

Задача А. Документы

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

У Андрея Геннадьевича, учителя лицея №1 города Диманово, на столе лежит стопка из документов. Как положено, документы пронумерованы различными числами и упорядочены по возрастанию номеров.

Ученики решили сыграть злую шутку над учителем: они взяли какое-то (возможно, нулевое) количество документов сверху и отложили в сторону. Далее взяли сверху несколько документов из оставшихся, перевернули их так, что первый стал последним, второй предпоследним и т. д., и положили обратно. Затем снова вернули в стопку отложенные в сторону документы в исходном порядке. Таким образом, если всего было n документов и перевернули документы с l -го по r -й по счёту, то в результате первые $(l - 1)$ документов остались нетронутыми, с l -го по r -й — идут в обратном порядке, а с $(r + 1)$ -го по n -й — идут в изначальном порядке.

Задана последовательность документов, полученная после шутки учеников. Требуется найти числа l и r .

Формат входных данных

В первой строке входного файла записано число n — количество документов на столе Андрея Геннадьевича ($2 \leq n \leq 10^5$).

В следующей строке задано n целых чисел a_1, a_2, \dots, a_n , где a_i обозначает номер i -го сверху документа ($1 \leq a_i \leq 10^9$). Гарантируется, что все номера различны.

Формат выходных данных

В выходной файл нужно вывести через пробел два требуемых числа — l и r .

Гарантируется, что решение существует и единственно.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 11 19 17 13 23	2 4

Задача В. Турнир

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В городе Диманово, на Васильковой поляне ежегодно проводится турнир по смешанным единоборствам. До начала соревнований осталось совсем немного времени, а тренер команды города Петровска так и не придумал, какого из своих бойцов выставить на бой.

В турнире участвует чётное число бойцов. Каждый боец имеет некоторую силу, характеризуемую натуральным числом. Перед началом боя все участники разбиваются на пары. Спортсмены, состоящие в одной паре, бьются до безоговорочной победы одного из них. После первого боя все спортсмены устают настолько, что турнир приходится завершать, и бойцы, побитые, едут домой.

Назовём предсказуемостью поединка между двумя бойцами разность между силой сильнейшего в паре бойца и силой второго. Жюри турнира предпочитает непредсказуемость явственной расправе, поэтому старается максимально сблизить силы соперников. А именно: бойцы разбиваются на пары так, что сумма предсказуемостей поединков была минимальна.

На данный момент осталось лишь одно место участника на турнире. Тренер команды города Петровска хочет выставить на турнир такого спортсмена, которому попался бы соперник слабее его. Ваша задача — помочь тренеру выбрать подходящих бойцов.

Формат входных данных

В первой строке входного файла записано четное целое число n — количество участников турнира ($2 \leq n \leq 2 \cdot 10^5$). В следующей строке через пробел даны $n - 1$ чисел s_i — численные характеристики силы бойцов, которые уже зарегистрировались ($1 \leq s_i \leq 10^9$). В третьей строке дано целое число m — количество претендентов на участие из города Петровска ($1 \leq m \leq 2 \cdot 10^5$). В последней строке через пробел заданы m чисел p_j — силы бойцов из города Петровска ($1 \leq p_j \leq 10^9$).

Гарантируется, что все силовые характеристики бойцов из входного файла различны.

Формат выходных данных

В единственную строку выходного файла нужно вывести m чисел без пробелов: i -е число равно 0, если боец с номером i при участии в турнире окажется слабее своего противника и равно 1 иначе.

Пример

input.txt	output.txt
4	10
4 2 5	
2	
3 1	

Замечание

Силы участников не обязательно даны в отсортированном порядке.

Задача С. Сладкоежки и торты

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Флатландия давно славится своими кондитерскими изделиями. Сладкоежки Байтландии завидуют жителям Флатландии, ведь шанс насытиться сладостями соседей им выпадает лишь на ежегодном фестивале сладостей *Candies Festival (CF)*. Кондитеры фестиваля работают без устали целые сутки, радуя посетителей мероприятия произведениями гастрономического искусства.

На очередном фестивале *CF* различные компании участников решили отведать Флатландских сладостей, и заказали множества пирожных. Приготовив сладости, кондитер выложил пирожные на прилавок. При виде столь аппетитных лакомств, нетерпеливые сладкоежки изъявили желание тут же их попробовать. Они попросили кондитера разрезать каждое из пирожных на две равные по площади части — одну половинку каждый из них съест прямо в магазине, а оставшейся половинкой насладится, придя домой.

Кондитер хочет выполнить просьбу клиентов как можно скорее, и его интересует вопрос: можно ли поделить каждое из пирожных пополам, сделав всего один разрез? Помогите работнику фестиваля с этой задачей.

Каждое пирожное имеет форму прямоугольника. Пирожные могут накладываться друг на друга (однако кондитер выкладывает их крайне аккуратно и нисколько ни портит их внешний вид и вкусовые качества).

Формат входных данных

В первой строке входного файла записано целое число T — количество прилавков, для которых требуется решить задачу ($1 \leq T \leq 10$). Далее следует описание прилавков. Каждое описание состоит из нескольких строк. В первой строке задано целое число N — количество заказанных пирожных ($1 \leq N \leq 10^4$). В следующих N строках задано описание пирожных, состоящее из восьми целых чисел $x_{i1}, y_{i1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}, x_{i4}, y_{i4}$ — координат вершин i -го пирожного, перечисленных соответственно их обходу (по или против часовой стрелки) ($-10^4 \leq x_{ij}, y_{ij} \leq 10^4$). Все координаты целые.

Формат выходных данных

Для каждого из T прилавков необходимо вывести в отдельную строку выходного файла либо целое неотрицательное число K — количество способов произвести описанный разрез, либо, в случае, если способов провести разрез бесконечно много, вместо числа K выведите слово «Infinity» (без кавычек).

Пример

input.txt	output.txt
2	Infinity
1	1
0 0 2 0 2 2 0 2	
2	
0 0 2 0 2 2 0 2	
3 3 4 3 4 4 3 4	

Замечание

Тест считается пройденным, если число способов правильно подсчитано для всех T прилавков.

Задача D. Транспортировка арбузов

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вот и наступила осень — пора собирать оставшийся урожай. К счастью, почти вся работа была сделана летом, поэтому на сентябрь остались только арбузы. Тем не менее, чтобы упростить процесс сбора урожая и не носить арбузы далеко, было решено расставить сразу несколько коробок в ряд, и каждый арбуз был положен в одну из коробок.

Однако, собрать урожай — это ещё не всё дело, теперь его нужно перевезти. Для этого необходимо, чтобы в каждой из коробок лежало *примерно одинаковое количество арбузов*. А именно: от вас требуется переложить арбузы так, чтобы для любых двух коробок количество арбузов, лежащих внутри них, будет различаться не более, чем на единицу. Для того, чтобы достичь этого, за одну минуту вы можете взять любой арбуз и переложить его в одну из **соседних** коробок.

Определите минимальное количество минут, которое вам потребуется, чтобы переложить арбузы указанным выше способом.

Формат входных данных

В первой строке входного файла записаны два числа: n и m — количество арбузов и количество коробок соответственно ($2 \leq m \leq n \leq 1000$).

В следующей строке задано n целых чисел, i -е из них обозначает номер коробки, в которую положили арбуз под номером i ($1 \leq a_i \leq m$).

Формат выходных данных

В выходной файл нужно вывести единственное число — минимальное количество минут, которые требуются, чтобы переложить арбузы описанным способом.

Примеры

<code>input.txt</code>	<code>output.txt</code>
3 3 3 2 3	2
5 3 1 1 2 2 2	1
3 2 1 2 1	0

Замечание

В первом тестовом примере мы можем, например, перенести первый арбуз во вторую коробку, а второй — в первую, тогда в каждой коробке будет лежать ровно по одному арбузу.

Во втором примере мы можем переложить любой из последних трёх арбузов в третью коробку, тогда их количество в первой и второй коробке будет совпадать и различаться с третьей ровно на единицу.

В третьем примере арбузы изначально разложены по коробкам примерно одинаково.

Задача Е. Бомбослав и дорога в Берляндию

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Любитель приключений Бомбослав в свободное от школьных занятий время решил отправиться к своим родственникам в Берляндию. Ввиду особенностей географического расположения Берляндии, добираться Бомбослав решил с помощью морского транспорта: теплоходы, паромы, лайнеры... Возможно, придётся добираться с пересадками.

Путешествие обещает быть долгим и увлекательным, и поэтому Бомбослав хочет запечатлеть как можно больше моментов на фото. Пусть маршрут Бомбослава проходит по очереди через города s_1, s_2, \dots, s_k . Во время каждого плавания между городами Бомбослав хочет, помимо всего прочего, успеть отфильтровать фотографии с предыдущего плавания. Для этого ему потребуется некоторое время, и поэтому необходимо, чтобы каждое следующее плавание длилось **не меньше**, чем предыдущее. Помогите Бомбославу составить маршрут своего путешествия в Берляндию, удовлетворяющий его требованиям.

Формат входных данных

В первой строке входного файла заданы два целых числа N и M — число городов и количество различных рейсов морского транспорта, соответственно ($2 \leq N \leq 10^5$, $1 \leq M \leq 10^5$). В следующих M строках даны описания рейсов. Каждый рейс задан тремя целыми числами u_i, v_i, c_i , записанными через пробел — номерами городов, между которыми ходит i -й рейс, и длительность плавания между этой парой городов в часах ($1 \leq u_i, v_i \leq N$, $1 \leq c_i \leq 10$, $u_i \neq v_i$). Рейсом можно воспользоваться для передвижения как из u_i в v_i , так и из v_i в u_i . Гарантируется, что никакие два рейса не ходят между одной и той же парой городов. В начальный момент Бомбослав находится в городе номер 1, Берляндия имеет номер N .

Формат выходных данных

Если требуемый маршрут существует, то в первую строку выходного файла нужно вывести целое число K — количество городов в маршруте. В следующую строку выведите K **различных** чисел $1 = p_1, p_2, \dots, p_{K-1}, p_K = N$ — номера городов в том порядке, в котором их должен посетить Бомбослав, чтобы добраться до Берляндии. Для каждой пары (p_i, p_{i+1}) из маршрута должен существовать соответствующий рейс, курсирующий между этими городами. Также для каждых двух последовательных рейсов из маршрута время, затраченное на следующий рейс, должно быть не меньше, чем предыдущий. Если подходящих решений несколько, выведите любое.

Если требуемого маршрута нет, в первой строке выведите 0.

Пример

<code>input.txt</code>	<code>output.txt</code>
4 6	3
3 1 2	1 3 4
3 2 5	
4 2 7	
4 1 10	
3 4 6	
1 2 8	