

Задача 1. A+B

<i>Ввод</i>	<code>aplusb.in</code> или с клавиатуры
<i>Вывод</i>	<code>aplusb.out</code> или на экран
<i>Ограничение по времени</i>	1 с
<i>Ограничение по памяти</i>	256 Мб
<i>Максимальный балл за задачу</i>	100

Напишите программу, которая складывает два числа.

Формат входных данных

Во входном файле находятся два числа, A и B . Числа целые и не превосходят по модулю 1000.

Формат выходных данных

Выведите в выходной файл одно число — сумму двух данных чисел.

Пример

<i>Ввод</i>	<i>Вывод</i>
2 2	4
-4 2	-2

Задача 2. Строгий Город

<i>Ввод</i>	<code>dist.in</code> или с клавиатуры
<i>Вывод</i>	<code>dist.out</code> или на экран
<i>Ограничение по времени</i>	2 с
<i>Ограничение по памяти</i>	256 Мб
<i>Максимальный балл за задачу</i>	100

Однажды Вася настолько устал, решая задачи по геометрии, что уснул прямо за тетрадкой в процессе решения очередной задачи. Ему приснилось, что он попал в город Строгий, в котором всё подчинялось строгим законам. В городе была введена система координат, а все дороги были либо строго вертикальными (параллельными оси ОУ), либо строго горизонтальными (параллельными оси ОХ). Точки пересечения вертикальных дорог с горизонтальными дорогами образовывали перекрестки. Все дороги в Строгом односторонние — перемещаться по каждой из них можно лишь в одном направлении, а перемещаться не по дорогам строго запрещено.

Во сне перед Васей постоянно вставала задача — насколько быстро можно добраться из одного перекрестка города в другой и можно ли это сделать вообще? На каждую такую задачу Вася должен был дать точный ответ, в противном случае он был бы строго наказан. Помогите Васе! По конфигурации дорог города для каждого запроса нужно определить наименьшее расстояние, которое придется преодолеть, чтобы попасть из начального перекрестка в конечный, не нарушая правил перемещения по городу.

Формат входных данных

В первой строке заданы три целых числа — V , H , Q — количество вертикальных дорог в городе, количество горизонтальных дорог в городе и количество запросов ($1 \leq V, H, Q \leq 50\,000$).

В каждой из следующих V строк описывается одна вертикальная дорога. Дорога задается двумя целыми числами X_i и T_i — координатой пересечения с осью ОХ и направлением ($-10^9 \leq X_i \leq 10^9$, $T_i = 1$ или $T_i = -1$). Если T_i равно 1, то по дороге можно двигаться в сторону увеличения координаты, если T_i равно -1 — в сторону уменьшения. Все X_i различны и указаны строго в порядке возрастания.

В следующих H строках описаны горизонтальные дороги в том же формате. Аналогично, горизонтальные дороги перечислены в порядке возрастания Y_i — координаты пересечения с осью ОУ.

В каждой из следующих Q строк содержится по одному запросу. Запрос задается четырьмя целыми числами X_a , Y_a , X_b , Y_b — координатами начальной точки и конечной точки соответственно. Гарантируется, что как начальная, так и конечная точки являются перекрестками, то есть лежат на пересечении вертикальной и горизонтальной дорог.

Формат выходных данных

Для каждого запроса необходимо вывести единственное число — минимальное расстояние, которое придется преодолеть по дорогам города, чтобы попасть из начальной точки в конечную. Если этого сделать невозможно, не нарушая правил движения, нужно вывести -1 . Ответы на запросы следует разделять пробелами или переводами строк.

Система оценивания

В тестах общей стоимостью 30 баллов координаты дорог не будут превосходить 50 по абсолютному значению, количество запросов также не будет превышать 50.

В тестах общей стоимостью 60 баллов количество вертикальных дорог, горизонтальных дорог и количество запросов не будет превышать 50.

Пример

<i>Ввод</i>	<i>Вывод</i>
2 2 4	1
0 -1	2
2 -1	3
0 -1	-1
1 1	
2 1 2 0	
2 0 0 0	
2 1 0 0	
0 0 2 1	

Задача 3. Выборы

<i>Ввод</i>	<code>elect.in</code> или с клавиатуры
<i>Вывод</i>	<code>elect.out</code> или на экран
<i>Ограничение по времени</i>	1 с
<i>Ограничение по памяти</i>	256 Мб
<i>Максимальный балл за задачу</i>	100

В Берляндии регулярно проходят выборы президента. На любых выборах в Берляндии бывают фальсификации. И вот, на очередных выборах так сложилось, что все проголосовали против нужного вам кандидата. Вам нужно обеспечить победу вашего кандидата, но проблема в том, что слишком явные фальсификации вызывают народное недовольство.

Всего в Берляндии n избирательных участков, на i -м из них k_i избирателей, из них проголосовало a_i (все против). Вам доступно два типа фальсификаций:

- Вброс голосов за пенсионеров. На i -м участке можно сделать максимум b_i таких вбросов, каждый из них увеличивает количество голосов за нужного вам кандидата на 1, но увеличивает народное недовольство на x .
- Подмена голосов. На i -м участке можно сделать максимум a_i подмен. Каждая подмена увеличивает количество голосов за нужного вам кандидата на 1, при этом, если на участке было сделано всего y подмен, народное недовольство увеличивается на y^2 .

Кандидат считается победителем, если процент голосов за него строго больше половины от общего количества избирателей (не только проголосовавших, но всех вообще).

Определите, можете ли вы обеспечить победу своего кандидата, и если да, то с каким минимальным народным недовольством вы можете это сделать.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leq n \leq 10^5$) — количество избирательных участков.

Вторая строка содержит n целых чисел k_1, k_2, \dots, k_n — общее количество избирателей на каждом участке ($0 \leq k_i \leq 10^4$).

Третья строка содержит n целых чисел a_1, a_2, \dots, a_n — количество проголосовавших на каждом участке ($0 \leq a_i \leq 10^4$).

Четвертая строка содержит n целых чисел b_1, b_2, \dots, b_n — максимально возможное количество фальсификаций первого типа на каждом участке ($0 \leq b_i \leq 10^4$).

Пятая строка содержит единственное целое число x — увеличение недовольства при совершении одной фальсификации первого типа ($1 \leq x \leq 10^8$).

Гарантируется, что $a_i + b_i \leq k_i$ для всех i .

Формат выходных данных

Выведите единственное целое число — минимально возможное недовольство, необходимое для победы вашего кандидата, или -1 , если это невозможно.

Пример

<i>Ввод</i>	<i>Вывод</i>
3 14 9 8 10 5 3 4 4 5 4	52
2 20 10 9 4 1 0 5	-1

Задача 4. Львы

<i>Ввод</i>	<code>lions.in</code> или с клавиатуры
<i>Вывод</i>	<code>lions.out</code> или на экран
<i>Ограничение по времени</i>	2 с
<i>Ограничение по памяти</i>	256 Мб
<i>Максимальный балл за задачу</i>	100

Иван — профессиональный фотограф. В погоне за хорошим кадром он готов ехать в любую точку мира. И сейчас он приехал в Уганду с целью сфотографировать львов в их естественной среде обитания. К сожалению, львов не очень заинтересовали профессиональные качества Ивана, зато их весьма заинтересовали его вкусовые качества.

Для простоты будем считать саванну бесконечной плоскостью, а Ивана и львов — точками. В начальный момент времени Иван выбирает направление, в котором он будет ехать, спасаясь от львов, и дальше все время едет по прямой. Львы, будучи великолепными охотниками, сразу определяют направление, выбранное Иваном, и планируют его поимку соответственно. В частности, львы понимают, что Иван поедет по прямой. Скорости всех львов постоянны и равны между собой. Как мы уже увидели, Иван не очень умный, поэтому фотографировать львов он отправился в медленном автомобиле, скорость которого также постоянна и равна скорости львов.

Помогите Ивану выбрать такое направление, чтобы его не догнал ни один лев.

Формат входных данных

В первой строке входных данных дано целое число n — количество львов ($1 \leq n \leq 5 \cdot 10^5$).

Во второй строке даны два целых числа — координаты Ивана.

Далее следуют n строк, в каждой из которых даны два целых числа — координаты львов.

Все координаты не превосходят 10^8 по абсолютной величине. Гарантируется, что все точки во входных данных различны.

Формат выходных данных

Если Иван не может убежать, выведите одно слово «NO» (без кавычек).

Иначе в первой строке выведите одно слово «YES» (тоже без кавычек), а во второй строке — координаты точки, по направлению к которой нужно ехать. Иван будет ехать по лучу, соединяющему его начальное местоположение и указанную точку; в частности, когда он доедет до этой точки, он без остановки будет двигаться дальше по прямой.

Выведенная точка не должна совпадать с начальным местоположением Ивана, а ее координаты должны быть целыми и не должны превышать 10^9 по абсолютной величине. Можно показать, что в случае существования ответа существует ответ в заданном формате. Если существует несколько подходящих точек, выведите любую из них.

Система оценивания

В тестах суммарной стоимостью не менее 37 баллов будет выполняться ограничение $n \leq 5000$.

Пример

<i>Ввод</i>	<i>Вывод</i>
2 0 0 1 1 2 2	YES -1 -1
4 0 0 1 1 -1 -1 1 -1 -1 1	NO

Задача 5. Запасы воды

Ввод	equal.in или с клавиатуры
Вывод	equal.out или на экран
Ограничение по времени	1 с
Ограничение по памяти	256 Мб
Максимальный балл за задачу	100

Крот Аркадий построил у себя в огороде систему подземных бункеров на случай внезапного нападения орлов. Система состоит из n бункеров, соединенных $n - 1$ тоннелями; все бункеры находятся на разной глубине: чем меньше номер, тем ближе к поверхности земли бункер. Бункер с номером 1 находится у самой поверхности земли, а любой другой бункер соединен тоннелем ровно с одним бункером выше.

В каждом бункере, кроме всего прочего, есть некоторый запас воды, сейчас в бункере i находится l_i литров воды. Аркадий хочет сделать так, чтобы во всех бункерах было поровну воды. Для этого он может в любой момент времени перелить из любого бункера a в любой другой бункер b любое количество воды, но только если бункеры a и b соединены тоннелем и бункер a находится выше бункера b .

Аркадий скоро осознал, что не всегда возможно распределить воду поровну, поэтому он разрешил себе доливать в любой бункер воду, принесенную из дома. Определите, какое минимальное количество воды нужно принести Аркадию из дома, чтобы можно было равномерно распределить воду по бункерам.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — количество бункеров.

Вторая строка содержит $n - 1$ целое число p_2, p_3, \dots, p_n ($1 \leq p_i < i$), где p_i означает, что бункер i соединен тоннелем с бункером p_i , лежащим выше.

Третья строка содержит n целых чисел l_1, l_2, \dots, l_n ($0 \leq l_i \leq 10^6$) — текущий объем воды в каждом из бункеров.

Формат выходных данных

Выведите одно число — минимальный объем воды, которую нужно принести Аркадию из дома, чтобы распределить воду поровну между всеми бункерами.

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-6} .

Формально, пусть ваш ответ равен a , а ответ жюри равен b . Ваш ответ будет зачтен, если и только если $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$.

Система оценивания

В тестах общей стоимостью 20 баллов выполняются дополнительные ограничения $n \leq 10$, $l_i \leq 10$.

В тестах общей стоимостью 40 баллов выполняется дополнительное ограничение $n \leq 1000$.

В тестах общей стоимостью не менее 30 баллов выполняется дополнительное условие $p_i = i - 1$.

Пример

Ввод	Вывод
5 1 2 3 4 1 2 3 4 5	10.00000000000000000000
3 1 2 1 2 1	0.50000000000000000000
4 1 1 3 1 3 4 1	3.00000000000000000000

Примечание: В первом примере можно долить 4 литра воды в первый бункер, 3 литра воды во второй, 2 литра воды в третий бункер и 1 литр в четвертый. Тогда во всех бункерах будет по 5 литров воды, а всего нужно будет долить 10 литров.

Во втором примере нужно перелить 0.5 литра воды из второго бункера в третий и долить 0.5 литра в первый бункер. Тогда в каждом бункере будет по 1.5 литра воды, а долить всего придется 0.5 литра.

В третьем примере нужно перелить один литр из третьего бункера в четвертый, а затем долить один литр воды в четвертый бункер и два литра воды в первый бункер.

Задача 6. Шаблонная задача

Ввод	prod.in или с клавиатуры
Вывод	prod.out или на экран
Ограничение по времени	1 с
Ограничение по памяти	256 Мб
Максимальный балл за задачу	100

Как известно, в изучении математики кроме освоения теории стоит уделять внимание задачам. Увы, придумать что-то новое и содержательное среди основ трудно, так что задачи как правило шаблонны и различаются незначительными деталями. Но делать шаблонные задачи не так легко, как кажется. С одной стороны, даже при незначительных изменениях можно сильно усложнить задачу. С другой стороны, задача не должна быть слишком легкой.

Пусть нам нужно сделать задачу из ровно двух похожих пунктов, и у нас есть n вариантов для каждого из пунктов. Сложность i -го варианта равна a_i . Если мы выберем пункты i и j (так как нам нужно 2 различных пункта, то $i \neq j$), то итоговая сложность задачи будет $a_i \cdot a_j$. Она должна быть максимально возможной, но не больше заранее известного числа A . Найдите такую сложность.

Формат входных данных

В первой строке входных данных находятся два целых числа n и A — количество имеющихся пунктов и ограничение на сложность задачи ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq A \leq 10^{18}$).

Во второй строке находятся n целых чисел a_i ($1 \leq a_i \leq 10^{18}$) — сложности пунктов.

Формат выходных данных

В первой строке выведите одно целое число — максимальную сложность задачи, которую можно составить ровно из двух пунктов. Во второй строке выведите два целых числа i и j ($1 \leq i, j \leq n$, $i \neq j$) — номера пунктов, для которых достигается оптимальный ответ. Пункты нумеруются с 1 в том порядке, в котором они указаны в условии.

Если удовлетворяющую условию задачу составить нельзя, выведите одно число 0.

Система оценивания

В тестах общей стоимостью не менее 15 баллов $n \leq 5000$.

В тестах общей стоимостью не менее 33 баллов $A \leq 1000\,000$.

Пример

Ввод	Вывод
4 16 8 1 4 1	8 1 2
4 16 16 8 4 2	16 2 4

Задача 7. Дораскладывать косынку

Ввод	sol.in или с клавиатуры
Вывод	sol.out или на экран
Ограничение по времени	1 с
Ограничение по памяти	256 Мб
Максимальный балл за задачу	100

Как известно, у классического пасьянса «Косынка» есть финальная стадия, когда все карты уже открыты, колоды нет, и надо просто переложить все карты на тузов. В большинстве компьютерных реализаций «Косынки» это делается автоматически, но мы предлагаем это сделать вам.

Правила этой стадии можно вкратце описать следующим образом. В пасьянсе используется стандартная колода из 52 карт. А именно, каждая карта характеризуется *достоинством* (в порядке возрастания: туз, 2, 3, 4, 5, 6, 7, 8, 9, 10, валет, дама, король) и *мастью* (червы, бубны, трефы, пики); существует ровно одна карта для каждой пары (достоинство, масть). Обратите внимание, что (в отличие от ряда других карточных игр) туз считается картой с наименьшим достоинством.

На столе определены четыре места для стопок карт — так называемые *дома*. На каждом доме можно формировать стопку из карт одной масти по возрастанию достоинств. А именно, если на доме пока не лежит ни одной карты, то на него можно положить любого туза. Иначе на стопку карт, лежащих на доме, можно положить следующую по достоинству карту той же масти. Например, на пикового туза можно положить пиковую двойку, поверх нее пиковую тройку и т.д.

Кроме того, есть так называемые *ряды*, их ровно 7 штук. В процессе раскладывания пасьянса в них могут быть закрытые и открытые карты, но на интересующем нас этапе закрытых карт нет, и в каждом ряду лежит последовательность (возможно, пустая) открытых карт. Масти карт в одном ряду могут быть различны, но достоинства должны образовывать непрерывную убывающую последовательность.

За один ход можно взять одну карту с низа любого ряда (она всегда будет иметь минимальное достоинство в этом ряду) и переложить ее на любой дом с соблюдением правил размещения карт в доме. Т.е. туза можно перекладывать только на пустой дом, а любую другую карту X — только на дом, на котором уже есть стопка той же масти, и только при условии, что верхняя карта в этой стопке имеет достоинство на единицу меньше чем достоинство карты X .

Например, если в каком-нибудь ряду нижняя карта — это четверка пик, и на одном из домов есть стопка, наверху которой есть тройка пик, то четверку пик можно переложить на эту стопку поверх тройки пик.

Цель пасьянса — переложить все карты на дома, так, чтобы все ряды оказались пустыми.

Формат входных данных

В первой строке дано описание четырех карт — верхних карт в стопках на каждом *доме*. Если вместо карты записано «00», то это означает, что соответствующая стопка пуста. Гарантируется, что масти этих карт различны, и что никакие карты из стопок на домах не присутствуют в рядах, описываемых ниже.

Далее в 7 строках следует описание *рядов* карт. Каждое описание начинается с целого числа k ($0 \leq k \leq 13$) — количества карт в очередном ряду. Далее перечислены k карт данного ряда.

Каждая карта во входных данных представлена строкой длины 2 или 3 символа, последний символ в которой — масть, а первые 1 или 2 символа — достоинство. Достоинства обозначаются или числами от 2 до 10, или символами: J — валет, Q — дама, K — король, A — туз. Масти обозначаются следующим образом: h — червы, d — бубны, s — трефы, s — пики. Гарантируется, что каждая карта встретится в описании не более одного раза, и что положение карт удовлетворяет правилам пасьянса «Косынка».

Формат выходных данных

Выведите последовательность карт в том порядке, в котором их нужно перекладывать на дома. Гарантируется, что решение существует.

Пример

<i>Ввод</i>	<i>Вывод</i>
Kh Ks Kd Kc 0 0 0 0 0 0 0	
4h 6c 6d 5s 9 Kh Qc Jd 10c 9h 8s 7d 6s 5h 8 Ks Qd Js 10h 9c 8h 7c 6h 2 8c 7h 0 5 Kd Qs Jh 10s 9d 7 Kc Qh Jc 10d 9s 8d 7s 0	5h 6h 6s 7c 7d 7h 7s 8c 8d 8h 8s 9c 9d 9h 9s 10c 10d 10h 10s Jc Jd Jh Js Qc Qd Qh Qs Kc Kd Kh Ks
5h 5c 5d 5s 8 Kh Qc Jd 10c 9h 8s 7h 6c 0 8 Kd Qs Jh 10s 9d 8c 7d 6s 8 Ks Qh Js 10d 9c 8d 7c 6h 0 0 8 Kc Qd Jc 10h 9s 8h 7s 6d	6c 6d 6h 6s 7c 7d 7h 7s 8c 8d 8h 8s 9c 9d 9h 9s 10c 10d 10h 10s Jc Jd Jh Js Qc Qd Qh Qs Kc Kd Kh Ks

Примечание: В примерах выше вывод разбит на строки для удобства чтения. Ваше решение должно выводить ответ в одной строке.

В первом примере все карты и так уже лежат на домах, никаких действий не требуется.

Во втором примере можно, например, сначала переложить пятерку червей с первого ряда на стопку на первом доме. После этого можно туда же переложить шестерку червей со второго ряда, дальше можно переложить шестерку пик с первого ряда на соответствующий дом, и т.д.