

Задача А. Сад

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	32 мегабайта

Петя — владелец самого красивого сада города Новосадска. Он посадил n роз у себя в саду. Пришло лето, цветы выросли большими и красивыми. Петя понял, что ему не хватает сил заботиться обо всех розах в своём саду. Поэтому он решил нанять двух садовников себе в помощь. Каждому из них Петя собирается выделить прямоугольный участок, за всеми розами внутри которого садовник и будет ухаживать. Участки должны быть непересекающимися и каждый должен содержать ровно k роз.

Петя хочет построить забор вокруг каждого участка, но денег у него на это немного. Помогите ему выбрать такие два участка, что суммарная длина забора, который будет необходимо построить, будет как можно меньше.

Сад представляет собой прямоугольник l метров в длину и w в ширину. Он разделён на $l \times w$ квадратов 1×1 метр. Зафиксируем координатную систему с осями, параллельными сторонам сада. У каждого квадрата будут целые координаты (x, y) , удовлетворяющие ограничениям $1 \leq x \leq l$, $1 \leq y \leq w$. В каждом квадрате может быть любое число роз.

Участки, которые нужно выбрать, должны иметь стороны, параллельные осям координат. Координаты углов должны быть целыми. Если $1 \leq l_1 \leq l_2 \leq l$ и $1 \leq w_1 \leq w_2 \leq w$, прямоугольный участок с углами (l_1, w_1) , (l_1, w_2) , (l_2, w_2) , (l_2, w_1) :

- содержит все квадраты с координатами (x, y) : $l_1 \leq x \leq l_2$ и $w_1 \leq y \leq w_2$ и
- имеет периметр $2 \cdot (l_2 - l_1 + 1) + 2 \cdot (w_2 - w_1 + 1)$.

Два участка должны быть непересекающимися, то есть у них не должно быть ни одного общего квадрата. Если у них есть общая сторона, либо часть стороны, эта часть всё равно должна быть отделена двумя различными заборами.

Напишите программу, которая найдёт пару непересекающихся участков, каждый из которых содержит ровно k роз, с минимальной суммой периметров.

Формат входных данных

В первой строке записано два целых числа l и w ($1 \leq l, w \leq 250$) — длина и ширина сада. Во второй строке записано два целых числа n и k ($2 \leq n \leq 5000$, $1 \leq k \leq \frac{n}{2}$) — количество роз в саду и количество роз, которое должно быть на каждом из участков.

В следующих n строках записано по два целых числа l_i и w_i ($1 \leq l_i \leq l$, $1 \leq w_i \leq w$) — координаты квадрата, содержащего i -ю розу. В одном квадрате может быть одновременно произвольное число роз.

Формат выходных данных

Выведите единственное целое число — минимальную сумму периметров двух прямоугольных участков. Если же выбрать такие участки невозможно, выведите вместо него строку «NO».

Пример

стандартный ввод	стандартный вывод
6 5 7 3 3 4 3 3 6 1 1 1 5 5 5 5 3 1	22

Система оценки

В половине тестов $l, w \leq 40$.

Задача В. Средняя последовательность

Имя входного файла: **стандартный ввод**
Имя выходного файла: **стандартный вывод**
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим неубывающую последовательность $s_1, \dots, s_{n+1}$ ($s_i \leq s_{i+1}$). Последовательность $m_1, \dots, m_n$, в которой каждый член определен как $m_i = \frac{s_i + s_{i+1}}{2}$, назовем *средней последовательностью* для последовательности $s_1, \dots, s_{n+1}$. Например, средняя последовательность для последовательности 1, 2, 2, 4 есть 1.5, 2, 3. Заметим, что элементы средней последовательности могут быть дробными числами. Тем не менее, в данной задаче используются только те средние последовательности, в которых все числа целые. Для заданной неубывающей последовательности из n целых чисел $m_1, \dots, m_n$ необходимо вычислить количество всех неубывающих последовательностей из $n + 1$ целых чисел $s_1, \dots, s_{n+1}$, для которых заданная последовательность $m_1, \dots, m_n$ является средней последовательностью.

Напишите программу, которая:

- читает из стандартного ввода неубывающую последовательность целых чисел;
- вычисляет количество всех неубывающих последовательностей целых чисел, для которых заданная последовательность является средней последовательностью;
- выводит ответ в стандартный вывод.

Формат входных данных

Первая строка стандартного ввода содержит одно целое число n ($2 \leq n \leq 5\,000\,000$). Оставшиеся n строк содержат значения последовательности $m_1, \dots, m_n$. Строка $i + 1$ содержит одно целое число m_i ($0 \leq m_i \leq 10\,000\,000\,000$).

Формат выходных данных

Ваша программа должна вывести в стандартный вывод ровно одно целое число — количество всех неубывающих последовательностей целых чисел, для которых входная последовательность является средней последовательностью.

Пример

стандартный ввод	стандартный вывод
3 2 5 9	4

Задача С. Горы

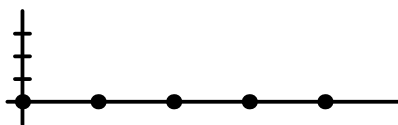
Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В парке развлечений «Ай-ой-ай» открылся новейший аттракцион: польские горки. Трек состоит из n рельс, присоединённых одна к концу другой. Начало первой рельсы находится на высоте 0. Оператор Петя может конфигурировать аттракцион, изменяя по своему желанию подъём нескольких последовательных рельс. При этом подъём всех остальных рельс не изменяется. При каждом изменении конфигурации рельс положение следующих за изменяемыми подбирается таким образом, чтобы весь трек оставался связным.

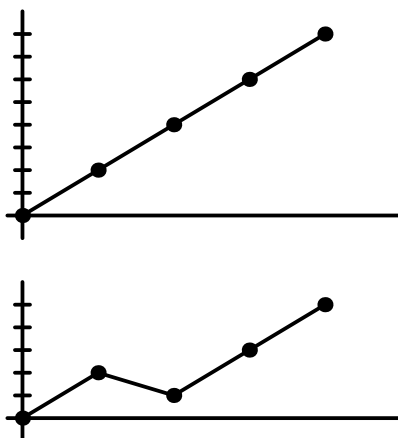
Каждый запуск вагонетки осуществляется с энергией, достаточной для достижения высоты h . Это значит, что вагонетка будет двигаться до тех пор, пока высота не превысит h , либо пока не закончится трек.

По записям о всех изменениях конфигурации рельс и временах запусков вагонетки для каждого запуска определите, сколько рельс вагонетка проедет до остановки.

Трек можно представить как последовательность n подъёмов d_i , по одному на рельс. Изначально все рельсы горизонтальны, то есть $d_i = 0$ для всех i .



Каждое изменение конфигурации определяется числами a , b и D : все рельсы с a -й по b -ю включительно после этого действия имеют подъём D .



Каждый запуск вагонетки определяется единственным целым числом h — максимальной высотой, на которую способна подняться вагонетка.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 10^9$) — число рельс. Следующие строки содержат запросы трёх видов:

- $I \ a \ b \ D$ — изменение конфигурации. Рельсы с a -й по b -ю включительно после выполнения запроса имеют подъём, равный D .

- Q h — запуск вагонетки. Требуется найти число рельс, которое проедет вагонетка, которая способна подняться на высоту h .
- E — конец ввода. Этот запрос встретится ровно один раз в конце файла.

В любой момент высота любой точки трека лежит от 0 до 10^9 . Во вводе не более 100 000 строк.

Формат выходных данных

Для каждого запроса Q выведите единственное целое число — количество рельс, которое проедет вагонетка.

Пример

стандартный ввод	стандартный вывод
4	4
Q 1	1
I 1 4 2	0
Q 3	3
Q 1	
I 2 2 -1	
Q 3	
E	

Задача D. День рождения

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Пети сегодня день рождения. На праздник пришло n детей, включая Петю. Их пронумеровали натуральными числами от 1 до n . Вокруг большого круглого стола расставили n стульев. Дети приходят по одному и рассаживаются вокруг стола: первый садится на первое место, второй — на следующее слева, и так далее. Наконец, ребёнок под номером n занимает последнее место между первым и $(n - 1)$ -м.

Родители Пети хорошо знают всех детей и знают, что некоторые дети будут шуметь слишком сильно, если их посадить рядом. Поэтому они решили пересадить детей особым образом. Нужный порядок можно описать перестановкой $p_1, p_2, \dots, p_n$: ребёнок p_1 должен будет сесть между p_n и p_2 , ребёнок p_2 — между p_1 и p_3 , ..., ребёнок p_n — между p_{n-1} и p_1 . Заметьте, что ребёнок p_1 может сесть как слева, так и справа от ребёнка p_n .

Чтобы пересадить детей нужным образом, родителям нужно будет пересаживать детей вправо и влево на несколько мест. Для каждого ребёнка им нужно выбрать направление перемещения (вправо или влево) и расстояние (число мест). По сигналу дети одновременно встанут со своих мест, перейдут к нужным местам и займут их.

Пересаживание неизбежно повлечёт за собой беспорядок. Назовём значением беспорядка максимальное число мест, на которое придётся переместиться какому-то из детей. Помогите найти такой способ пересадить всех детей, чтобы значение беспорядка было минимальным.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 1\,000\,000$) — число детей. Во второй строке записано n различных натуральных чисел $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) — перестановка детей.

Формат выходных данных

Выведите единственное целое число — наименьший возможный беспорядок.

Пример

стандартный ввод	стандартный вывод
6 3 4 5 1 2 6	2

Система оценки

В половине тестов $n \leq 1000$.

Задача Е. Игра с прямоугольником

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Рассмотрим игру для двух игроков. Игрокам дан прямоугольник размером $x \times y$ (где x и y — положительные целые числа). Игроки ходят по очереди. Ход состоит в разделении прямоугольника на два прямоугольника одним горизонтальным или вертикальным разрезом. Полученные прямоугольники должны иметь положительные целочисленные размеры.

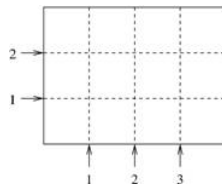


Рис. 1: Возможные разрезы прямоугольника размером 4×3 .

После каждого разреза прямоугольник с меньшей площадью убирается, а оставшийся передается другому игроку. Если прямоугольник разделен на две равные части, то убирается одна из них. Игрок, который получает прямоугольник размером 1×1 , проигрывает, так как не может сделать следующий ход.

Ваша задача — написать программу, которая бы играла в игру с прямоугольниками и выигрывала. Начальные размеры прямоугольника — целые числа от 1 до 100 000 000. Как минимум один из размеров больше 1. К тому же, в 50% тестов размеры прямоугольника не будут превышать 25.

Формат входных данных

Это интерактивная задача. Перед первым ходом и после каждого хода соперника вашей программе на стандартный ввод будет подаваться два числа — текущие размеры прямоугольника. Когда игра будет закончена на вход вашей программе будет передано два числа -1.

Формат выходных данных

В ответ на размеры доски вы должны вывести свой ход. Описание хода состоит из символа «H» или «V» (без кавычек) и номера горизонтального или вертикального разреза соответственно.

После каждого хода ваша программа должна делать операцию `flush` для очистки буфера. Для очистки буфера необходимо выполнить команду `flush(output)` в Pascal, `fflush(stdout)` в C, `cout.flush()` в C++.

Пример

стандартный ввод	стандартный вывод
4 3	V 1
3 2	H 1
2 1	V 1
-1 -1	

Задача F. Реки

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В одном уголке земного шара раскинулась живописная страна, покрытая лесами и реками. Маленькие речушки сливаются в большие, те — в ещё большие, и в конце концов все реки сливаются в одну большую реку, впадающую в море неподалёку от городка.

А ещё в этой стране есть n поселений лесорубов. Каждое поселение расположено около какой-то реки. Лесорубы валят лес, а затем сплавляют его по воде до городка. В городке находится лесопилка.

Чтобы сократить расходы на транспортировку леса по воде, решено было выбрать k поселений лесорубов и построить дополнительные лесопилки прямо у этих поселений. Из остальных поселений лес будет сплавляться вниз по течению до первой же лесопилки, будь то старая лесопилка в городке или одна из k новых. Лес, обработанный в лесопилке, сплавлять дальше по воде не требуется.

Ни одна река не разветвляется, так что из любого поселения существует единственный путь вниз по течению до городка.

Для каждого поселения лесорубов известно, сколько деревьев валят жители этого поселения за год. Требуется поставить новые лесопилки так, чтобы ежегодные расходы на сплав леса оказались как можно меньше. Известно, что транспортировка одного дерева на один километр по любому отрезку речной сети обходится в одну монету. Напишите программу, которая по заданному описанию речной сети, а также числу k вычисляет, каковы будут минимальные ежегодные расходы на сплав леса в монетах.

Формат входных данных

В первой строке ввода содержится два целых числа n и k — количество поселений лесорубов и количество новых лесопилок, которые можно построить ($2 \leq n \leq 100$, $1 \leq k \leq 50$, $k \leq n$). Следующие n строк описывают деревни; i -я из этих строк содержит три целых числа w_i , v_i и d_i — количество деревьев, ежегодно срубаемых жителями i -го поселения, номер ближайшего населённого пункта вниз по течению и расстояние до него ($0 \leq w_i \leq 10\,000$, $0 \leq v_i \leq n$, $1 \leq d_i \leq 10\,000$). Поселения пронумерованы целыми числами, начиная с единицы, в том порядке, в котором они заданы во входных данных; городок обозначается числом 0.

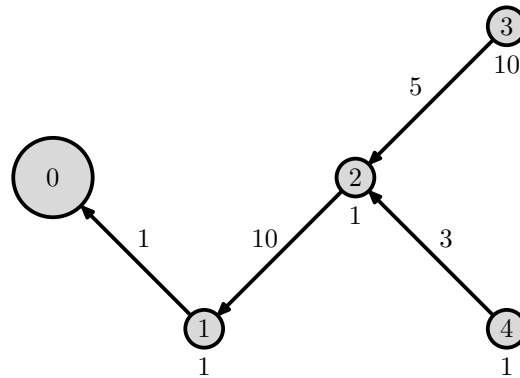
Гарантируется, что до постройки дополнительных лесопилок ежегодные расходы на сплав составляли не более 2 000 000 000 монет.

Формат выходных данных

В первой строке выведите одно число — минимальные ежегодные расходы на сплав леса в монетах.

Пример

стандартный ввод	стандартный вывод
4 2 1 0 1 1 1 10 10 2 5 1 2 3	4



На картинке изображена речная сеть из примера. Числа внутри кругов — номера поселений. Числа под кругами — количества деревьев, ежегодно срубаемых в поселениях. Числа над стрелками — длины рек.

Оптимальное решение — построить дополнительные лесопилки в поселениях лесорубов 2 и 3.

Система оценки

В половине тестов $n \leq 20$.