

Задача А. Голосование

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

Компания TopRunner, занимающаяся проведением международных соревнований по лёгкой атлетике, объявила о сокращении программы соревнований (иначе называемых Simultaneously Running Masters, или SRM), вызванном экономическим кризисом.

Для того, чтобы определить, какое из N соревнований, запланированных на месяц, требуется оставить в календаре, TopRunner прибегла к голосованию. Всего в соревнованиях участвуют M участников. Каждый участник составил список из N соревнований, начинающийся с самого ценного, по его мнению, и заканчивающийся самым малоценным. Подсчёт ведётся следующим образом. Сначала определяется одно соревнование, сумма мест для которого максимальна, и удаляется из списка. В случае, если таких соревнований несколько, удаляется соревнование с наибольшим номером. Удалённое соревнование вычёркивается из всех списков, составленных участниками, после чего процесс повторяется до тех пор, пока не останется ровно одно соревнование.

Вам предстоит обработать результаты голосования и определить, соревнование с каким номером осталось.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. Первая строка каждого тестового примера содержат два числа G и N ($2 \leq N \leq 5$, $1 \leq G \leq 20$) — количество групп участников с совпадающими списками предпочтений, на которые делятся команды, и количество соревнований в данном месяце. Затем в G строках идёт описание группы, в формате $M_i a_{i1} a_{i2} \dots a_{iN}$, где $1 \leq M_i \leq 20$ — количество участников в группе, $a_{i1} \dots a_{iN}$ — перестановка из чисел от 1 до N , определяющая список предпочтений данной группы. Входной файл заканчивается примером с $G = N = 0$, который обрабатывать не нужно.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите номер оставшегося соревнования.

Пример

vote.in	vote.out
3 4	4
10 1 4 2 3	1
15 3 2 1 4	
12 4 3 2 1	
3 2	
10 1 2	
10 1 2	
20 2 1	
0 0	

Задача В. Фотографии победителей

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

Режим экономии компании TopRunner коснулся и печатной продукции, выпускаемой по итогам соревнований, организованных компанией. Если раньше фотографии победителей во всех m номинациях (бег на 100 метров, марафон, бег с пиццей, бег в упряжи, бег в колесе и так далее) публиковались на отдельных страницах, то сейчас принято решение все фотографии печатать на одной полосе.

Размер полосы — $w \times h$. Требуется разбить её на m прямоугольных частей ненулевой площади с целыми сторонами разрезами, параллельными сторонам полосы так, чтобы в результате площадь наибольшей из получившихся частей была минимальной. При этом разрезы проводятся последовательно, и каждый разрез делит какую-то одну из уже существующих на тот момент частей на две части ненулевой площади.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. Каждый тестовый пример начинается с новой строки, содержащей три целых числа w , h и m — соответственно длины сторон полосы и количество номинаций, $1 \leq w, h, m \leq 20$, а также $m \leq wh$. Входной файл заканчивается примером с $w = h = m = 0$, который обрабатывать не нужно.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите одно число — площадь наибольшей получившейся части при требуемом разрезании.

Пример

<code>pic.in</code>	<code>pic.out</code>
4 4 4	4
4 4 3	6
0 0 0	

Задача С. Оптимизация ников

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

Режим экономии коснулся и ников, под которыми на сайте TopRunner фигурировали участники соревнований. Было замечено, что ник лучше архивируется (и занимает меньше места на диске) если между любыми двумя одинаковыми буквами не может быть никакой третьей буквы (например, конфигурации типа «уагу» являются недопустимыми).

Ваша задача — написать программу, которая из заданного ника выкидывает наименьшее количество букв так, чтобы ни между какими двумя одинаковыми буквами не было никакой третьей буквы.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. Первая строка каждого примера содержат два числа M и K — длина ника и количество букв в алфавите, где $1 \leq M \leq 100$ и $1 \leq K \leq 5$. Во второй строке записаны M целых положительных чисел, не превосходящих K — буквы ника, заданные их номерами в алфавите. Входной файл заканчивается примером с $M = K = 0$, который обрабатывать не нужно.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите одно число — наименьшее число букв, которое необходимо убрать из ника, чтобы выполнялось требование задачи.

Пример

<code>nick.in</code>	<code>nick.out</code>
10 3	2
2 1 2 2 1 1 3 1 3 3	
0 0	

Задача D. Улучшение результатов

Имя входного файла: dog.in
Имя выходного файла: dog.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

В новых экономических условиях повышение уровня показанных на соревнованиях результатов может оказаться жизненно важным для сохранения спонсорской поддержки на прежнем уровне. В связи с этим руководитель группы спортивных психологов компании TopRunner предложил следующий вариант.

Обычная дистанция финального забега составляет 100 метров. Обычно каждый бегун пробегает за секунду r метров (или расстояние до финиша, если до финиша осталось менее r метров). Психолог предложил использовать собаку, лай которой подгонял бы бегунов. Однако это оказалось справедливо для F участников — в секунду, в которую они слышали лай, они пробежали $r + 2$ метра. S участников, наоборот, начинали бежать хуже и в ту секунду, когда лаяла собака, пробежали $r - 1$ метр. На N участников собачий лай никакого влияния не оказывал. При этом количество метров, пробегаемых любым участником за секунду, зависит только от наличия или отсутствия собачьего лая в эту секунду.

Требуется найти минимальное значение суммы времён, показанных на дистанции каждым участником. Участники стартуют одновременно, время каждого участника округляется вверх до целого количества секунд. Собака может лаять только в начале каждой секунды.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. Каждый пример содержит 4 целых числа F , S , N и r ($0 \leq F, S, N \leq 1000$, $1 \leq r \leq 100$). Файл завершается примером с $F = S = N = 0$, который обрабатывать не надо.

Формат выходного файла

Для каждого тестового примера в выходной файл выведите минимальное суммарное время.

Пример

dog.in	dog.out
3 1 1 2	188
1 3 0 2	200
0 0 0 0	

Задача E. Как увеличить свой рейтинг

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

Одной из революционных идей совета директоров компании TopRunner было сокращение не только предстоящих, но и прошлых соревнований. Таким образом недовольство участников можно будет через некоторое время погасить ссылками на то, что «всегда так было». Заодно была предложена более простая формула для рейтинга:

Для этого первоначально было решено провести опрос участников, какие из соревнований стоит «убрать» из прошлого. Появился способ улучшения рейтинга «задним числом», без участия в соревнованиях. В TopRunner для участника, выступавшего в n соревнованиях, рейтинг равен

$$R = 100 \times \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}$$

Здесь a_i — количество баллов, набранное участником, b_i — количество баллов, набранное победителем. При этом если участник за все турниры получил в сумме 0 баллов, его рейтинг равен нулю.

По заданным результатам соревнований для участника и результатам, показанным в этом же соревновании победителем, вычислите, какого максимального рейтинга можно добиться, если выкинуть ровно k соревнований.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров, каждый из которых состоит из трёх строк. В первой задано количество соревнований для данного участника n ($1 \leq n \leq 1000$) и количество сокращаемых соревнований k ($0 \leq k \leq n$). Во второй строке заданы n чисел a_i — результаты участника в i -м конкурсе, в третьей — n чисел b_i — результаты победителя i -го конкурса. При этом $0 \leq a_i \leq b_i \leq 10^9$. Входной файл заканчивается примером с $n = k = 0$, который обрабатывать не нужно.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите максимальный рейтинг, возможный после выбрасывания k соревнований, округлённый до ближайшего целого (гарантируется, что ни один ответ не находится ближе, чем на 10^{-3} к числам вида $(2n + 1)/2$).

Пример

<code>rate.in</code>	<code>rate.out</code>
3 1	83
5 0 2	100
5 1 6	
4 2	
1 2 7 9	
5 6 7 9	
0 0	

Задача F. Прошлые контесты

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

Так как мнения участников о том, какие прошлые SRM надо сокращать, ожидаемо не совпали, то при выборе соревнований, которые будут сокращены в прошлом, был принят следующий принцип. Изначально соревнования занумерованы подряд идущими натуральными числами, начиная с 1. Выкидываются все соревнования, в десятичной записи номера которых содержатся какие-либо две одинаковых цифры, после чего оставшиеся соревнования перенумеровываются.

Узнав про этот замысел руководства TopRunner, руководство независимого спортивного издания CheshireNews решило сохранить результаты всех SRM. В качестве дополнительного сервиса планируется возможность по заданному новому номеру n выдавать номер соревнования до сокращения.

Ваша задача — написать программу, реализующую данную возможность.

Формат входного файла

Входной файл состоит из нескольких тестовых примеров, каждый из которых записан на отдельной строке. Каждый тестовый пример состоит из числа n — номера соревнования по новой нумерации, $1 \leq n \leq 10^6$. Входной файл заканчивается примером с $n = 0$, который обрабатывать не нужно.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите номер соревнования согласно прежней нумерации.

Пример

<code>past.in</code>	<code>past.out</code>
25	27
10000	26057
0	