

Задача А. Баржа

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

На барже располагается K грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь не трогая остальных. Будем называть такие бочки крайними.

Изначально баржа пуста. Затем она последовательно проплывает через N доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более P бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.

Формат входных данных

В первой строке три целых числа N , K и P ($1 \leq N, K, P \leq 100$). Далее следует N строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «+ $A B$ », где A — номер отсека, в который помещается бочка, а B — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «- $A B$ », где A — номер отсека, из которого извлекается бочка, а B — номер ожидаемого вида топлива.

Формат выходных данных

Вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржей маршрута, либо вывести слово «Error» в противном случае.

Пример

ship.in	ship.out
6 1 2	2
+ 1 1	
+ 1 2	
- 1 2	
- 1 1	
+ 1 3	
- 1 3	

Задача В. Трёхцветная улица

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Жители Трёхцветной улицы решили раскрасить свои дома, согласно названию, в три цвета — красный, зелёный и синий, каждый в свой цвет. При этом из эстетических соображений жители не хотят, чтобы какие-либо два стоящих рядом дома были покрашены в один и тот же цвет. Для дома номер i стоящими рядом с ним считаются дома $i - 1$ и $i + 1$; первый и последний дома имеют по одному соседнему дому.

Недавно состоялось собрание жильцов всех домов этой улицы, на котором было выяснено, во сколько обойдётся покраска каждого дома в каждый из трёх возможных цветов. Теперь жильцы обратились за помощью к Вам, чтобы Вы, располагая этой информацией, выбрали, в какой цвет красить каждый дом так, чтобы учесть эстетические соображения жильцов и при этом заплатить за покраску как можно меньшую сумму.

Формат входных данных

В первой строке входного файла находится целое число N ($1 \leq N \leq 20$) — количество домов на Трёхцветной улице. В последующих N строках записано в каждой по три целых числа R_i G_i B_i через пробел — стоимость покраски i -го дома в красный, зелёный и синий цвета, соответственно. Известно, что $1 \leq R_i, G_i, B_i \leq 1000$.

Формат выходных данных

Выведите в первую строку выходного файла одно число — минимальную стоимость требуемой покраски.

Примеры

rgb.in	rgb.out
3 1 100 100 100 1 100 100 100 1	3
3 1 100 100 100 100 100 1 100 100	102
3 26 40 83 49 60 57 13 89 99	96
6 30 19 5 64 77 64 15 19 97 4 71 57 90 86 84 93 32 91	208
8 71 39 44 32 83 55 51 37 63 89 29 100 83 58 11 65 13 15 47 25 29 60 66 19	253

В первом примере дешевле всего будет покрасить первый дом в красный, второй — в зелёный и третий — в синий цвет.

Во втором примере первый и третий дома надо красить в красный цвет, а второй — в зелёный или синий.

Задача С. Таблица

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Рассмотрим прямоугольную таблицу n на m . Занумеруем строки таблицы числами от 1 до n , а столбцы — числами от 1 до m . Таблица последовательно заполняется числами. Обозначим через a_{ij} число, стоящее на пересечении i -ой строки и j -ого столбца. Первая строка таблицы заполняется заданными числами — $a_{11}, a_{12}, \dots, a_{1m}$. Затем заполняются строки с номерами от 2 до n . Число a_{ij} вычисляется как сумма всех чисел таблицы, находящихся в "треугольнике" над элементом a_{ij} . Все вычисления при этом выполняются по модулю r .

•	•	•	•	•	•	•
	•	•	•	•	•	•
		•	•	•	•	•
			•	•	•	
				a_{ij}		

Более точно, значение a_{ij} вычисляется по следующей формуле:

$$a_{i,j} = \left(\sum_{k=1}^{i-1} \sum_{\substack{t=j+k \\ 1 \leq t \leq m}} a_{i-k,t} \right) \bmod r$$

Например, если таблица состоит из трех строк и четырех столбцов, и первая строка состоит из чисел 2,3,4,5, а $r = 40$ то таблица выглядит следующим образом (взятие по модулю показано только там, где оно приводит к изменению числа):

2	3	4	5
$5 = 2 + 3$	$9 = 2+3+4$	$12 = 3+4+5$	$9 = 4 + 5$
$23 = 2+3+4+5+9$	$0 = (2+3+4+5+9+12) \bmod 40 = 40$	$4 = (2+3+4+5+9+12+9) \bmod 40 = 44$	$33 = 3+4+5+12+9$

Дана первая строка таблицы ($a_{11}, a_{12}, \dots, a_{1m}$), требуется вычислить последнюю строку. Поскольку числа в ответе могут быть достаточно большими, посчитайте ответ по модулю r .

Формат входных данных

Первая строка входного файла содержит числа n , m и r ($2 \leq n, m \leq 2000$, $2 \leq r \leq 10^9$) — число строк и столбцов таблицы соответственно, а так же число, по модулю которого надо посчитать ответ. Следующая строка содержит m целых чисел — первую строку таблицы: $a_{11}, a_{12}, \dots, a_{1m}$. Все a_{1i} неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в первой строке выходного файла m чисел — последнюю строку таблицы: $a_{n1}, a_{n2}, \dots, a_{nm}$.

Примеры

table.in	table.out
2 3 10 1 2 3	3 6 5
3 3 10 1 1 1	8 0 8
3 4 40 2 3 4 5	23 0 4 33

Система оценки и описание подзадач

Задача содержит три подзадачи. Баллы за каждую подзадачу начисляются только в том случае, если программа прошла все тесты из предыдущих подзадач.

Подзадача 1

Дополнительное ограничение $n, m \leq 50$. Подзадача оценивается в 40 баллов. Баллы за подзадачу начисляются, если все тесты данной подзадачи прошли.

Подзадача 2

Дополнительное ограничение $n, m \leq 300$. Подзадача оценивается в 20 баллов. Баллы за подзадачу начисляются, если все тесты данной подзадачи прошли.

Подзадача 3

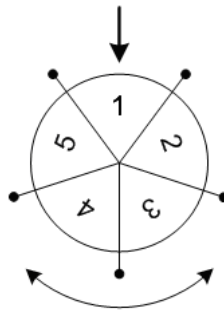
Дополнительные ограничения отсутствуют. Подзадача оценивается в 40 баллов. Тесты данной подзадачи оцениваются независимо.

Задача D. Колесо Фортуны

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 Мб

Развлекательный телеканал транслирует шоу "Колесо Фортуны". В процессе игры участники шоу крутят большое колесо, разделенное на сектора. В каждом секторе этого колеса записано число. После того как колесо останавливается, специальная стрелка указывает на один из секторов. Число в этом секторе определяет выигрыш игрока.

Юный участник шоу заметил, что колесо в процессе вращения замедляется из-за того, что стрелка задевает за выступы на колесе, находящиеся между секторами. Если колесо вращается с угловой скоростью v градусов в секунду, и стрелка, переходя из сектора X к следующему сектору, задевает за очередной выступ, то текущая угловая скорость движения колеса уменьшается на k градусов в секунду. При этом если $v \leq k$, то колесо не может преодолеть препятствие и останавливается. Стрелка в этом случае будет указывать на сектор X .



Юный участник шоу собирается вращать колесо. Зная порядок секторов на колесе, он хочет заставить колесо вращаться с такой начальной скоростью, чтобы после остановки колеса стрелка указала на как можно большее число. Колесо можно вращать в любом направлении и придавать ему начальную угловую скорость от a до b градусов в секунду.

Требуется написать программу, которая по заданному расположению чисел в секторах, минимальной и максимальной начальной угловой скорости вращения колеса и величине замедления колеса при переходе через границу секторов вычисляет максимальный выигрыш.

Формат входных данных

Первая строка входного файла содержит целое число n — количество секторов колеса ($3 \leq n \leq 100$).

Вторая строка входного файла содержит n положительных целых чисел, каждое из которых не превышает 1000 — числа, записанные в секторах колеса. Числа приведены в порядке следования секторов по часовой стрелке. Изначально стрелка указывает на первое число.

Третья строка содержит три целых числа: a , b и k ($1 \leq a \leq b \leq 10^9$, $1 \leq k \leq 10^9$).

Формат выходных данных

В выходном файле должно содержаться одно целое число — максимальный выигрыш.

Примеры

wheel.in	wheel.out
5 1 2 3 4 5 3 5 2	5
5 1 2 3 4 5 15 15 2	4
5 5 4 3 2 1 2 5 2	5

Замечание

В первом примере возможны следующие варианты: можно придать начальную скорость колесу равную 3 или 4, что приведет к тому, что стрелка преодолеет одну границу между секторами, или придать начальную скорость равную 5, что позволит стрелке преодолеть 2 границы между секторами. В первом варианте, если

закрутить колесо в одну сторону, то выигрыш получится равным 2, а если закрутить его в противоположную сторону, то — 5. Во втором варианте, если закрутить колесо в одну сторону, то выигрыш будет равным 3, а если в другую сторону, то — 4.

Во втором примере возможна только одна начальная скорость вращения колеса — 15 градусов в секунду. В этом случае при вращении колеса стрелка преодолеет семь границ между секторами. Тогда если его закрутить в одном направлении, то выигрыш составит 4, а если в противоположном направлении, то — 3.

Наконец, в третьем примере оптимальная начальная скорость вращения колеса равна 2 градусам в секунду. В этом случае стрелка вообще не сможет преодолеть границу между секторами, и выигрыш будет равен 5.

Система оценки и описание подзадач

Задача содержит две подзадачи. Подзадачи оцениваются независимо.

Подзадача 1

Дополнительное ограничение $1 \leq a \leq b \leq 1000$. Подзадача оценивается в 50 баллов. Баллы за подзадачу начисляются только в том случае, если все тесты этой подзадачи пройдены.

Подзадача 2

Дополнительные ограничения отсутствуют. Подзадача оценивается в 50 баллов. Тесты этой подзадачи оцениваются независимо.

Задача Е. Информатизация садоводства

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Дачный участок Степана Петровича имеет форму прямоугольника размером $a \times b$. На участке имеется n построек, причем основание каждой постройки — прямоугольник со сторонами, параллельными сторонам участка.

Вдохновленный успехами соседей, Степан Петрович хочет посадить на своем участке m видов плодовых культур (участок Степана Петровича находится в северной местности, поэтому $m = 1$ или $m = 2$). Для каждого вида растений Степан Петрович хочет выделить отдельную прямоугольную грядку со сторонами, параллельными сторонам участка. Само собой, грядки не могут занимать территорию, занятую постройками или другими грядками.

Степану Петровичу хочет расположить грядки таким образом, чтобы их суммарная площадь была максимальной. Грядки не должны пересекаться, но могут касаться друг друга.

Требуется по заданным размерам участка и координатам построек определить оптимальное расположение грядок.

Формат входных данных

В первой строке входного файла содержатся два целых числа n и m ($0 \leq n \leq 10$; $1 \leq m \leq 2$).

Во второй строке содержатся два целых числа a и b ($1 \leq a, b \leq 10000$).

Далее следуют n строк, каждая из которых содержит четыре целых числа $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$ — координаты двух противоположных углов постройки ($0 \leq x_{i,1} < x_{i,2} \leq a$, $0 \leq y_{i,1} < y_{i,2} \leq b$). Различные постройки не могут пересекаться, но могут касаться друг друга.

Формат выходных данных

В выходной файл выведите m строк, каждая из которых содержит координаты двух противоположных углов предлагаемой грядки. Координаты должны быть целые (всегда можно добиться максимальной суммарной площади грядок, располагая их в прямоугольниках с целыми координатами).

В случае, если в вашем решении Степану Петровичу следует посадить менее m грядок, выведите для грядок, которые не следует сажать, строку "0 0 0 0" (см. пример 2).

Примеры

garden.in	garden.out
2 2 7 5 4 2 6 4 0 1 2 2	0 2 4 5 2 0 7 2
3 2 4 4 0 0 4 1 0 1 1 4 3 1 4 4	1 1 3 4 0 0 0 0

Система оценки и описание подзадач

В этой задаче тесты оцениваются независимо. Решения, рассматривающие только случай $m = 1$, будут оцениваться из 50 баллов.