -2	0.707107

**Задача R. Вписанная окружность (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Задан выпуклый многоугольник. Будем говорить, что некоторая окружность является вписанной в данный многоугольник, если все ее точки содержатся внутри многоугольника или на его границе. Требуется найти вписанную окружность с наибольшим радиусом.

**Ограничения**

$$3 \leq N \leq 100, -10^6 \leq x_i, y_i \leq 10^6.$$

**Формат входного файла**

В первой строке задается одно целое число  $N$ . В каждой из последующих  $N$  строк содержатся по два вещественных числа  $x_i, y_i$ , имеющих не более 6 знаков после десятичной точки и определяющие координаты соответствующей вершины многоугольника. Вершины многоугольника задаются в порядке обхода. Никакие три точки не лежат на одной прямой.

**Формат выходного файла**

Выведите одно число — радиус максимальной вписанной окружности с точностью не менее  $10^{-5}$ .

**Пример**

stdin	stdout
3 2.0 0.0 0.0 0.0 1.0 1.0	0.414214
4 -1 0 0 1 2 -1 1 -2	0.707107

**Задача S. Разделяющая прямая (высшая лига)**

Вход: stdin  
 Выход: stdout  
 Ограничение по времени: 1 с  
 Ограничение по памяти: 64 Мб

Задано множество, состоящее из  $N$  точек на плоскости. Прямая называется разделяющей по отношению к данному множеству, если найдутся две точки этого множества, лежащие в разных полуплоскостях относительно этой прямой (но не на самой прямой). Ваша задача — определить для каждой из заданных прямых, является ли она разделяющей или нет.

**Ограничения**

$3 \leq N, M \leq 10^5$ , все координаты — целые числа, не превосходящие  $10^9$  по абсолютной величине.

**Формат входного файла**

В первой строке задается целое число  $N$ . В каждой из последующих  $N$  строк содержатся по два целых числа  $x_i, y_i$ , определяющие координаты соответствующей точки множества. В  $(N + 2)$ -ой строке задается число  $M$ , а в последующих  $M$  строках — по 4 числа  $X_1, Y_1, X_2, Y_2$ , где  $(X_1, Y_1)$  и  $(X_2, Y_2)$  — две различные точки на соответствующей прямой.

**Формат выходного файла**

Выведите  $M$  строк, каждая из которых определяет результат для соответствующей прямой. Если прямая является разделяющей, строка должна содержать два числа — номера каких-либо точек множества, лежащих в разных полуплоскостях относительно прямой. Если же прямая — неразделяющая, строка должна содержать одно число 0.

**Пример**

stdin	stdout
4	1 3
0 2	0
-1 -2	
3 1	
2 -1	
2	
-1 -3 1 1	
2 3 5 4	

**Задача Т. Разделяющая прямая (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Задано множество, состоящее из  $N$  точек на плоскости. Прямая называется разделяющей по отношению к данному множеству, если найдутся две точки этого множества, лежащие в разных полуплоскостях относительно этой прямой (но не на самой прямой). Ваша задача – определить для каждой из заданных прямых, является ли она разделяющей или нет.

**Ограничения**

$3 \leq N, M \leq 10^4$ , все координаты — целые числа, не превосходящие  $10^9$  по абсолютной величине.

**Формат входного файла**

В первой строке задается целое число  $N$ . В каждой из последующих  $N$  строк содержатся по два целых числа  $x_i, y_i$ , определяющие координаты соответствующей точки множества. В  $(N + 2)$ -ой строке задается число  $M$ , а в последующих  $M$  строках — по 4 числа  $X_1, Y_1, X_2, Y_2$ , где  $(X_1, Y_1)$  и  $(X_2, Y_2)$  — две различные точки на соответствующей прямой.

**Формат выходного файла**

Выведите  $M$  строк, каждая из которых определяет результат для соответствующей прямой. Если прямая является разделяющей, строка должна содержать два числа – номера точек множества, лежащих в разных полуплоскостях относительно прямой. Если же прямая – неразделяющая, нужно вывести одно число 0.

**Пример**

stdin	stdout
4	1 3
0 2	0
-1 -2	
3 1	
2 -1	
2	
-1 -3 1 1	
2 3 5 4	

## Задача U. Стрингангуляция (высшая лига)

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Пусть задана некоторая строка  $s_1s_2\dots s_l$ . Предполагаем, что она циклическая, то есть за  $s_l$  следует символ  $s_1$ . Будем называть стрингангуляцией этой строки ее разбиение на три последовательные подстроки (уже не циклические, обозначим их соответственно  $a$ ,  $b$  и  $c$ ), для каждой из которых выполнено неравенство треугольника, то есть

$$a + b > c, b + c > a, c + a > b,$$

где знаком “+” понимается как обычно для строк их конкатенация.

Ваша задача — найти для заданной строки количество различных ее стрингангуляций. Стрингангуляции, отличающиеся лишь обозначениями последовательных подстрок, не считаются различными.

### Ограничения

Длина строки  $s$  находится в диапазоне от 3 до 2014.

### Формат входного файла

Единственная строка определяет строку  $s$ .

### Формат выходного файла

Выведите одно число – количество различных стрингангуляций строки  $s$ .

### Пример

<code>stdin</code>	<code>stdout</code>
aaa	1
cbccbcbcb	2
stringangulation	0

## Задача V. Стрингангуляция (юниорская лига)

Вход: `stdin`  
Выход: `stdout`  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

Пусть задана некоторая строка  $s_1s_2\dots s_l$ . Предполагаем, что она циклическая, то есть за  $s_l$  следует символ  $s_1$ . Будем называть стрингангуляцией этой строки ее разбиение на три последовательные подстроки (уже не циклические, обозначим их соответственно  $a$ ,  $b$  и  $c$ ), для каждой из которых выполнено неравенство треугольника, то есть

$$a + b > c, b + c > a, c + a > b,$$

где знаком “+” понимается как обычно для строк их конкатенация.

Ваша задача — найти для заданной строки количество различных ее стрингангуляций. Стрингангуляции, отличающиеся лишь обозначениями последовательных подстрок, не считаются различными.

### Ограничения

Длина строки  $s$  находится в диапазоне от 3 до 100.

### Формат входного файла

Единственная строка определяет строку  $s$ .

### Формат выходного файла

Выведите одно число — количество различных стрингангуляций строки  $s$ .

### Пример

<code>stdin</code>	<code>stdout</code>
aaa	1
cbccbcbcb	2
stringangulation	0

**Задача W. Построение куба (высшая лига)**

Вход: stdin  
 Выход: stdout  
 Ограничение по времени: 1 с  
 Ограничение по памяти: 64 Мб

В трехмерном пространстве задана плоскость уравнением  $Ax + By + Cz + D = 0$ . Требуется построить куб, вершины которого удалены от этой плоскости на заданные расстояния  $d_1, d_2, \dots, d_8$ .

**Ограничения**

Все значения — вещественные, не превышают  $10^4$  по абсолютной величине и имеют не более двух знаков в дробной части. Хотя бы одно из чисел  $A, B$  и  $C$  отлично от нуля, все значения  $d_i$  — неотрицательны.

**Формат входного файла**

В первой строке задаются коэффициенты уравнения плоскости  $A, B, C$  и  $D$ , а во второй строке расстояния  $d_1, d_2, \dots, d_8$ .

**Формат выходного файла**

Выведите 8 строк, каждая из которых будет содержать по три числа — координаты  $x, y$  и  $z$  соответствующей вершины куба. Первая выведенная точка должна находиться на расстоянии  $d_1$  от плоскости (с точностью до  $10^{-5}$ ), вторая — на расстоянии  $d_2$  и т.д.

**Пример**

stdin	stdout
1.0 1.0 1.0 1.0 1.0 3.0 2.0 4.0 2.0 3.0 3.0 2.0	0.244017 0.244017 0.244017 1.976068 1.976068 0.244017 1.976068 0.244017 0.244017 1.976068 1.976068 1.976068 0.244017 1.976068 0.244017 1.976068 0.244017 1.976068 0.244017 1.976068 1.976068 0.244017 0.244017 1.976068
-1 2 -3 6 0 0 0 0 0 0 0 0	Impossible

**Задача X. Построение квадрата (юниорская лига)**

Вход: stdin  
Выход: stdout  
Ограничение по времени: 1 с  
Ограничение по памяти: 64 Мб

На плоскости задана прямая уравнением  $Ax + By + C = 0$ . Требуется построить квадрат, вершины которого удалены от этой прямой на заданные расстояния  $d_1, d_2, d_3, d_4$ .

**Ограничения**

Все значения — вещественные, не превышают  $10^4$  по абсолютной величине и имеют не более двух знаков в дробной части. Хотя бы одно из чисел  $A$  и  $B$  отлично от нуля, все значения  $d_i$  — неотрицательны.

**Формат входного файла**

В первой строке задаются коэффициенты уравнения прямой  $A, B, C$ , а во второй строке расстояния —  $d_1, d_2, d_3, d_4$ .

**Формат выходного файла**

Выведите 4 строки, каждая из которых будет содержать по два числа — координаты  $x$  и  $y$  соответствующей вершины квадрата. Первая выведенная точка должна находиться на расстоянии  $d_1$  от прямой (с точностью до  $10^{-5}$ ), вторая — на расстоянии  $d_2$  и т.д.

**Пример**

stdin	stdout
1.0 2.0 3.0 1.0 2.0 3.0 4.0	-0.152786 -0.305573 2.083282 -0.305573 -0.152786 1.930495 2.083282 1.930495
-1 2 -3 6 0 0 0 0 0 0 0 0	Impossible