

## А. Разрешение мониторов

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Дано разрешение монитора в формате  $W \times H$ , где  $W$  – ширина, а  $H$  – высота. Требуется для каждого разрешения определить его тип: 4:3, 16:9 или ни то и ни другое.

### Входные данные

Разрешение монитора в формате  $W \times H$ .  $W$  и  $H$  – целые числа из диапазона от 10 до 10000. Два числа разделяет строчная латинская буква «икс».

### Выходные данные

Выведите одну из трёх строк в зависимости от разрешения монитора: «4:3», «16:9» или «another».

### Примеры

Ввод	Вывод
1024x768	4:3
1600x900	16:9
1280x1024	another
768x1024	another

## В. Прочность стены

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Рассмотрим стену длины  $L$  и высотой  $H$ . Стена выложена из кирпичей  $1 \times W$ , где  $W$  – ширина кирпича, целое положительное число. Кирпичи плотно прилегают друг к другу, полностью покрывая стену, не выходя за её границы и не образуя отверстий. Однако,  $N$  из этих кирпичей считаются *ненадёжными*. Также к *ненадёжным* кирпичам относятся те кирпичи, край (левый, правый или сразу оба) которых находится на *ненадёжном* кирпиче. Под краем кирпича понимается участок кирпича длиной 1 от левого или правого конца кирпича. Нужно найти все кирпичи, которые считаются *ненадёжными*.

### Входные данные

Два целых числа  $L$  ( $1 \leq L \leq 20$ ) и  $H$  ( $1 \leq H \leq 20$ ) – ширина и высота стены. Далее в  $N$  строках от верхнего уровня стены к нижнему уровню идёт по  $L$  заглавных латинских букв. Каждый кирпич обозначен как последовательность одинаковых заглавных букв. Никакие два кирпича не обозначаются одной буквой. Далее целое число  $N$  ( $0 \leq N \leq 26$ ) – количество кирпичей, которые считаются *ненадёжными*. В следующих  $N$  строках задано по одной заглавной латинской букве, которой обозначается *ненадёжный* кирпич.

### Выходные данные

Выведите  $H$  строк по  $L$  символов в том же формате что и во входных данных. На месте *ненадёжных* кирпичей должны стоять символы \* вместо букв.

### Пример

Ввод	Вывод
10 6 HAAABBBCCC JJFFFFFFGG IIIIIIIIII KKLLLLMMMU NNNNOOOOOT DDDDDDRRS 3 R J B	*****CCC **FFFFFFGG IIIIIIIIII KK*****U NNNN*****T DDDDDD**S

### Пояснение примера

H	A	B	C	
J	F		G	
I				
K	L	M	U	
N		O	T	
D			R	S

Кирпичи R, J, B – *ненадёжные*, согласно исходным данным. Правый край кирпича O находится на кирпиче R. Таким образом, кирпич O также является *ненадёжным*. Кирпич L правым краем находится на кирпиче O, значит и кирпич L считается *ненадёжным*. Рассуждая аналогично, *ненадёжными* будут кирпичи A, B, H, J, L, M, O, R.

## С. 17 делителей

Ограничение по времени: **2 секунды**  
Ограничение по памяти: **65 мегабайт**

Дано целое положительное число N. Требуется найти наибольшее целое положительное число A не превосходящее N, имеющее в точности 17 различных целых положительных делителей. Например, число 12 имеет 6 делителей: 1, 2, 3, 4, 6, 12.

### Входные данные

Целое число N ( $1 \leq N \leq 10^9$ ).

### Выходные данные

Выведите целое положительное число A. Если решения не существует, выведите 0.

### Пример

Ввод	Вывод
18	0

### Пояснение примера

Ни одно число от 1 до 18 не имеет в точности 17 различных целых положительных делителей.

## D. Подсчёт строк

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Дано  $N$  различных строк  $S_1, S_2, \dots, S_N$ , состоящих только из строчных букв латинского алфавита. Последовательно запишем символы всех строк по кругу. Сначала символы первой строки, затем символы второй строки, затем символы третьей строки и т. д. Конец получившейся последовательности соединяется с её началом. Таким образом, у получившейся последовательности нет ни начала, ни конца. Для каждой строки  $S_1, S_2, \dots, S_N$  требуется посчитать, сколько раз она встречается в таком кольце, начиная с различных по месту расположения символов. Например, в кольце *abc*, строки *a*, *b*, *c*, *ab*, *bc*, *ca*, *abc*, *bca*, *cab* встречаются по одному разу, а в строке *aaa* строки *a*, *aa*, *aaa* встречаются по три раза (также смотрите пояснение примера).

### Входные данные

Количество строк  $N$  ( $3 \leq N \leq 10$ ). В следующих  $N$  строках записаны сами строки  $S_1, S_2, \dots, S_N$ . Строки имеют длину от 1 до 10 символов.

### Выходные данные

Выведите строки в том же порядке, что и на входе. После каждого слова через пробел выведите, сколько раз оно встретилось. Используйте формат вывода как в примерах.

### Примеры

Ввод	Вывод
3 abc bdab dabd	abc 1 bdab 3 dabd 2
5 a b c d ab	a 2 b 2 c 1 d 1 ab 2

### Пояснение первого примера

Запишем последовательно символы всех строк. Получим строку *abcdbabdabd*. Таким образом, получили кольцо из одиннадцати символов. Учитывая то, что символы записаны по кругу, строка *abc* встречается только начиная с первого символа, строка *bdab* встречается начиная с четвертого, седьмого и десятого символа, а строка *dabd* начиная с пятого и восьмого символа. Соответственно строки в кольце встречаются 1, 3 и 2 раза.

## Е. Центральная симметрия

Ограничение по времени: **2 секунды**  
Ограничение по памяти: **65 мегабайт**

Дано  $N$  точек на плоскости. Никакие две точки не имеют одинаковые координаты. Требуется определить, существует ли такая точка, что относительно неё данный набор точек обладает центральной симметрией. *Центральной симметрией* относительно точки  $A$  называют преобразование пространства, переводящее точку  $X$  в такую точку  $X'$ , что  $A$  — середина отрезка  $XX'$ . Заметьте, что такая точка не обязана быть среди точек из набора.

### Входные данные

Количество точек  $N$  ( $1 \leq N \leq 100$ ). В следующих  $N$  строках даны пары целых чисел  $x, y$  ( $-100 \leq x \leq 100, -100 \leq y \leq 100$ ), описывающие координаты точек. Все  $y$ -координаты различны. Все  $x$ -координаты различны.

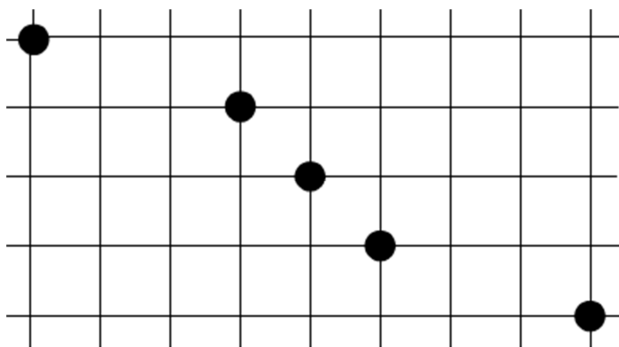
### Выходные данные

Выведите Yes, если набор точек обладает центральной симметрией, или No в противном случае.

### Примеры

Ввод	Вывод
5 0 4 3 3 4 2 5 1 8 0	Yes
3 0 0 1 0 0 1	No

### Пояснение первого примера



Набор точек симметричен относительно точки с координатами (4, 2).

## Ф. Просто задача

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Дано целое число  $N$ . Требуется найти сумму следующих чисел: самого числа  $N$ , количества целых положительных делителей числа  $N$ , квадратного корня числа  $N$  округлённого вниз, произведения цифр числа  $N$ , квадрата числа  $N$ .

### Входные данные

Целое положительное число  $N$  ( $1 \leq N \leq 15$ ).

### Выходные данные

Вывести полученную сумму.

### Пример

Ввод	Вывод
10	117

### Пояснение примера

Само число - 10, количество целых положительных делителей – 4 (1, 2, 5, 10), квадратный корень округлённый вниз – 3 (3.16227766...), произведение цифр – 0 (1 \* 0), квадрат числа – 100 ( $10^2$ ). Их сумма:  $10 + 4 + 3 + 0 + 100 = 117$ .

## Г. Четыре в ряд 3D

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Компания «SMS 3D» разрабатывает сервис для игры «Четыре в ряд 3D» через SMS-сообщения. Два человека поочередно отсылают ходы – пара целых чисел  $x$  и  $y$  через пробел. Ваша задача определить победителя или сообщить о ничьей.



Перед началом игры у каждого игрока имеется по 32 шарика своего цвета. Игровое пространство представляет собой 16 вертикальных столбиков. На каждый из столбиков можно надеть до четырёх шариков (шарики имеют сквозные отверстия). Основания столбцов расположены в 4 ряда по 4 столбика в каждом ряду. Пусть число  $x$  – номер ряда, а число  $y$  – номер столбика в этом ряду. Тогда ход из двух чисел  $x$  и  $y$ , ни что иное как надеть шарик своего цвета на этот столбик. Шарики не могут висеть на столбиках, т. е.

шарик располагается в самой нижней свободной позиции на столбике. Выигрывает тот, кто первым расположит какие-либо четыре бусинки своего цвета в ряд по горизонтали, вертикали или диагонали (см. рис.). При попытке надеть на столбик более 4 шариков считается, что выиграл противник на предыдущем ходе.

### Входные данные

Ввод содержит 64 строки. В каждой по два целых числа  $x$  и  $y$  ( $1 \leq x \leq 4$ ,  $1 \leq y \leq 4$ ). Первый ход – ход первого игрока, второй ход – ход второго игрока, третий ход – ход первого игрока и т. д.

### Выходные данные

Выведите номер выигравшего игрока и номер хода, на котором он выиграл. В случае ничьей выведите 0.

### Пример

Ввод	Вывод
<pre>1 1 1 1 1 1 1 1 1 1 ... (остальные строки не влияют на результат)</pre>	4

## Н. Ломаная линия

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Ломаная линия задана отрезками, соединяющими точки с координатами  $(1, Y_1), (2, Y_2), \dots, (N, Y_N)$  в указанном порядке. Требуется посчитать количество точек на этой линии с целочисленными координатами, у которых есть две соседние точки, причём одна точка выше неё, а другая ниже (одна  $y$ -координата строго больше, а другая  $y$ -координата строго меньше).

### Входные данные

Первая строка содержит целое число  $N$ . В следующей строке через пробел записано  $N$  целых чисел  $Y_1, Y_2, \dots, Y_N$  ( $y$ -координаты точек). Все числа в диапазоне от 1 до 1000.

### Выходные данные

Выведите одно число – количество точек, удовлетворяющих условию задачи.

### Примеры

Ввод	Вывод
<pre>5 1 3 5 7 4</pre>	2
<pre>7 4 2 5 3 6 7</pre>	1

# I. Обнуляющая процедура

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Дана одна из перестановок чисел  $(1, 2, \dots, N)$ . Это значит, что задана последовательность чисел из  $N$  элементов, где все числа целые, различные и лежат в диапазоне от 1 до  $N$ . Определим следующую процедуру. Для каждого участка наибольшей длины, содержащего только положительные числа, находим максимум на этом участке. Заменяем значение каждого числа на этом отрезке на найденный максимум минус это число. После какого по счёту выполнения определённой процедуры в последовательности не останется положительных чисел?

## Входные данные

В первой строке задано целое число  $N$  ( $3 \leq N \leq 1000$ ). Во второй строке через пробел записано  $N$  чисел – перестановка чисел  $(1, 2, \dots, N)$ .

## Выходные данные

Выведите количество выполнений процедуры для получения последовательности без положительных чисел.

## Примеры

Ввод	Вывод
6 1 2 5 6 4 3	4
5 1 2 3 4 5	5

## Пояснение первого примера

$$\begin{array}{l}
 \text{1.} \quad \overbrace{1\ 2\ 5\ 6\ 4\ 3}^{\text{max}=6} \rightarrow 5\ 4\ 1\ 0\ 2\ 3 \\
 \text{2.} \quad \overbrace{5\ 4\ 1\ 0}^{\text{max}=5} \quad \overbrace{2\ 3}^{\text{max}=3} \rightarrow 0\ 1\ 4\ 0\ 1\ 0 \\
 \text{3.} \quad \overbrace{0\ 1\ 4\ 0}^{\text{max}=4} \quad \overbrace{1\ 0}^{\text{max}=1} \rightarrow 0\ 3\ 0\ 0\ 0\ 0 \\
 \text{4.} \quad \overbrace{0\ 3\ 0\ 0\ 0\ 0}^{\text{max}=3} \rightarrow 0\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

## Ж. Чёрный круг

Ограничение по времени: **2 секунды**  
Ограничение по памяти: **65 мегабайт**

Дано поле  $N \times N$ , состоящее из квадратов  $1 \times 1$ . Каждый квадрат покрашен в белый или чёрный цвет. Задача – найти наибольший радиус круга с центром в углу какого либо квадрата  $1 \times 1$ , чтобы накрытая этим кругом площадь была полностью чёрной. Если не существует такого круга, следует принять радиус равным 0.

### Входные данные

В первой строке дана длина стороны поля  $N$  ( $3 \leq N \leq 300$ ). В следующих  $N$  строках по  $N$  чисел, описывающих цвета квадратов  $1 \times 1$ . Чёрному квадрату соответствует «1», белому – «0». Гарантируется, что квадраты  $1 \times 1$  на границе поля всегда белые.

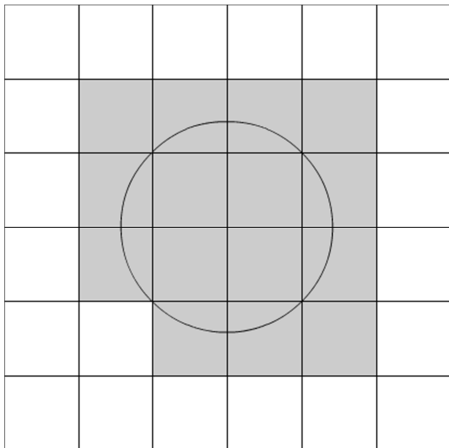
### Выходные данные

Выведите наибольший радиус круга, отвечающего условию задачи. Ответ считается правильным, если он отличается от истинного не более чем на 0.01.

### Примеры

Ввод	Вывод
6 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0	1.414
4 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0	0.000

### Пояснение к первому примеру





## К. Адреса и телефоны

Ограничение по времени: **2 секунды**  
 Ограничение по памяти: **65 мегабайт**

Одна крупная компания-производитель мобильных телефонов занимается разработкой новой операционной системы для своих устройств. Одной из её особенностей будет возможность мгновенно выделить все номера телефонов и адреса электронной почты из любого текста.

Вам, как одному из разработчиков этой системы, поручили реализовать эту функцию.

*Адресом электронной почты* – две непустые последовательности символов (заглавных и строчных латинских букв, цифр, точек и знаков подчёркивания “\_”), разделённые символом “@”. Никакая из этих последовательностей не может начинаться и заканчиваться точкой или содержать несколько подряд идущих точек.

*Номер телефона* – строка, состоящая из цифр и, возможно дефисов (“-“), причём два или более дефиса не могут стоять подряд, а также номер не может начинаться или заканчиваться дефисом.

Каждый адрес и каждый телефон в тексте должен быть ограничен пробелами (или концом строки).

### Входные данные

В единственной строке дана последовательность символов (длиной не более 1000) с ASCII – кодами, большими 31.

### Выходные данные

Адреса и телефоны следует выводить по одному в строке, в порядке появления их в тексте, в формате:

- для адреса электронной почты: “Send to *адрес*”
- для номера телефона: “Call to *номер*” (не «выбрасывайте» дефисы из номера)

### Пример

Ввод	Вывод
hello, can you call 8-990-225 or send e-mail on vasya@mail.ru ?	Call to 8-990-225 Send to vasya@mail.ru
45-09 9-7, 123	Call to 45-09 Call to 123

Все тесты содержат только одну строку.