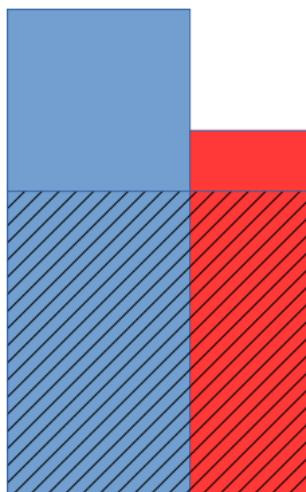


Задача 1. Большой квадрат

Ограничение по времени: 1 секунда

Даны два прямоугольника размера $a \times b$ и $c \times d$. Можно соединить их вместе, приложив сторону одного прямоугольника к стороне другого и склеив место соединения. Прямоугольники можно поворачивать перед склеиванием. После этого из полученной фигуры нужно вырезать квадрат со сторонами, параллельными сторонам прямоугольника. Определите максимальное возможное значение стороны квадрата.

На рисунке изображены два прямоугольника со сторонами 8×3 и 6×2 , из которых можно вырезать квадрат со стороной 5 (заштрихован).



Формат входных данных

Программа получает на вход натуральные числа a, b, c, d , каждое в отдельной строке — стороны первого и второго прямоугольников. Все числа не превосходят 10^9 .

Формат выходных данных

Программа должна вывести одно целое число — максимальную возможную сторону квадрата.

Пример

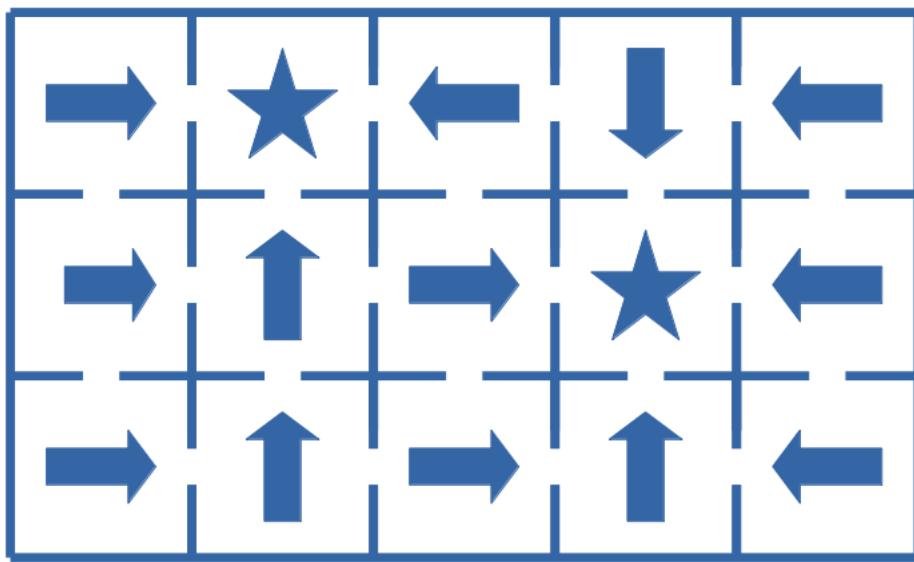
стандартный ввод	стандартный вывод
6	5
2	
3	
8	

Задача 2. План эвакуации

Ограничение по времени: 1 секунда

Этаж здания представляет собой прямоугольник из $n \times m$ квадратных комнат. Из каждой комнаты есть проходы в соседние комнаты. В двух комнатах находятся лестницы. Необходимо разработать план эвакуации — указать для каждой комнаты направление движения в одну из соседних комнат так, чтобы, передвигаясь по комнатам только в указанных направлениях, можно было бы достичь одной из двух лестниц, пройдя минимальное расстояние.

На рисунке изображён возможный план эвакуации для примера из условия. Комнаты с лестницами обозначены звёздочками.



Формат входных данных

Первая строка входных данных содержит число n — количество строк в плане эвакуации, $1 \leq n \leq 100$. Вторая строка входных данных содержит число m — количество столбцов в плане эвакуации, $2 \leq m \leq 100$. Следующие две строки содержат числа r_1 и c_1 — номера строки и столбца комнаты, в которой находится первая лестница, $1 \leq r_1 \leq n$, $1 \leq c_1 \leq m$. Следующие две строки содержат числа r_2 и c_2 — номера строки и столбца комнаты, в которой находится вторая лестница, $1 \leq r_2 \leq n$, $1 \leq c_2 \leq m$. Гарантируется, что $r_1 \neq r_2$ или $c_1 \neq c_2$. Строки нумеруются сверху вниз числами от 1 до n , столбцы нумеруются слева направо числами от 1 до m .

Формат выходных данных

Программа должна вывести n строк, каждая строка должна содержать m символов. Каждый символ соответствует одной комнате. В двух комнатах с лестницами должен находиться символ «S» (прописная английская буква). В остальных комнатах находятся символы, указывающие направление движения:

- «<» (символ «меньше») — налево.
- «>» (символ «больше») — направо.
- «^» (символ находится на клавише «б») — вверх.
- «v» (строчная английская буква) — вниз.

Никакие другие символы, например, пробелы, выводить не нужно. Вы можете вывести любой подходящий план эвакуации.

Пример

стандартный ввод	стандартный вывод
3	>S<v<
5	>^>S<
1	>^>^<
2	
2	
4	

Замечание

Решения, правильно работающие, когда $n = 1$, будут оцениваться в 20 баллов.

Решения, правильно работающие, когда $c_1 = c_2$, будут оцениваться в 20 баллов.

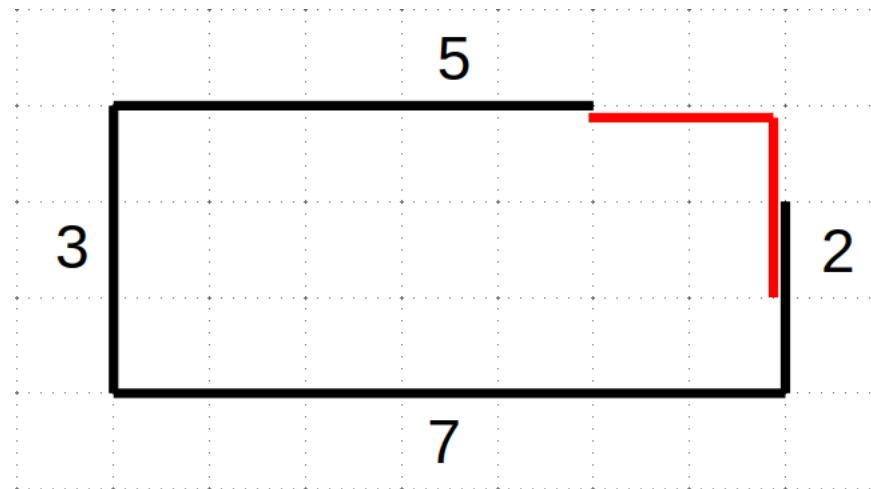
Решения, правильно работающие, когда лестницы находятся в двух противоположных углах здания, будут оцениваться в 20 баллов.

Задача 3. Аркадий Аркадьевич делает грядки

Ограничение по времени: 1 секунда

Аркадий Аркадьевич, широко известный в узких кругах инженер, после выхода на пенсию решил отдохнуть и заняться садоводством. Он уже купил себе участок и собирается сажать там различные овощи и фрукты. Но всем известно, что растения должны расти в грядках, поэтому сейчас Аркадий Аркадьевич занят их сооружением.

Для того чтобы собрать прямоугольную грядку, нужны 4 доски. В идеале это должны быть две пары досок равной длины, тогда из них можно сложить ровный прямоугольник. Но если доски имеют неравную длину, то в одном из углов полученной грядки можно разместить пластиковый уголок: две планки длины r , скреплённые под прямым углом. Уголок со стороной r позволит увеличить длины двух досок на величину, не превосходящую r . Если противоположными сторонами грядки будут доски длины a и b , а также c и d соответственно, то для того чтобы сделать прямоугольную грядку из этих досок, понадобится уголок размера $\max(|a - b|, |c - d|)$. Например, чтобы сделать грядку из досок длины 5, 7, 3, 2, понадобится уголок размера 2. На рисунке чёрным цветом изображены доски и красным цветом изображён уголок.



В сарае у Аркадия Аркадьевича нашлись n досок, i -я из которых имеет длину l_i . Теперь он хочет выбрать из них четыре и сложить из них грядку таким образом, чтобы использовать уголок наименьшего размера. Помогите ему.

Формат входных данных

Первая строка входных данных содержит число n ($4 \leq n \leq 10^5$) — количество досок в сарае у Аркадия Аркадьевича.

Следующие n строк содержат числа l_1, \dots, l_n ($1 \leq l_i \leq 10^9$) — длины досок.

Формат выходных данных

Программа должна сначала вывести число r — минимально возможный размер уголка.

Во второй строке выведите 4 числа a, b, c, d — длины досок, которые необходимо выбрать для грядки. При этом противоположными сторонами прямоугольника будут доски a и b , а также c и d . Если есть различные варианты выбора досок для грядки с одной и той же величиной уголка, можно вывести любой из них.

Система оценки

Решения, правильно работающие, когда $n \leq 30$, будут оцениваться в 20 баллов.

Решения, правильно работающие, когда $n \leq 100$, будут оцениваться в 45 баллов.

Решения, правильно работающие, когда $n \leq 500$, будут оцениваться в 65 баллов.

Решения, правильно работающие, когда все $l_i \leq 30$, будут оцениваться в 10 баллов.

Примеры

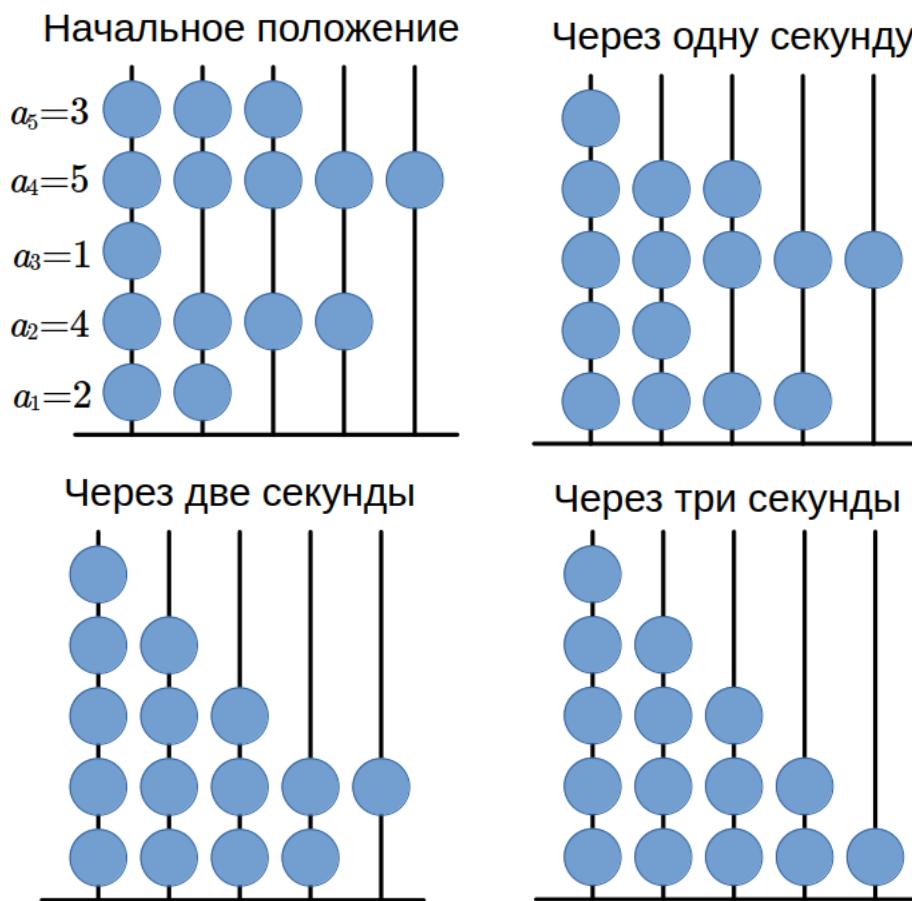
стандартный ввод	стандартный вывод
6 5 9 12 3 7 2	2 2 3 5 7
6 1 7 3 6 3 8	1 3 3 6 7

Задача 4. Гравитационная сортировка

Ограничение по времени: 2 секунды

Алгоритм гравитационной сортировки позволяет упорядочить массив целых неотрицательных чисел. Для сортировки чисел используется набор стержней и бусинок. Рассмотрим алгоритм сортировки массива целых неотрицательных чисел $[a_1, a_2, \dots, a_n]$. Если данные числа могут достигать значения n , нужно использовать n стержней и $a_1 + a_2 + \dots + a_n$ бусинок. Наденем по одной бусинке на самые левые a_1 стержней, они представляют значение a_1 . Затем наденем по одной бусинке на самые левые a_2 стержней, разместив их во втором ряду, над первым рядом бусинок. При этом, если $a_2 > a_1$, то некоторые бусинки второго ряда окажутся как бы висящими в воздухе, они не будут опираться на бусинки нижнего ряда. Затем на самых левых a_3 стержнях в третьем ряду разместим a_3 бусинок, часть из них также может не опираться ни на какие предыдущие бусинки и т.д. После того, как все бусинки будут размещены, они начинают двигаться по стержням вниз. Когда движение завершится, количество бусинок в каждом ряду будет равно значениям упорядоченного массива.

Будем считать, что все бусинки двигаются одновременно. За секунду одна бусинка сползает на один ряд вниз, если в ряду ниже пусто или находится бусинка, которая также спускается вниз. На рисунках показано начальное расположение бусинок для массива $[2, 4, 1, 5, 3]$ и состояние через 1, 2 и 3 секунды.



Для каждого стержня определите время, необходимое для того, чтобы на этом стержне все бусинки закончили движение. В данном примере на первом стержне бусинки будут неподвижны, на втором стержне они закончат движение через одну секунду, на третьем и четвёртом стержнях — через две секунды, на пятом стержне — через три секунды.

Формат входных данных

Первая строка входных данных содержит число n ($1 \leq n \leq 10^5$) — количество сортируемых чисел и количество стержней (то есть максимальное значение одного числа). Следующие n строк содержат по одному числу a_i ($0 \leq a_i \leq n$) — значения упорядочиваемых чисел (начальные значения числа бусинок в каждом ряду).

Формат выходных данных

Для каждого из n стержней слева направо программа должна вывести одно число — количество секунд, через которое на данном стержне бусинки прекратят движение.

Система оценки

Решения, правильно работающие при $n \leq 10$, будут оцениваться в 30 баллов.

Решения, правильно работающие при $n \leq 2000$, будут оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
5	0
2	1
4	2
1	2
5	3
3	

Задача 5. Шифр mint

Ограничение по времени: 1 секунда

Многие старейшие шифры основаны на замене букв на числа, например, в шифре A1Z26 каждая буква заменяется на её порядковый номер в алфавите. Вдохновившись этой идеей, первоклассник Петя решил придумать свой шифр-замену. Он хочет каждую букву от «A» до «R» (первые 18 букв латинского алфавита) заменять на одно из чисел 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90. Числа выбраны так, чтобы при дешифровке легко разделить последовательность цифр на коды букв, причём весь алфавит Петя не смог использовать, ибо сотни он ещё не узнал.

Если применить один из таких шифров к строке, он получит натуральное число. Если использовать разные шифры, удовлетворяющие условию, то будут получаться разные числа.

Петя интересно, какое наименьшее число может быть шифром заданной строки. К сожалению, сравнивать длинные числа слишком сложная задача для первоклассника, поэтому он просит вас написать программу, шифрующую заданное слово минимальным числом.

Формат входных данных

Программа получает на вход непустую строку s , состоящую из прописных букв латинского алфавита от «A» до «R», длина строки не превышает 1000 символов.

Формат выходных данных

Программа должна вывести одно число — шифр строки s . Обратите внимание, число может быть длинным.

Система оценки

Решения, правильно работающие, когда строка состоит не более чем из 4 символов, будут оцениваться в 20 баллов.

Решения, правильно работающие, когда строка состоит из букв «A» и «B», будут оцениваться в 20 баллов.

Решения, правильно работающие, когда строка состоит из букв от «A» до «I», будут оцениваться в 44 балла.

Примеры

стандартный ввод	стандартный вывод
REGION	123456
OLIMPIADA	123453676
CONFIRMABLE	1012023456789