

## Задача A. A + B

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Напишите программу, которая находит сумму двух заданных целых чисел.

### Формат входных данных

В единственной строке задаются два целых числа  $a$  и  $b$ , не превышающих  $2 \cdot 10^9$  по абсолютной величине.

### Формат выходных данных

Выведите одно число – сумму чисел  $a$  и  $b$ .

### Примеры

standard input	standard output
2 4	6
3 -2	1

## Задача В. Обобщенный принцип Дирихле

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Тем, кто участвовал в олимпиадах по математике, наверняка хорошо известен принцип Дирихле, который формулируется следующим образом:

*Предположим, что  $M + 1$  кроликов рассажены по  $M$  клеткам. Тогда найдется хотя бы одна клетка, в которой будет больше одного кролика.*

В данной задаче мы рассмотрим более общий случай этого классического математического факта. Пусть есть  $N$  кроликов, которых рассаживают по  $M$  клеткам.

Ваша задача – найти точную верхнюю оценку минимального количества кроликов и точную нижнюю оценку максимального количества кроликов в клетках. Иными словами, необходимо найти наименьшее из чисел  $k_1$  такое, что при любой рассадке кроликов по клеткам найдется хотя бы одна клетка, в которой будет не более  $k_1$  кроликов, а также наибольшее из чисел  $k_2$  такое, что при любой рассадке найдется хотя бы одна клетка, в которой будет не менее  $k_2$  кроликов.

### Формат входных данных

В единственной строке задаются два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 10^9$ ).

### Формат выходных данных

Выведите два числа – точную верхнюю оценку минимального количества кроликов  $k_1$  и точную нижнюю оценку максимального количества кроликов  $k_2$ .

### Примеры

standard input	standard output
3 2	1 2
4 1	4 4

## Задача С. Машина Юрия

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Вдохновившись моделями вычислений выдающихся ученых прошлого, молодой программист Юрий решил построить собственную вычислительную машину. Ее память состоит всего из одной ячейки, которая может содержать неотрицательное целое значение. Изначально в этой ячейке содержится значение 0.

Программа же для машины Юрия представляет собой последовательность команд. Есть всего два вида команд:

- инкремент (обозначается буквой `i`): увеличивает значение в ячейке памяти на 1;
- прыжок в начало (обозначается буквой `j`): выполняет безусловный переход к началу программы, к ее самой первой команде.

Юрий быстро понял, что программы, которые содержат хотя бы один прыжок, приводят к заикливанию и никогда не завершаются. Поэтому он решил усовершенствовать машину таким образом, что после выполнения команды прыжка, она удаляется из программы.

После этого Юрий написал некоторую программу для усовершенствованной машины и хотел бы узнать, каким будет значение в ячейке памяти по окончании выполнения этой программы.

### Формат входных данных

Задается строка из символов `i` и `j`, определяющих последовательность команд в программе Юрия. Длина программы не превышает  $10^5$ .

### Формат выходных данных

Выведите одно целое число – результат работы программы.

### Примеры

<code>standard input</code>	<code>standard output</code>
<code>i</code>	1
<code>j</code>	0
<code>ijjijijji</code>	17

## Задача D. Пицца для лентяя

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Пицца разделена радиальными разрезами на  $N$  различных по размеру кусочков, но не полностью – все кусочки соединены между собой расплавленным сыром. В связи с этим, чтобы взять какой-то кусочек, нужно отделить его от соседей. Так как пицца имеет форму круга, у каждого из кусочков есть ровно два соседа.

К пицце по очереди подходят  $N$  человек. Каждый из них выбирает из всех оставшихся кусочков пиццы наибольший. Если этот кусочек соединен со своими соседями, человек отрезает его.

Одним из этих  $N$  человек является Вася, который может занять любое место в очереди. Конечно же Вася хотел бы получить как можно больший кусок пиццы, но при этом он категорически не хочет самостоятельно отрезать себе пиццу.

Помогите Васе узнать, какой наибольший кусочек пиццы он может получить, чтобы ему не пришлось отрезать этот кусочек от соседних.

### Формат входных данных

В первой строке задается одно натуральное число  $N$  ( $N \leq 10^5$ ). Далее следует  $N$  различных натуральных чисел, определяющих размеры соответствующих кусочков пиццы в порядке обхода по кругу. Все числа различны и не превосходят  $10^9$ .

### Формат выходных данных

Выведите единственное целое число – размер наибольшего кусочка пиццы, который не потребует отрезать.

### Примеры

<code>standard input</code>	<code>standard output</code>
3 1 2 3	1
5 8 5 6 10 3	5

## Задача А. Размер бумаги

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Международные стандарты ISO 216:1975 и ISO 269:1985 определяют бумажные форматы серий А, В, С. Все форматы бумаги ISO основываются на метрической системе мер и имеют одно и то же соотношение сторон, равное  $1 : \sqrt{2}$  (соотношение Лихтенберга). Благодаря этому при разрезании листа с такими пропорциями пополам (по длинной стороне) образовавшиеся половины будут сохранять то же отношение.

Серия форматов А определяется следующим образом. Базовый лист бумаги (A0) имеет площадь в  $1 \text{ м}^2$  и соотношение сторон  $1 : \sqrt{2}$ . Последующие размеры получаются разрезанием длинной стороны на две равные части. То есть лист  $A(i + 1)$  имеет в два раза меньшую площадь, чем лист  $Ai$ .

Серия форматов В определена в стандарте как дополнительная серия к А, у которой такое же соотношение сторон  $1 : \sqrt{2}$ , а площадь листа  $Vi$  является геометрическим средним площадей листов  $Ai$  и  $A(i - 1)$  (средним геометрическим двух неотрицательных чисел  $x$  и  $y$  считается число  $\sqrt{xy}$ ). При этом, и площадь листа  $Ai$  является геометрическим средним площадей листов  $Vi$  и  $V(i + 1)$ .

Серия форматов С предназначена для конвертов для листов серии А. Отношение сторон такое же  $1 : \sqrt{2}$ , а площадь листа  $Si$  является средним геометрическим площадей  $Ai$  и  $Vi$ .

Для заданного формата бумаги определите ее размеры.

### Формат входных данных

В единственной строке задаются формат бумаги в виде серии (буква А, В или С) и номера (число от 0 до 10) без пробелов.

### Формат выходных данных

Выведите два числа – ширину (меньшую сторону) и высоту (большую сторону) листа заданного формата в миллиметрах. Результаты должны быть округлены до ближайших целых чисел.

### Пример

	<code>standard input</code>	<code>standard output</code>
A4		210 297

## Задача В. Таблица умножения

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Для совершенствования навыков устного счета Петя любит составлять таблицы умножения. Для этого он нумерует строки таблицы последовательными целыми числами от  $a$  до  $b$ , а столбцы – числами от  $c$  до  $d$ . Далее в каждой клетке таблицы он записывает произведение номера соответствующей строки на номер соответствующего столбца. В результате получается  $(b - a + 1) \times (d - c + 1)$  произведений.

Теперь Петю интересует вопрос – какое из всех этих произведений будет  $k$ -ым по величине.

### Формат входных данных

В единственной строке задаются пять целых числа  $a, b, c, d, k$  ( $-10^6 \leq a \leq b \leq 10^6$ ,  $-10^6 \leq c \leq d \leq 10^6$ ,  $1 \leq k \leq (b - a + 1)(d - c + 1)$ ).

### Формат выходных данных

Выведите одно число –  $k$ -ое по величине произведение, записанное в таблицу.

### Пример

standard input	standard output
2 9 2 9 1	4
3 5 5 7 5	24

## Задача С. Восстановление сообщения

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Центром было перехвачено сообщение, в котором записано предложение, состоящее из слов некоторого языка, но с удаленными пробелами. Центр обратился к вам как лучшим специалистам в области криптографии и попросил разобраться с этим сообщением, восстановив пробелы. При этом вам дается словарь языка, на котором предположительно написано данное предложение.

### Формат входных данных

В первой строке находится сообщение, в котором содержится не более 4000 строчных латинских букв. Во второй строке задается целое число  $N$  ( $1 \leq N \leq 5000$ ), определяющее количество слов в словаре. В каждой из последующих  $N$  строк записано по одному слову языка. Каждое слово состоит от 1 до 20 строчных латинских букв.

### Формат выходных данных

Выведите предложение с восстановленными пробелами. В полученном предложении все слова должны быть из словаря, между парой соседних слов должен быть ровно один пробел.

Если существует несколько вариантов восстановления, необходимо выбрать тот, в котором будет минимальное количество слов. Если вариантов с минимальным количеством слов несколько, можно вывести любой из них.

Если не существует ни одного варианта восстановления, выведите сообщение "NO SOLUTION".

### Пример

standard input	standard output
thisisatestsentence 10 word is a this test sentence his the at ten	this is a test sentence

## Задача D. Переливания

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Есть три сосуда объемом  $V_1$ ,  $V_2$  и  $V_3$  литров, в которых изначально есть  $S_1$ ,  $S_2$  и  $S_3$  литров воды. Кроме того, есть источник с неограниченным количеством воды. Разрешается выполнять следующие действия:

- наполнить любой сосуд водой из источника;
- вылить всю воду из любого сосуда;
- переливать воду из одного сосуда в другой.

В последнем случае переливание просходит до тех пор, пока не произойдет одно из двух событий – либо первый сосуд полностью опустошится, либо второй сосуд полностью наполнится водой.

По заданной последовательности действий нужно определить, какое количество воды окажется в каждом из сосудов.

### Формат входных данных

В первой строке задается три целых числа  $V_1$ ,  $V_2$ ,  $V_3$ , а во второй –  $S_1$ ,  $S_2$  и  $S_3$  ( $0 \leq S_i \leq V_i \leq 1000$ ). В третьей строке задается целое число  $N$  ( $0 \leq N \leq 1000$ ), определяющее количество действий. В каждой из последующих  $N$  строк задаются по два целых числа  $i$  и  $j$  из диапазона от 0 до 3 ( $i \neq j$ ). Число  $i$  определяет номер сосуда, из которого осуществляется переливание, а  $j$  – номер сосуда, в который переливается вода. Значение  $i = 0$  определяет наполнение из сосуда  $j$  из источника, а  $j = 0$  – опустошение сосуда  $i$ .

### Формат выходных данных

Выведите три целых числа – количество воды, которое окажется в первом, втором и третьем сосуде после того, как будут выполнены все действия.

### Пример

standard input	standard output
3 5 7	3 2 2
0 0 1	
6	
0 1	
1 2	
0 1	
1 2	
1 3	
2 1	



## Задача Е. Гардероб

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Как известно, театр начинается с вешалки. Когда очередной зритель приходит в театр, он сдает верхнюю одежду в гардероб и получает номерок. По окончании представления зритель возвращает номерок и получает взамен свою одежду.

В нашем театре есть несколько залов, в которых дают представления. Эти представления могут иметь разную продолжительность. Поэтому разные зрители могут приходить и уходить из театра в разное время.

Известно, что сегодня театр посетит  $N$  групп зрителей.  $i$ -ая группа состоит из  $k_i$  зрителей. Время их прихода –  $b_i$ , а ухода –  $e_i$ . Будем считать, что если к гардеробу подходят несколько человек одновременно, то сначала обслуживаются те, кому нужно сдать одежду (они ведь спешат на представление), а затем те, кто забирает одежду.

Помогите администрации театра узнать, какое количество номерков (и соответственно вешалок) необходимо иметь в гардеробе, чтобы можно было обслужить всех зрителей.

### Формат входных данных

В первой строке задается целое число  $N$  ( $1 \leq N \leq 10^5$ ). В каждой из последующих  $N$  строк записаны по три целых числа, определяющих соответственно количество зрителей  $k_i$ , время прихода  $b_i$  и время ухода  $e_i$  ( $1 \leq k_i \leq 1000$ ,  $0 \leq b_i < e_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число – минимальное количество номерков, которое должно быть изначально в гардеробе.

### Пример

<code>standard input</code>	<code>standard output</code>
3	7
2 1 5	
3 7 10	
4 4 8	

## Задача F. Игра

Имя входного файла:	<code>standard input</code>
Имя выходного файла:	<code>standard output</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

В новом телевизионном шоу участнику предлагают сыграть в игру. Поле для игры представляет собой  $N$  клеток, выстроенных в одну линию и пронумерованных последовательно целыми числами от 1 до  $N$ . При этом в каждой клетке лежит определенная сумма денег.

Игра просходит следующим образом. Изначально игрока помещают в некоторую клетку с номером  $k$ . Находясь в клетке с номером  $i$ , игрок имеет две возможности:

1. Подбросить правильную монетку (то есть монетку, у которой обе стороны выпадают с равными вероятностями). Если на ней выпадет орел, игрок перемещается в следующую клетку (с номером  $i + 1$ ), а если решка – в предыдущую клетку (с номером  $i - 1$ ).
2. Забрать деньги, находящиеся в клетке  $i$ , и покинуть игру.

Игра завершается в двух случаях: либо когда игрок взял деньги в какой-то клетке, либо когда он выйдет за пределы поля (в клетке  $N$  продвинется вперед или в клетке 1 переместится назад). В последнем случае игрок не получит ничего.

Разумеется, игрок хочет выиграть как можно больше денег. Поэтому при выборе стратегии в игре он руководствуется тем, чтобы повысить математическое ожидание выигрыша.

Определить, каким будет ожидаемый выигрыш игрока при оптимальной стратегии.

Математическое ожидание выигрыша определяется как  $\sum_{i=1}^m p_i w_i$ , где  $p_i$  – вероятность того, что игрок выиграет сумму  $w_i$ . Суммирование выполняется по всем возможным исходам игры.

### Формат входных данных

В первой строке задаются два целых числа  $N$  и  $k$  ( $1 \leq k \leq N \leq 3 \cdot 10^5$ ). Вторая строка содержит  $N$  целых чисел, каждое из которых определяет сумму денег, находящуюся в соответствующей клетке. Все числа находятся в диапазоне от 0 до  $10^6$ .

### Формат выходных данных

Выведите одно число – математическое ожидание выигрыша игрока при оптимальной стратегии с точностью не ниже  $10^{-6}$ .

### Пример

<code>standard input</code>	<code>standard output</code>
2 1 1 3	1.5

### Пояснение к примеру

Находясь в клетке 1, игрок мог бы забрать деньги и тем самым выиграть сумму 1. Однако это будет неоптимальной стратегией.

Если же игрок подбросит монетку, то с одной стороны он рискует остаться без денег (это произойдет с вероятностью  $1/2$ ), но с другой стороны он может попасть в клетку 2 (также с вероятностью  $1/2$ ). Если игрок попадет в клетку 2, разумеется он уже не станет подбрасывать монетку, а просто заберет находящуюся в этой клетке сумму 3. При такой стратегии ожидаемый выигрыш будет равен  $0 \cdot 1/2 + 3 \cdot 1/2 = 1.5$ .

## Задача G. Алгоритм Луна

Имя входного файла:	<code>standard input</code>
Имя выходного файла:	<code>standard output</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Если на пластиковой карте, на которую вам перечисляют стипендию, будет испорчена одна цифра номера, это не беда. Ее легко можно восстановить, благодаря знанию алгоритма проверки корректности этих номеров – алгоритму Луна.

Алгоритм Луна заключается в следующем. Сначала все цифры проверяемой последовательности нумеруются справа налево последовательными натуральными числами, начиная с 1. Цифры, оказавшиеся на нечетных местах, остаются без изменений. Те же цифры, которые стоят на четных местах, умножаются на 2. Если в результате умножения возникает число больше 9, оно заменяется суммой своих цифр. Все полученные в результате преобразования цифры складываются. Если сумма кратна 10, то исходные данные верны. Если же нет, значит заданная последовательность ошибочна.

Напишите программу, которая по номеру, в котором неизвестна одна цифра, восстановит эту цифру таким образом, чтобы получившийся номер был корректным.

### Формат входных данных

В первой строке задается одно число  $N$  ( $1 \leq N \leq 100$ ). Вторая строка имеет длину  $N$  и состоит из цифровых символов  $0, \dots, 9$  и ровно одного символа  $x$ , обозначающего неизвестную цифру.

### Формат выходных данных

Выведите возможное значение неизвестной цифры, при котором заданная строка является корректной в соответствии с алгоритмом Луна последовательностью.

### Примеры

<code>standard input</code>	<code>standard output</code>
11 7992739871x	3
5 x2464	5
10 93380x1696	1

## Задача Н. Кенгуру

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Три кенгуру играют в пустыне. Пустыня представляет собой числовую прямую, на которой кенгуру могут находиться в различных точка с целочисленными координатами.

За один ход один из крайних кенгуру (находящихся в точке с минимальной или максимальной координатой) может прыгнуть в любую целую точку строго между двумя другими кенгуру, если хотя бы одна такая точка существует.

Необходимо выяснить, сколько может продолжаться эта игра.

### Формат входных данных

В единственной строке задаются три различных целых числа, определяющих начальные координаты кенгуру. Все числа различны и не превосходят  $2 \cdot 10^9$  по абсолютной величине.

### Формат выходных данных

Выведите два числа – минимальное и максимальное количество ходов, которое может быть выполнено кенгуру во время этой игры.

### Пример

standard input	standard output
4 7 9	1 2

### Пояснение к примеру

Минимальное количество ходов достигается, если кенгуру, находящийся в точке 4, прыгнет в точку 8. После этого у всех кенгуру не будет допустимых прыжков.

Максимальное количество ходов может быть достигнуто следующим образом: кенгуру из точки 9 прыгает в точку 6, а затем кенгуру из точки 7 прыгает в точку 5.

## Задача I. Крестики-нолики

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Крестики-нолики – логическая игра между двумя противниками на квадратном поле размера  $3 \times 3$ . Один из игроков играет “крестиками” (тот, кто ходит первым), другой – “ноликами”. Изначально все клетки пусты. Игроки по очереди ставят на свободные клетки поля свои знаки. Первый, выстроивший в ряд три своих фигуры по вертикали, горизонтали или диагонали, выигрывает и на этом игра заканчивается. В том случае, когда все клетки заполнены и победитель не определен, игра завершается ничьей.

По заданной на некоторый момент в игре позиции, постарайтесь восстановить последовательность ходов игроков с учетом того, что игроки могут играть неоптимально.

### Формат входных данных

В трех строках задаются по три символа. Эти данные определяют некоторую позицию в игре. Символ “X” (прописная латинская «икс») обозначает крестик, символ “O” (прописная латинская «о») – нолик, а символ “.” – пустую клетку.

### Формат выходных данных

Выведите последовательность ходов игроков, приводящих к заданной позиции. Каждый ход выводится в отдельной строке. Ход определяется двумя числами: первое число обозначает номер строки, а второе – номер столбца, в который ставится очередной знак. Нечетные ходы выполняют “крестики”, а четные – “нолики”.

Если указанная позиция в игре невозможна, выведите только одно сообщение “Impossible”.

### Примеры

standard input	standard output
.OX .X. .O.	2 2 1 2 1 3 3 2
XOO OXO OOX	Impossible
X.O X.O X.O	Impossible

## Задача J. Ток-шоу

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

На новое телевизионное ток-шоу были приглашены  $M$  гостей, которых посадили за круглый стол. В ходе передачи планируется обсудить  $N$  различных тем, по каждой из которых гость может высказаться за или против. Для каждого гостя организаторам ток-шоу известно его мнение по каждой теме. Все мнения гостя кодируются целым числом в пределах от 0 до  $2^N - 1$ . Если гость имеет положительное мнение по теме  $i$ , то бит  $i$  в двоичном представлении этого числа равен единице, в противном случае – бит  $i$  равен нулю.

Для того, чтобы ток-шоу казалось максимально объективным, организаторы хотят разделить всех участников на две команды так, чтобы каждый гость попал в какую-то одну из команд. Каждая команда должна состоять из гостей, сидящих рядом друг с другом (то есть одну команду от другой можно отделить прямой линией, пересекающей окружность стола в двух точках). И самое главное условие – если в одной команде есть представитель, который имеет определенное мнение по какой-либо теме, то в другой команде должен быть хотя бы один представитель, имеющий такое же мнение по той же самой теме.

Помогите организаторам определить количество различных допустимых вариантов разбиения гостей на команды. Варианты, отличающиеся только порядком команд, считаются одинаковыми.

### Формат входных данных

В первой строке задается два целых числа  $N$  и  $M$  ( $2 \leq N \leq 30$ ,  $3 \leq M \leq 4 \cdot 10^5$ ). Во второй строке задаются  $M$  целых чисел из диапазона от 0 до  $2^N - 1$ , кодирующих мнения гостей, в порядке рассадки за круглым столом.

### Формат выходных данных

Выведите одно число – количество способов объективно разделить гостей на команды.

### Пример

standard input	standard output
4 5 1 10 0 11 3	2

### Пояснение к примеру

Есть два способа разделить участников. При первом способе гости  $\{1, 2\}$  формируют одну команду, а гости  $\{3, 4, 5\}$  – другую. При втором способе гости разбиваются на команды  $\{3, 4\}$  и  $\{5, 1, 2\}$ .

## Задача К. Круги

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Маше нужно нарисовать на достаточно большом листе бумаги  $N$  кругов с заданными диаметрами. При этом круги не должны пересекаться, но могут касаться друг друга. После этого Маша проводит ломаную, вершинами которой являются центры нарисованных кругов. Ломаная должна пройти через каждый из центров, соединять их можно в любом порядке.

Найдите минимальную длину, которая может быть у такой ломаной.

### Формат входных данных

В первой строке задается целое число  $N$  ( $1 \leq N \leq 1000$ ). Вторая строка содержит  $N$  натуральных чисел, определяющих диаметры кругов, которые необходимо нарисовать Маше. Все числа не превосходят  $10^6$ .

### Формат выходных данных

Выведите одно число – минимальную длину ломаной, проходящей через центры кругов при их оптимальном размещении.

### Пример

<code>standard input</code>	<code>standard output</code>
3 3 5 7	9