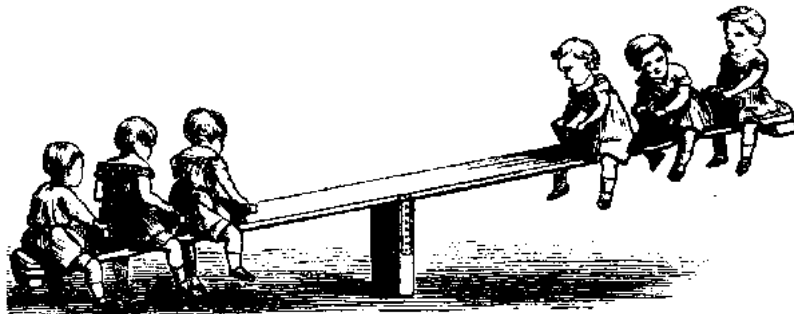


Задача 1. Качели

Ограничение по времени: 0.5 секунд

Трое друзей — Аня, Боря и Саш — пришли на детскую площадку, чтобы покататься на качелях-балансире. Качели представляют собой длинную балку, закреплённую в центре, на которую дети садятся с разных концов.



Массы детей равны A , B и C кг. Чтобы держать баланс на качелях, разница масс на двух концах качелей должна быть не более D кг. Друзьям повезло: рядом с площадкой оказалась груда достаточно тяжёлых камней. Один из детей может взять с собой любой камень, чтобы сделать разность масс на концах качелей допустимой. Помогите друзьям определить минимальную массу камня, благодаря которому они смогут покататься на качелях.

Формат входных данных

Программа получает на вход три числа A , B , C , записанных в отдельных строках, — массы друзей. В четвёртой строке записано число D — наибольшая допустимая разница масс на концах качелей. Все числа — целые, положительные и не превосходящие 10^9 .

Формат выходных данных

Программа должна вывести одно целое число — минимальную необходимую массу камня, которую нужно добавить на одну из сторон качелей, чтобы друзья смогли покататься на них, сев оптимально. Если камень им не понадобится, программа должна вывести число 0.

Система оценки

Решения, правильно работающие, когда все входные числа не превосходят 10^5 , будут оцениваться в 40 баллов.

Примеры

стандартный ввод	стандартный вывод
30 40 35 10	15
30 20 45 10	0

Замечание

В первом примере Аня и Саша сядут на одну сторону, их суммарная масса будет равна 65 кг. На другую сторону сядет Боря, взяв 15-килограммовый камень, тогда масса Бори с камнем составит 55 кг. Разница весов на концах качелей примет значение 10 кг.

Во втором примере Аня и Боря сядут на одну сторону (50 кг), Саша — на другую сторону (45 кг). Разница весов будет равна 5 кг, поэтому камень не понадобится.

Задача 2. Фонари

Ограничение по времени: 1 секунда

Вдоль прямой улицы на равном расстоянии располагаются N домов. Будем считать расстояние между домами за единицу длины.

Около каждого дома можно поставить один фонарь. Всего имеется A фонарей, которые могут освещать дома на расстоянии X (включительно), и B фонарей, которые могут освещать дома на расстоянии Y (включительно). В частности, при $X = 0$ или $Y = 0$ такой фонарь освещает только тот дом, у которого он установлен.

Вам необходимо расставить минимальное число фонарей так, чтобы все дома были освещены. Один дом может быть освещён несколькими фонарями. Освещать участки улицы между домами необязательно.

Формат входных данных

Первая строка входных данных содержит целое число N ($1 \leq N \leq 10^5$). Следующие четыре строки содержат целые неотрицательные числа A , X , B и Y соответственно, которые не превосходят 10^5 .

Формат выходных данных

Программа должна вывести столько строк, сколько фонарей необходимо установить. Каждая строка должна содержать два целых числа через пробел — координату фонаря и расстояние, которое он освещает (то есть одно из чисел X или Y). Координаты представляют из себя целые числа от 1 до N , рядом с каждым домом можно поставить только один фонарь.

При наличии нескольких правильных ответов можно вывести любой из них. Если ответа не существует, программа должна вывести одно число -1 .

Система оценки

Решения, правильно работающие при $A = 0$ или $B = 0$, будут оцениваться в 30 баллов.

Решения, правильно работающие при $A \neq 0$, $B \neq 0$, $n \leq 1000$, будут оцениваться в 40 баллов.

Примеры

стандартный ввод	стандартный вывод
10 3 1 1 2	2 1 5 2 9 1
10 1 1 1 2	-1

Замечание

В ответе к первому примеру фонарь у дома 2 освещает также дома 1 и 3, фонарь у дома 5 — также дома 3, 4, 6 и 7, а фонарь у дома 9 — также дома 8 и 10. В результате все дома освещены. Во втором примере фонарей недостаточно.

Задача 3. Красная Шапочка на болоте

Ограничение по времени: 1 секунда

Красная Шапочка отправилась на болото для сбора клюквы, чтобы испечь пирожки для бабушки. Клюквенное болото представляет собой координатную прямую. Берег, на котором стоит девочка, имеет координату 0, а клюквенная поляна — координату $N + 1$. В точках с координатами $1, 2, \dots, N$ расположены кочки. Первоначально у девочки E единиц энергии. Красная Шапочка может прыгнуть из точки x в точку y ($x < y$), потратив на это $(y - x)$ единиц энергии, то есть затраченная энергия равна расстоянию между кочками. После того, как девочка приземлится на кочке с координатой i , она получает a_i единиц энергии (при этом значение a_i может оказаться отрицательным, тогда энергия Красной Шапочки уменьшится при приземлении). Нельзя, чтобы энергия Красной Шапочки в какой-либо момент оказалась меньше нуля. Например, Красная Шапочка не может прыгнуть с кочки 1 на кочку 3, имея одну единицу энергии, вне зависимости от того, сколько энергии она получит на 3-й кочке, так как для осуществления такого прыжка необходимо две единицы энергии.

Так как Красной Шапочке ещё надо вернуться обратно, девочке интересно, какое максимальное количество энергии у неё может оказаться, когда она достигнет поляны (точки с координатой $N + 1$).

Формат входных данных

Первая строка входных данных содержит целое число E — первоначальный запас энергии Красной Шапочки, $1 \leq E \leq 10^9$.

Вторая строка входных данных содержит целое число N — количество кочек на болоте, $1 \leq N \leq 10^5$.

Следующие N строк содержат по одному целому числу a_i — энергия, которую получает Красная Шапочка на i -й кочке, $-2000 \leq a_i \leq 2000$.

Формат выходных данных

Программа должна вывести одно число — максимальное количество единиц энергии, которое останется у Красной Шапочки после достижения клюквенной поляны. Если девочка не сможет достигнуть цели, выведите одно число «-1» (без кавычек).

Система оценки

Решения, правильно работающие при $N \leq 15$, будут оцениваться в 20 баллов.

Решения, правильно работающие при $N \leq 900$, будут оцениваться в 70 баллов.

Решения, правильно работающие, когда все $a_i \geq 0$, будут набирать не менее 20 баллов.

Примеры

стандартный ввод	стандартный вывод
2 3 1 -1 1	0
1 4 -1 100 -1 -1	-1

Замечание

В первом примере три кочки и первоначально 2 единицы энергии у Красной Шапочки. Она прыгает на кочку 1, что требует 1 единицу энергии, и у неё остаётся 1 единица энергии. На кочке 1 девочка получает 1 единицу энергии, и у неё становится 2 единицы энергии. Затем она прыгает

с кочки 1 на кочку 3, потратив 2 единицы энергии, и у неё становится 0 энергии. Приземлившись на кочку 3, Красная Шапочка получает 1 единицу энергии, этого достаточно, чтобы перепрыгнуть с кочки 3 на поляну в точке 4, после чего у Красной Шапочки останется 0 единиц энергии.

Во втором примере у Красной Шапочки первоначально только 1 единица энергии, поэтому она может прыгнуть только на кочку 1, но значение $a_1 = -1$, то есть после приземления на кочку 1 у Красной Шапочки энергия станет отрицательной, и она не сможет продолжить свой путь.

Задача 4. Деление шоколадки

Ограничение по времени: 1 секунда

У Маши есть прямоугольная шоколадка, состоящая из $m \times n$ квадратных долек. Маша хочет разделить эту шоколадку между своими друзьями, разломив шоколадку по линиям на k кусочков, то есть каждому другу достанется прямоугольный кусочек шоколадки.

У Юры сегодня день рождения, поэтому Маша хочет разделить шоколадку так, чтобы Юре достался самый большой кусок (содержащий как можно больше долек). Определите число долек в этом куске.



Формат входных данных

Программа получает на вход три натуральных числа, каждое в отдельной строке: m , n и k . Все числа — целые положительные, при этом m и n не превосходят 10^6 , а $k \leq mn$.

Обратите внимание на то, что значение mn , а, значит, и значение k в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

Формат выходных данных

Программа должна вывести одно целое число — максимально возможное количество долек в том прямоугольном куске, который получит Юра.

Система оценки

Решения, правильно работающие при $m \leq 1000$ и $n \leq 1000$, будут оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
4 5 4	16

Замечание

В примере из условия нужно разделить шоколадку 4×5 на 4 кусочка. Самый большой кусочек будет состоять из 16 долек, как показано на картинке.

Задача 5. Кодовый замок

Ограничение по времени: 1 секунда

В разведывательное управление доставили сейф с секретной информацией, кодовый замок на котором открывается комбинацией из n цифр, каждая цифра может принимать b различных значений от 0 до $b - 1$. Код неизвестен, однако разведчики передали несколько донесений о том, что сумма цифр кода в некоторых заданных позициях равна какому-то известному числу. Используя информацию из всех полученных донесений, определите, сколько существует возможных кодов, удовлетворяющих этим условиям.

Формат входных данных

Первая строка входных данных содержит число b — количество различных значений одной цифры кода, $2 \leq b \leq 10$. Вторая строка содержит число n — количество цифр в коде, $n \geq 1$, $b^n \leq 60\,000$. Третья строка содержит число t — количество имеющихся донесений о сумме каких-то цифр кода, $t \geq 1$.

Следующие $2t$ строк содержат информацию об имеющихся донесениях. Каждое донесение состоит из двух строк. Первая из этих строк («маска цифр») содержит n символов, записанных слитно и равных «0» или «1», где цифра «1» обозначает, что в донесении говорится об этой цифре кода. Например, маска цифр «01011» означает сумму цифр, стоящих в коде на 2-й, 4-й и 5-й позициях. Во второй строке донесения записано число s , равное сумме цифр кода, стоящих на данных позициях.

Гарантируется, что каждая маска цифр содержит хотя бы одну единицу и что все маски цифр различаются. Общее число донесений может быть любым, удовлетворяющим этим условиям.

Формат выходных данных

Программа должна вывести одно целое число — количество различных кодов, которые удовлетворяют всем донесениям.

Система оценки

Решения, правильно работающие, когда $n \leq 4$ и каждая маска цифр содержит ровно один символ «1», будут оцениваться в 28 баллов.

Решения, правильно работающие, когда $n \leq 4$, будут оцениваться в 64 балла.

Пример

стандартный ввод	стандартный вывод
8	3
3	
2	
110	
7	
011	
12	

Замечание

В примере из условия каждая цифра кода может принимать 8 различных значений от 0 до 7, код состоит из 3 цифр. Получены 2 донесения, из первого донесения известно, что сумма первой и второй цифры кода равна 7, из второго донесения известно, что сумма второй и третьей цифр кода равна 12. Существуют 3 кода, удовлетворяющие этим условиям: «075», «166», «257».