

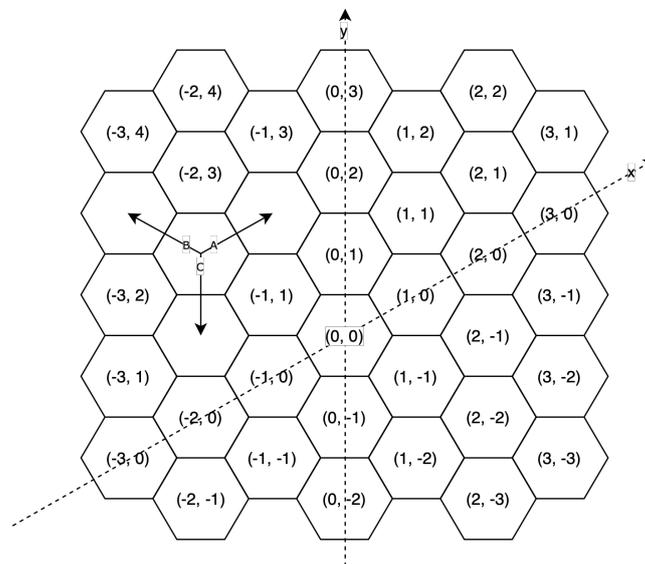
Задача А. Шестиугольная сетка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В некоторых играх серии «Цивилизация» геймплей строится на шестиугольной сетке. Каждый тайл (клетка) может быть использован для строительства зданий или добычи ресурсов. На этой сетке введена система координат с двумя осями: x и y , расположенными под 30° .

От вас же требуется написать модуль, отвечающий за автоматическое перемещения тяжелых войск по этому полю. Всего есть шесть направлений движения из каждого тайла: в любую из соседних с данным по стороне. Однако тяжелым войскам доступно только три направления: A , B и C .

На иллюстрации ниже приведена сетка из шестиугольных тайлов с осями x и y , а также размечены координаты тайлов и указаны направления A , B и C .



Перемещение в каждом из трех доступных направлений занимает ровно один ход. Для каждого из q данных вам тайлов выведите, какое количество шагов в каждом из направлений требуется сделать, чтобы за минимальное число ходов попасть в этот тайл из тайла $(0, 0)$.

Формат входных данных

В первой строке дано целое число q — количество тайлов, которые надо обработать ($1 \leq q \leq 2 \cdot 10^5$).

В i -й из q следующих строк даны два целых числа x_i и y_i — координаты i -го тайла ($-10^8 \leq x_i, y_i \leq 10^8$).

Формат выходных данных

Для каждого тайла выведите на отдельной строке три целых неотрицательных числа — количество шагов направлениях A , B и C соответственно, которые надо сделать, чтобы попасть в него из тайла $(0, 0)$.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	7	$\text{dist}((0, 0), (x_i, y_i)) \leq 1$		полная
2	8	$ x_i , y_i \leq 2$	1	полная
3	12	$x_i, y_i \geq 0$		первая ошибка
4	12	$y_i = 0$		первая ошибка
5	25	$\text{dist}((0, 0), (x_i, y_i)) \leq 8, q \leq 100$	0 – 2	первая ошибка
6	36	нет	0 – 5	первая ошибка

Здесь за $\text{dist}(T_1, T_2)$ обозначено расстояние между тайлами T_1 и T_2 . Расстояние считается как минимальное количество шагов **в любых из шести направлений**, необходимое, чтобы попасть из T_1 в T_2 .

Пример

стандартный ввод	стандартный вывод
4	1 3 0
-2 3	2 0 1
2 -1	0 0 0
0 0	6 1 0
5 1	

Задача В. Странствующий полководец

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

Это интерактивная задача.

В партии в «Цивилизацию» играют два друга: Дима и Паша. Паша несколько ходов назад разместил на поле Великого полководца, а Дима все еще даже не приблизился к его получению. Уже уверенный в своей победе, Паша, уже уверенный в своей победе, предложил Диме сыграть в игру: если тот сможет отгадать местоположение полководца в тумане войны, полководец не будет участвовать в игре.

Будем считать, что их партия в «Цивилизацию» идет на декартовой плоскости, а в качестве клеток выступают точки с целочисленными координатами. Дима может попросить Пашу передвинуть полководца относительно его текущего местоположения на вектор $(\Delta x, \Delta y)$ не более 100 раз.

Назовем городом точку на плоскости, у которой обе координаты целые.

Тогда после каждой просьбы Паша

- перемещает своего полководца: если до запроса он находился на клетке с координатами (x, y) , то после запроса он будет находиться на клетке $(x', y') = (x + \Delta x, y + \Delta y)$;
- сообщает Диме, сколько городов принадлежат отрезку от его столицы, расположенной на клетке с координатами $(0, 0)$, до клетки, на которой расположен Великий полководец, то есть клетки с координатами (x', y') .

Дима не может позволить Паше так легко победить. Помогите ему найти клетку, на которой расположен Великий полководец.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка входных данных содержит одно целое число t — количество наборов входных данных ($1 \leq t \leq 500$). Для каждого набора входных данных запускается процесс взаимодействия с интерактором.

Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете выполнить действие, описанное в условии, не более 100 раз.

Чтобы переместить полководца, выведите строку в формате «? $\Delta x \Delta y$ », после чего полководец переместится на вектор $(\Delta x, \Delta y)$ ($|\Delta x|, |\Delta y| \leq 2 \cdot 10^9$). Интерактор в ответ выведет на отдельной строке количество городов (точек с целыми координатами), которые принадлежат отрезку с краями в столице $(0, 0)$ и текущем местоположении полководца.

Чтобы вывести ответ на задачу, выведите строку «! $x y$ », где в качестве $x y$ должны быть **текущие** координаты полководца. Этот вывод не учитывается в количестве запросов. После этого интерактор выведет вердикт — 1, если ваше предположение верно, и 0 в противном случае. Если ваш ответ был неверен, ваше решение получит вердикт WA (Wrong Answer), а интерактор завершится. Во избежание получения некорректных вердиктов, считав информацию о том, что выведенный ответ неверен, ваше решение тоже должно завершиться.

Гарантируется, что исходная точка, в которой находится полководец, удовлетворяет условию $|x|, |y| \leq 10^9$.

Если в какой-то момент ваша программа превышает лимит в 100 запросов, ваша программа завершится с вердиктом WA.

Обратите внимание, что лимит в 100 запросов устанавливается на каждый из t наборов входных данных.

Важно: не забывайте после каждой выведенной строки сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++,

`System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт TL (Time Limit Exceeded) или IL (Idleness Limit Exceeded).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. В подзадачах значения x_0 , y_0 означают начальные координаты полководца (неизвестные вам).

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	8	$x_0 = 0$ или $y_0 = 0$		первая ошибка
2	10	$ x_0 , y_0 \leq 4$		первая ошибка
3	22	$ x_0 \leq 42$	0	первая ошибка
4	40	$ x_0 , y_0 \leq 10^6$	0, 2, 3	первая ошибка
5	20	нет	0 – 4	первая ошибка

Пример

стандартный ввод	стандартный вывод
1	? 5 5
6	? 3 5
3	? 2 0
11	! 10 10
1	

Замечание

В тесте из примера в самом начале полководец находился в столице — то есть точке с координатами $(0, 0)$.

После первого запроса он переместился в точку $(5, 5)$, и на отрезке, соединяющем его и столицу, оказалось 6 городов с координатами: $(0, 0)$, $(1, 1)$, $(2, 2)$, $(3, 3)$, $(4, 4)$, $(5, 5)$.

После второго запроса полководец перешел в точку $(5 + 3, 5 + 5) = (8, 10)$, тогда на отрезке между полководцем и столицей оказались 3 города: $(0, 0)$, $(4, 5)$ и $(8, 10)$.

Третьим запросом Дима передвинул полководца в точку $(8 + 2, 10 + 0) = (10, 10)$, и на отрезке оказались 11 городов: $(0, 0)$, $(1, 1)$, \dots , $(10, 10)$.

После этих действий Дима предположил, что полководец сейчас находится в точке $(10, 10)$ — и оказался прав.

Задача С. Торговля

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

Рынок и торговля — важный аспект цивилизации. Любой достаточно умелый игрок в «Цивилизацию» в какой-то момент достигает этапа игры, когда его склады завалены разнообразием редких ресурсов. И иногда их становится так много, что высчитывать оптимальную цену продажи каждого отдельного ресурса становится невозможно.

Вот и Паша попал в такую же ситуацию. Чтобы упростить себе остаток партии, он решил увеличить цены на некоторые из своих редких ресурсов таким образом, чтобы среди них осталось ровно k различных, а суммарный прирост цен был минимален.

Более формально, сейчас его цивилизация продает n товаров, каждый из них имеет свою стоимость s_i , причем все s_i различны. Необходимо выбрать новые цены для товаров e_i , что

- все e_i целые;
- $e_i \geq s_i$ для всех i ;
- среди всех e_i есть в точности k различных значений (не больше и не меньше);
- $\sum_{i=1}^n (e_i - s_i)$ минимальна.

Вам необходимо подобрать такой набор цен, и определить соответствующий ему суммарный прирост цен всех товаров.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке ввода находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 10^3$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и k — количество товаров и требуемое количество различных цен, соответственно ($1 \leq k \leq n \leq 2 \cdot 10^3$).

Вторая строка набора содержит n целых чисел s_1, s_2, \dots, s_n — цены товаров ($1 \leq s_i \leq 10^9$; все s_i различны).

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^3$.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите единственное число — минимальное суммарное повышение цен, соответствующее всем требованиям.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. За N обозначена сумма n по всем тестовым случаям.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	5	$N \leq 6, s_i \leq 8$		первая ошибка
2	5	$k = 1$		первая ошибка
3	5	$k = 2$		первая ошибка
4	5	$k = 3$		первая ошибка
5	20	$N \leq 200$	1	первая ошибка
6	15	$n - k \leq 100$	1	первая ошибка
7	45	нет	0 – 6	первая ошибка

Пример

стандартный ввод	стандартный вывод
3	2
4 2	6
1 2 4 3	4
7 3	
1 5 12 4 11 6 3	
9 5	
4 6 13 1 3 8 7 12 5	

Задача D. Дерево навыков

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В любой стратегической игре, и в «Цивилизации» в том числе, есть множество скиллов (навыков), которые игрок может изучить и приобрести.

Каждый из n навыков в «Цивилизации» обладает *типом* — военный, культурный, строительный, и так далее. Будем обозначать тип i -го навыка целым числом c_i . Также навыки упорядочены в подвешенное дерево — у каждого навыка есть один предшествующий ему, который необходимо изучить, чтобы получить к нему доступ. Процесс изучения навыков начинается с корня дерева — с базового навыка, изучение которого напрямую или косвенно необходимо для открытия всех остальных.

Скажем, что два навыка u и v *похожи*, если мультимножества типов навыков в их поддеревьях совпадают. Иными словами, чтобы два навыка были похожи, необходимо, чтобы в их поддеревьях было поровну навыков каждого типа.

«Цивилизация VII» еще не вышла, и разработчики проверяют различные гипотезы о том, как дерево навыков должно выглядеть. Для этого они q раз выполняют переподвешивание дерева за другую вершину и вычисляют число пар похожих навыков. При переподвешивании дерева за навык v он становится корневым (самым первым для изучения), а все остальные ребра дерева сохраняются, но на каких-то из них соответствующим образом меняется направление того, какой навык от какого зависит.

Для каждого из q запросов переподвешивания дерева посчитайте количество неупорядоченных пар похожих навыков после выполнения запроса. «Неупорядоченные пары» означает, что пара (u, v) и пара (v, u) считаются за одну. Поскольку ответ может быть большим, выведите его остаток по модулю $10^9 + 7$.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка входных данных содержит одно целое число t — количество наборов входных данных ($1 \leq t \leq 2 \cdot 10^5$). Далее следует описание наборов входных данных.

В первой описания набора входных данных дано целое число n — количество навыков в игре ($2 \leq n \leq 2 \cdot 10^5$).

Во второй строке перечислены n целых чисел c_i — типы навыков ($1 \leq c_i \leq 10^9$).

В i -й из следующих $n - 1$ строк даны два целых числа u_i и v_i — номера навыков, один из которых напрямую зависит от другого ($1 \leq u_i, v_i \leq n$). Какой навык от какого зависит, определяется тем, какой навык выбран «корневым». Гарантируется, что структура зависимостей образует дерево.

В следующей строке дано целое число q — количество запросов на переподвешивание дерева навыков ($1 \leq q \leq n$).

Последняя строка ввода содержит q целых чисел x_i — номера вершин, за которые переподвешивается дерево ($1 \leq x_i \leq n$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого запроса выведите в отдельной строке число пар похожих навыков по модулю $10^9 + 7$.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	10	$t \leq 10, n \leq 50$	0	первая ошибка
2	12	$t \leq 10, n \leq 500$	0, 1	первая ошибка
3	12	$c_i \leq 3$ для всех i		первая ошибка
4	17	$c_i \leq 20$ для всех i	0, 3	первая ошибка
5	25	$q = 1$		первая ошибка
6	24	нет	0 – 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
1 7 1 1 2 4 3 3 4 1 2 1 3 2 4 4 5 3 6 3 7 7 1 2 3 4 5 6 7	1 1 1 1 0 0 1
1 7 1 2 2 3 3 3 3 1 2 1 3 2 4 4 5 3 6 3 7 7 1 2 3 4 5 6 7	4 3 3 3 1 1 1