

Задача 1. A+B

<i>Ввод</i>	<code>aplusb.in</code> или стандартный ввод
<i>Вывод</i>	<code>aplusb.out</code> или стандартный вывод
<i>Ограничение по времени</i>	1 секунда
<i>Ограничение по памяти</i>	256 мегабайт
<i>Максимальный балл за задачу</i>	200

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. Кефир

Ввод	kefir.in или стандартный ввод
Вывод	kefir.out или стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Маленькая девочка Оля очень любит кефир. Каждый день она выпивает ровно k пакетов кефира, если их имеется хотя бы k , в противном случае — все, что есть. Но есть одна проблема: сроки годности. На каждом пакете написан день, позже которого этот пакет пить нельзя (ровно в этот день еще можно). Поэтому если в холодильнике у Оли в какой-то момент оказывается просроченный пакет, она его выкидывает.

Оля очень не любит выкидывать пакеты с кефиром, поэтому она применяет следующую стратегию: каждый раз, когда надо выпить очередной пакет, она выбирает тот, у которого срок годности заканчивается раньше всего. Несложно видеть, что эта стратегия минимизирует количество выкинутых пакетов и, в частности, позволяет обойтись без выкидывания, если это вообще возможно.



Но основная проблема, с которой столкнулась Оля, — это покупка нового кефира. Сейчас у Оли в холодильнике есть n пакетов кефира, про каждый известен его срок годности (через сколько дней он заканчивается). Оля пришла в магазин, и там есть m пакетов кефира, про каждый тоже известен его срок годности.

Определите, какое максимальное количество пакетов может купить Оля так, чтобы потом не пришлось ничего выбрасывать. Считайте, что Оля сегодня еще не пила ни одного пакета кефира.

Формат входных данных

В первой строке находятся три целых числа n, m, k ($1 \leq n, m \leq 10^6$, $1 \leq k \leq n + m$) — число пакетов кефира у Оли в холодильнике, число пакетов кефира в магазине и сколько пакетов выпивает Оля каждый день.

Во второй строке находятся n целых чисел f_1, f_2, \dots, f_n ($0 \leq f_i \leq 10^7$) — сроки годности пакетов кефира, которые уже есть у Оли в холодильнике. Срок годности выражается числом дней, на которые еще можно отложить употребление этого пакета. Таким образом, срок годности 0 означает, что пакет нужно выпить сегодня, 1 — что не позже завтра, и так далее.

Наконец, в третьей строке находятся m целых чисел s_1, s_2, \dots, s_m ($0 \leq s_i \leq 10^7$) — сроки годности пакетов, которые есть в магазине, в аналогичном формате.

Формат выходных данных

Если в любом случае Оля не сможет выпить даже уже имеющиеся у нее пакеты, выведите ровно одно число -1.

Иначе в первой строке выведите максимальное количество пакетов x , которые Оля может купить так, чтобы ничего не выбрасывать. В следующей строке выведите ровно x чисел — номера пакетов, которые

следует взять (пакеты нумеруются в том порядке, в котором они заданы во входном файле, начинается с 1). Естественно, числа не должны повторяться, однако могут идти в произвольном порядке. Если существует несколько подходящих наборов, разрешается вывести любой.

Система оценивания

В тестах суммарной стоимостью не менее 40 баллов все числа во входных данных не будут превосходить 10^4 .

Пример

Ввод	Вывод
3 6 2 1 0 1 2 0 2 0 0 2	3 1 2 3
3 1 2 0 0 0 1	-1
2 1 2 0 1 0	1 1

Примечание: В первом примере из условия $k = 2$ и дома у Оли есть три пакета со сроками годности 0, 1 и 1 (т. е. истекающими сегодня, завтра и завтра), а в магазине есть 3 пакета со сроком годности 0 и 3 пакета со сроком годности 2. Оля может купить три пакета, например, один со сроком годности 0 и два со сроком годности 2.

Во втором примере все три уже имеющихся пакета имеют срок годности, оканчивающийся сегодня, и поэтому Оля не успеет их выпить вне зависимости от того, возьмет ли она еще один пакет в магазине или нет.

В третьем примере сегодня Оля выпьет $k = 2$ пакета (один уже имеющийся и один из магазина), а завтра — оставшийся один пакет.

Задача 3. Спасение принцессы

Ввод	стандартный ввод
Вывод	стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Каждый год тысячам рыцарей приходится спасать принцесс, и наш герой Гарольд — не исключение. Его принцесса находится в самом центре замка, охраняемого злым драконом Русланом. Как и в любом нормальном замке, вокруг покоев нашей принцессы есть ров с водой. Будем считать, что он представляет собой окружность, причем ширина рва равна 0. Цель Гарольда — попасть в покои принцессы и снять с нее древние чары (другими словами, попасть в центр окружности, образованной рвом).



К сожалению, наш рыцарь слеп, поэтому самостоятельно добраться до принцессы он не в состоянии. Но у него есть верный конь Константин, который всегда выручает своего хозяина. Правда, лошади вечно хотят пить, поэтому все, что может Константин, это сбежать до рва с водой и рассказать, сколько он пробежал (то есть расстояние от местоположения Гарольда до ближайшей точки рва). После этого у Константина вырастают крылья, и он может перенести хозяина в любую точку карты.

В данной задаче вы играете роль Гарольда. Вы можете приказывать коню, куда лететь, а в ответ получать минимальное расстояние от данной точки до точек рва. Ваша задача — добраться до центра замка.

Формат взаимодействия с проверяющей программой

Это интерактивная задача. В процессе тестирования ваша программа будет с использованием стандартных потоков ввода/вывода (вывод «на экран»/ввод «с клавиатуры») взаимодействовать с программой жюри, которая моделирует работу коня Константина.

При каждом запуске вашей программы ей необходимо будет решить несколько тестовых случаев. В начале работы ваша программа должна считать одно натуральное число T ($1 \leq T \leq 10^4$) — количество тестовых случаев.

Далее, для каждого тестового случая, ваша программа должна следовать следующему протоколу:

- Для приказанья коню перенести рыцаря ваша программа должна вывести в стандартный поток вывода («на экран») запрос в формате «? x y », где x , y — вещественные числа, координаты точки перемещения. Они не должны превосходить 10^4 по абсолютному значению. После этого ваша программа должна считать из стандартного потока ввода («с клавиатуры») одно вещественное число с ровно 10 знаками после десятичной точки — расстояние от точки (x, y) до ближайшей точки рва. Конечно, из-за конечного числа знаков это число может не быть равно настоящему расстоянию, однако гарантируется, что данное расстояние отличается от настоящего не более, чем на 10^{-10} .
- Когда ваша программа найдет центр замка, она должна вывести его координаты (x, y) в формате «A x y », и перейти к следующему тестовому случаю или завершить работу, если этот тестовый случай был последним.

Координаты центра замка — вещественные числа, не превосходящие по абсолютному значению 10^4 . Радиус рва — неотрицательное вещественное число, не превосходящее 10^4 .

Ваша программа может сделать не более 5 запросов в каждом тестовом случае всего (считая как запросы на перемещение, так и вывод ответа).

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не будет превосходить 10^{-6} для каждой координаты. А именно: пусть ваш ответ равен (x, y) , а загаданный центр (x_1, y_1) . Проверяющая программа будет считать ваш ответ правильным, если $\frac{|x-x_1|}{\max(1, |x_1|)} \leq 10^{-6}$ и $\frac{|y-y_1|}{\max(1, |y_1|)} \leq 10^{-6}$.

Все расстояния в данной задаче измеряются в метрах.

Запросы вашей программы должны завершаться переводом строки и сбросом буфера потока вывода. Для этого используйте `flush(output)` в Pascal/Delphi; `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` в Python.

В случае возникновения каких-либо технических проблем с этой задачей обращайтесь к представителю жюри.

Система оценивания

В этой задаче 33 теста. Верное прохождение одного из тестов оценивается в 4 балла, каждого из остальных тестов — в три балла.

При этом ваша программа также будет получать частичные баллы, если ошибка составила больше 10^{-6} . А именно, если во всех T тестовых случаях одного теста абсолютная или относительная ошибка не будет превосходить 10^{-2} для каждой координаты, то ваша программа получит 1 балл за этот тест. Если во всех T тестовых случаях абсолютная или относительная ошибка не будет превосходить 10^{-4} для каждой координаты, то ваша программа получит 2 балла за этот тест.

Пример

Ввод	Вывод
1	?
2.0000000	? 0.000 0.000
1.0000000	? 3.000 0.000
0.1715729	? 5.000 6.000
	A 3.000 4.000

Примечание: В примере радиус рва равен 3 метрам.

Задача 4. Снековик

Ввод	snack.in или стандартный ввод
Вывод	snack.out или стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Согласно древней легенде, давным-давно жители Анк-Морпорка провинились перед Госпожой Удачей, и та прокляла их. Она сказала, что однажды на город упадут n снеков разных размеров, а жители должны будут составить их в один большой Снековик. При этом, естественно, внизу должны будут быть самые большие снеки, а наверху — самые маленькие.

Прошли годы, и однажды на город стали действительно падать самые разнообразные снеки — от огромных Кендер-сюр до маленьких Что-по-чёмс. И жители города принялись строить из них Снековика.



Правда, их поджидала одна неприятность. Каждый день на город выпадал один снек, но падали они в каком-то странном порядке. Поэтому жители не всегда могли водрузить очередной снек на вершину Снековика; иногда им приходилось выпавший только что снек откладывать до тех пор, пока не выпадут все снеки больше его. Конечно, чтобы не разозлить Госпожу Удачу, жители все-таки устанавливали каждый снек, как только для того появлялась возможность.

Напишите программу, которая будет моделировать деятельность жителей города по постройке Снековика.

Формат входных данных

В первой строке находится одно натуральное число n ($1 \leq n \leq 100\,000$) — общее количество снеков, которые выпадут в городе.

Во второй строке находятся n чисел, i -ое из которых равно размеру снека, выпадающего в i -ый день. Все размеры — различные натуральные числа от 1 до n .

Формат выходных данных

Выведите в выходной файл n строк. На i -й из них выведите размеры всех снеков, которые будут установлены на снековик в i -й день, в том порядке, в котором они будут установлены. Если в какой-то день не будет установлен ни один снек, оставьте соответствующую строку выходного файла пустой.

Пример

Ввод	Вывод
3	3
3 1 2	2 1

Примечание: В примере в первый день выпадает снек размера 3, и его сразу можно устанавливать. Во второй день выпадает снек размера 1, его устанавливать еще нельзя, т.к. снек размера 2 пока отсутствует. В третий день выпадает снек размера 2, его тут же устанавливают, после чего устанавливают выпавший ранее снек размера 1.

Задача 5. Поищите без сдачи!

Ввод	<code>nchange.in</code> или стандартный ввод
Вывод	<code>nchange.out</code> или стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Студент Арсений любит планировать свои действия ровно на n дней вперед. Он каждый день ходит в столовую обедать, и поэтому он уже определился, что он будет заказывать в каждый из этих n дней. Цены в столовой не меняются, поэтому в i -й из этих дней Арсений пообедает ровно на c_i рублей.

В ходу монеты номиналом 1 рубль и купюры номиналом 100 рублей. У Арсения сейчас имеется m монет и достаточно много купюр (вы можете считать, что бесконечно много). Арсений любит современные технологии, поэтому везде, кроме столовой, он расплачивается карточкой, а в столовой карточки не принимают, и ему приходится расплачиваться наличными.

Кассир всегда просит студента расплатиться без сдачи. К сожалению, это не всегда возможно, но Арсений старается минимизировать *недовольство* кассира. Недовольство кассира в каждый из дней определяется суммарным количеством монет и купюр в сдаче Арсению в этот день, а именно, если кассир сдал x купюр и монет в день i , то его недовольство в этот день равно $x \cdot w_i$. Кассир всегда выдает сдачу наименьшим возможным числом купюр и монет, у него достаточно монет и купюр для этого.



Арсений хочет расплачиваться так, чтобы минимизировать суммарное недовольство кассира за n дней. Помогите ему определить, как ему стоит расплачиваться в каждый из n дней!

Заметьте, что Арсению всегда хватит денег, чтобы расплатиться, так как у него бесконечно много купюр. Арсений может использовать монеты и купюры, полученные в сдаче, в любой из следующих дней.

Формат входных данных

В первой строке находятся два целых числа n и m ($1 \leq n \leq 10^5$, $0 \leq m \leq 10^9$) — число дней, на которые Арсений распланировал свои действия, и число имеющихся у него монет в данный момент.

Во второй строке находится последовательность целых чисел c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^5$) — суммы, которые Арсений собирается потратить в каждый из дней.

В третьей строке находится последовательность целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^5$) — коэффициенты недовольства кассира в каждый из дней.

Формат выходных данных

В первой строке выведите одно целое число — минимальное суммарное недовольство кассира, которого может достичь Арсений.

Далее выведите n строк. В i -й из этих строк выведите два числа — число купюр и число монет, которыми Арсений должен расплатиться в i -й день.

Естественно, в любой из дней сумма, которую даст Арсений, должна быть не меньше суммы, на которую он собирается пообедать. Также эта сумма не должна превышать 10^6 рублей: Арсений никогда не носит большие суммы с собой.

Если оптимальных ответов несколько, выведите любой.

Система оценивания

В тестах суммарной стоимостью не менее 20 баллов будут выполняться ограничения $1 \leq n \leq 20$, $0 \leq m \leq 500$.

В тестах суммарной стоимостью не менее 40 баллов будут выполняться ограничения $1 \leq n \leq 500$, $0 \leq m \leq 500$.

В тестах суммарной стоимостью не менее 50 баллов будут выполняться ограничения $1 \leq n \leq 500$.

Пример

<i>Ввод</i>	<i>Вывод</i>
5 42 117 71 150 243 200 1 1 1 1 1	79 1 17 1 0 2 0 2 43 2 0
3 0 100 50 50 1 3 2	150 1 0 1 0 0 50
5 42 117 71 150 243 200 5 4 3 2 1	230 1 17 1 0 1 50 3 0 2 0

Задача 6. Очередь

Ввод	queue.in или стандартный ввод
Вывод	queue.out или стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Наконец-то! Васе исполнилось 14 лет, а значит, пора получать паспорт! Для этого ему необходимо обратиться в паспортный стол. Но не все так просто. В паспортном столе работает только одно окно по приему документов, а очередь в него иногда занимают задолго до его открытия. Вася хочет подать документы завтра.

Он знает, что окно начинает работать спустя t_s минут после полуночи, а закрывается спустя t_f после полуночи (то есть $(t_f - 1)$ — последняя минута, когда окно еще работает). На прием документов от одного человека тратится ровно t минут. Если очередного клиента не успеют обслужить до закрытия окна (т.е. если его очередь наступает менее чем за t минут до закрытия), то его не обслуживают вообще.

Ещё Вася знает, что завтра придет ровно n посетителей. Для каждого посетителя Вася знает время, в которое он придет. Каждый приходящий посетитель занимает очередь в окно, и не уходит, пока его не обслужат (или пока окно по приему документов не закроется). Если в момент прихода посетителя окно свободно (в частности, если в этот же момент закончили обслуживать предыдущего посетителя), то пришедшего посетителя начинают сразу же обслуживать.



Времена приходов всех посетителей положительны, Вася, если надо, может прийти и в нулевой момент времени (т. е. ровно в полночь), однако он не может прийти в момент времени, выражающийся нецелым числом минут после полуночи. Если Вася приходит одновременно с несколькими посетителями, то он пропускает их вперед и встает в конец очереди.

Вася, безусловно, хочет прийти так, чтобы успеть подать документы до закрытия окна. Но из всех таких вариантов он хочет выбрать такой, чтобы стоять в очереди как можно меньше. Помогите ему.

Формат входных данных

В первой строке входных данных находятся три положительных целых числа: время открытия окна t_s , время закрытия окна t_f и время обслуживания одного человека t . В следующей строке дано одно целое число n — количество посетителей ($0 \leq n \leq 100\,000$). В третьей строке перечислены положительные целые числа в порядке неубывания — времена, в которые посетители приходят в паспортный стол.

Все времена заданы в минутах и не превосходят 10^{12} ; гарантируется, что $t_s < t_f$. Также гарантируется, что Вася может прийти так, чтобы успеть подать документы до закрытия окна.

Формат выходных данных

В выходной файл выведите одно целое неотрицательное число — время, когда должен прийти Вася. Если Вася приходит одновременно с несколькими посетителями, то он пропускает их вперед и встает в конец очереди. Если оптимальных решений несколько, выведите любое.

Пример

<i>Ввод</i>	<i>Вывод</i>
10 15 2 2 10 13	12
8 17 3 4 3 4 5 8	2

Примечание: В первом примере первый посетитель приходит точно к открытию окна, и его обслуживают две минуты. В момент времени 12 минут окно освобождается, и, если Вася подойдет к этому моменту, то его обслужат без очереди, т.к. следующий посетитель подойдет только в момент времени 13 минут.

Во втором примере, чтобы успеть подать документы, Васе надо прийти раньше всех.

Задача 7. Гирлянда

Ввод	garland.in или стандартный ввод
Вывод	garland.out или стандартный вывод
Ограничение по времени	1.5 секунды
Ограничение по памяти	256 мегабайт
Максимальный балл за задачу	100

Однажды под Новый Год Диме приснилось, что ему подарили сказочную гирлянду. Гирлянда — это набор лампочек, некоторые пары которых соединены проводами. Дима запомнил, что она представляла из себя единое целое, то есть любая пара лампочек была связана некоторой последовательностью проводов. Еще он заметил, что проводов было ровно на один меньше, чем лампочек.

Необычность гирлянды состояла в том, что яркость каждой лампочки зависела от температуры этой лампочки, которая могла быть как отрицательной, так и положительной! У Димы есть два друга, и он захотел поделиться с ними сказочным подарком. Для этого он планирует разрезать два разных провода так, чтобы гирлянда распалась на три части. Дима хочет, чтобы все три части светились одинаково, то есть чтобы в каждой из них суммарная температура лампочек совпадала. Конечно же, каждая из частей должна быть непустой, то есть в ней должна быть хотя бы одна лампочка.



Помогите ему найти способ, как это сделать, либо определите, что такого не существует.

Осматривая гирлянду, Дима поднял ее, взяв за какую-то лампочку. Таким образом, каждая лампочка, кроме той, за которую он взял, оказалась висящей на некотором проводе. Поэтому в качестве ответа вам нужно вывести номера двух различных лампочек, что будет означать, что Диме нужно разрезать провода, за которые они подвешены. Соответственно, в ответ не может входить лампочка, за которую Дима взял гирлянду.

Формат входных данных

В первой строке находится целое число n ($3 \leq n \leq 10^6$) — количество лампочек в гирлянде.

Затем следует n строк: в i -ой из них находится информация об i -ой лампочке, а именно, два целых числа — номер лампочки a_i , к которой она подвешена (если такой нет, то число 0), и температура t_i ($-100 \leq t_i \leq 100$). Лампочки пронумерованы от 1 до n .

Формат выходных данных

Если решения не существует, выведите ровно одно число -1 .

В противном случае выведите одну строку из двух различных чисел — номеров искоемых лампочек. Если правильных ответов несколько, выведите любой.

Система оценивания

В тестах суммарной стоимостью не менее 50 баллов будут выполняться ограничения $1 \leq n \leq 2 \cdot 10^4$.

В тестах суммарной стоимостью не менее 50 баллов будут выполняться ограничения $1 \leq t_i \leq 100$.

Пример

<i>Ввод</i>	<i>Вывод</i>
6 2 4 0 5 4 2 2 1 1 1 4 2	1 4
6 2 4 0 6 4 2 2 1 1 1 4 2	-1

Примечание: Схема гирлянды и разрезов к первому примеру из условия:

