

Задача A. Old Mobile Game

Имя входного файла: `game.in`
Имя выходного файла: `game.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Для старых моделей мобильных телефонов, в которых экран является только текстовым, была разработана следующая игра:

На клетках шахматной доски записаны строчные и заглавные буквы латинского алфавита. За один ход разрешается менять местами две буквы, которые стоят на полях, имеющих общую сторону. После каждого хода подсчитывается количество заработанных очков по следующей схеме: за каждые $N \geq 3$ одинаковых букв, идущих подряд по вертикали или горизонтали, даётся N очков. При этом в случае пересечения двух или более блоков одинаковых букв длины 3 и более оказавшиеся на пересечении буквы учитываются только по одному разу, то есть, например, за «уголок» из 3 горизонтальных и 4 вертикальных одинаковых букв даётся 6 очков, а не 7 — общая буква считается только один раз.

Вам задано игровое поле, подсчёт очков на котором даёт результат 0. Требуется найти все способы сделать один ход так, чтобы количество очков, набранных после этого хода, было максимально.

Формат входного файла

Во входном файле заданы 8 строк по 8 символов в каждой — начальная конфигурация поля. Строки состоят из строчных и заглавных латинских букв. Гарантируется, что никакие 3 одинаковые буквы не идут подряд по вертикали или по горизонтали.

Формат выходного файла

В первой строке выходного файла выведите одно число — максимальное количество набранных очков. Далее выведите все ходы, которые приводят к набору данного количества очков. Для обмена по вертикали выведите номер строки и столбца *верхней* из двух меняемых букв и символ 'D'. Для обмена по горизонтали выведите номер строки и столбца *левой* из двух меняемых букв и символ 'R'. Строки и столбцы нумеруются с нуля. Список ходов должен быть отсортирован по возрастанию номера строки, затем по возрастанию номера столбца, затем вертикальные ходы должны идти ранее горизонтальных.

Примеры

<code>game.in</code>	<code>game.out</code>
abcdefgh ABCDEFGH abcdefgh ABCDZFGH abcZeZgh ABCDZFGH abcdefgh ABCDEFGH	3 3 4 D 4 3 R 4 4 D 4 4 R
abcdefgh ABCDEFGH abcdZfgh ABCDZFGH abcZeZgh ABCeZeGH abcdefgh ABCDEFGH	8 4 4 D

Задача В. Letters in Ancient Temple

Имя входного файла: `letters.in`
Имя выходного файла: `letters.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Во время археологических раскопок в развалинах древнего храма были найдены несколько золотых многогранников, на каждой из граней которых было написано по одной букве древнего алфавита. По-видимому, жрецы выбирали несколько многогранников из набора и поворачивали их одной из граней вверх так, чтобы буквы на обращённых наверх гранях составляли имя одного из богов-покровителей храма. У археологов есть список, содержащий имена всех богов пантеона. Они поручили вам вычислить по заданному набору многогранников количество богов-покровителей данного храма, имена которых можно сложить, используя этот набор.

Формат входного файла

В первой строке входного файла заданы два целых числа M и N — количество многогранников и количество слов в словаре ($1 \leq M, N \leq 200$). Во второй строке заданы M последовательностей символов, описывающих многогранники. i -й многогранник задаётся строкой из l_i заглавных латинских букв, где l_i — количество граней многогранника ($2 \leq l_i \leq 12$). В третьей строке перечислены N имён богов пантеона. Каждое имя состоит из заглавных латинских букв, длина каждого имени не менее 1 и не более 200 символов.

Формат выходного файла

Одно число — количество имён богов из пантеона, которое можно сложить, используя заданный набор многогранников.

Пример

<code>letters.in</code>	<code>letters.out</code>
6 3 ABCD JKEM POQRU SXZ UV NN ARES MOMUS VENUS	2

Задача C. Matrix

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

Задана матрица $n \times m$, состоящая из целых положительных чисел. На вход поступают запросы, каждый из которых задаёт некоторую подматрицу. Для каждого запроса требуется вычислить НОД всех чисел, содержащихся в заданной в запросе подматрице.

Формат входного файла

В первой строке входного файла заданы три целых числа m , n и k — соответственно количество строк и столбцов в матрице и количество запросов ($1 \leq m, n \leq 100$, $1 \leq k \leq 1000$). Далее в последующих m строках заданы по n чисел — j -м в i -й строке идёт элемент на пересечении $(i - 1)$ -й строки и $(j - 1)$ -го столбца в матрице. Строки и столбцы нумеруются неотрицательными целыми числами, начиная с нуля, так, что элемент $(0, 0)$ находится в левом верхнем углу матрицы. В следующих k строках описаны запросы, по одному на строку. Каждый запрос состоит из четырёх чисел t , l , h , w , обозначающих соответственно номер самой верхней строки подматрицы, номер самого левого столбца подматрицы, количество строк в подматрице, количество столбцов в подматрице ($0 \leq t < m$, $0 \leq l < n$, $1 \leq h \leq m - t$, $1 \leq w \leq n - l$).

Формат выходного файла

Для каждого запроса выведите НОД всех чисел заданной в запросе подматрицы.

Примеры

<code>matrix.in</code>	<code>matrix.out</code>
2 2 4 2 3 5 6 0 0 1 1 0 1 2 1 0 0 1 2 0 0 1 1	2 3 1 2
2 2 1 1234321 1234321 2468642 2468642 0 0 2 2	1234321

Задача D. Message To Aliens

Имя входного файла: `message.in`
Имя выходного файла: `message.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

При запуске очередного межпланетного зонда на борт традиционно было помещено послание к инопланетным цивилизациям на случай, если зонд когда-нибудь будет ими обнаружен. Послание состоит из $2N$ квадратных изображений, расположенных на прямоугольной металлической пластине в два ряда и N столбцов. При этом высота ряда равна длине стороны наибольшего квадрата в ряду, аналогично ширина каждого столбца равна длине стороны наибольшего квадрата в этом столбце. Так как места на борту межпланетных зондов очень мало, то руководство проекта хочет минимизировать размеры пластины с посланием.

Вам заданы размеры изображений. Требуется разместить изображения так, чтобы площадь пластины была минимальна, и вывести соответствующее значение площади.

Формат входного файла

В первой строке входного файла задано целое число N — количество столбцов на пластине. В последующих $2N$ строках задано по одному целому числу — длины сторон изображений s_n , отсортированные по невозрастанию: $(1 \leq N \leq 10^6, 1 \leq s_n \leq s_{n-1} \dots \leq s_1 \leq 10^6)$.

Формат выходного файла

Выведите одно число — требуемую минимальную площадь

Пример

<code>message.in</code>	<code>message.out</code>
3	20382
80	
79	
78	
6	
5	
4	

Задача E. Numbers

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

На листе клетчатой бумаги в n подряд идущих по горизонтали клетках записаны целые числа a_1, \dots, a_n , по одному числу в клетке. За одно действие разрешается зачёркивать k подряд идущих клеток. Какова максимальная сумма чисел, которые могут быть вычеркнуты за m действий?

Формат входного файла

В первой строке входного файла заданы три целых числа n , m и k ($1 \leq n \leq 3 \cdot 10^4$, $1 \leq m \leq 500$, $1 \leq k \leq n$) — изначальное количество чисел, количество действий и количество клеток, которые можно зачеркнуть за одно действие, соответственно. В последующих n строках заданы числа по одному на строку, причём в i -й строке задано i -е слева число. Все n чисел целые, положительные, при этом каждое из чисел меньше, чем 10^4 .

Формат выходного файла

В выходной файл выведите одно число — максимальную сумму чисел, которые могут быть зачёркнуты за m действий. Гарантируется, что значение данной суммы не превосходит 10^9 .

Пример

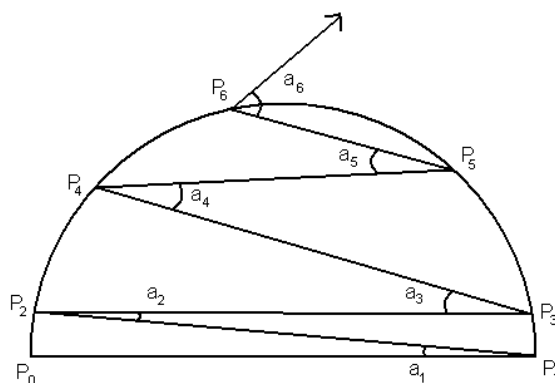
<code>numbers.in</code>	<code>numbers.out</code>
5 2 3 3 4 5 6 7	25
8 2 3 1 7 4 1 8 5 8 1	33

Задача F. Semicircle

Имя входного файла: `semicircle.in`
Имя выходного файла: `semicircle.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Задан полукруг радиуса 1 и последовательность угловых величин a_1, a_2, \dots, a_n . Пусть P_0 и P_1 — концы диаметра, на котором построен полукруг. Выберем точку P_2 на дуге полукруга так, чтобы угол $P_0P_1P_2$ был равен a_1 .

Точки P_i ($2 \leq i \leq n$) строятся следующим образом: пусть точки P_0, \dots, P_{k-1} уже построены. Тогда точку P_k Выберем на дуге полукруга так, чтобы угол $P_{k-2}P_{k-1}P_k$ был равен a_{k-1} . Если подобный выбор сделать невозможно, считается, что последовательность *покинула полукруг* в точке P_{k-1} .



Также вам задано вещественное число l . Вычислите, что произойдет раньше — длина некоторой хорды P_kP_{k+1} станет меньше или равной l или же последовательность покинет полукруг. Для каждого из этих случаев выведите значение k , после которого произошло соответствующее событие.

Формат входного файла

В первой строке входного файла задано целое число n — длина последовательности (a_i) ($1 \leq n \leq 1000$). Во второй строке заданы n вещественных чисел, i -е из которых задает значение угла a_i в градусах ($0 \leq a_i < 90$), в третьей — вещественное число l ($0 < l < 2.0$).

Гарантируется, что одно из событий (последовательность покинет полукруг или длина очередной хорды станет не более l) для заданной последовательности заведомо произойдет не позднее, чем после n -го шага.

Также гарантируется, что значения a_k и l подобраны таким образом, что любая хорда, которая длиннее l , превосходит l более, чем на 10^{-5} , и разность между текущим углом и минимальным углом, при котором последовательность покидает круг на данном шаге, превосходит 10^{-5} градусов.

Формат выходного файла

В случае, если последовательность покинет полукруг ранее, чем длина очередной хорды станет меньше или равной l , выведите в выходной файл строку “`Out after`” и затем, через пробел, номер точки, в которой последовательность покинула полукруг.

В случае, если длина очередной хорды станет меньше или равной l до того, как последовательность покинет полукруг, выведите в выходной файл строку “`Reached after`” и затем, через пробел, наименьший из номеров точек, которые соединяет соответствующая хорда.

Пример

semicircle.in	semicircle.out
13 2 4 6 8 10 12 14 16 18 20 22 24 26 1	Reached after 8
3 43 70 1.509758906676004 0.5347231675486146	Out after 2