

Задача А. Тимбилдинг

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Руководство Большой Софтверной Компании решило провести тренинги по тимбилдингу для всех n сотрудников компании. На тренинги отведено два дня, в течение которых участники будут выполнять различные задания командами по k человек. Известно, что количество сотрудников компании делится нацело на k , таким образом, в каждый из двух дней будет образовано ровно n/k команд по k человек в каждой. В оба дня возможно деление на произвольные команды, в частности, разбиение на команды во второй день может никак не зависеть от разбиения на команды в первый день.

Сейчас организаторы тренингов заняты составлением графика распределения людей по командам в каждый из двух дней. Так как одна из целей тренингов — увидеть, как сотрудники действуют в одной команде с самыми разными людьми, к распределению по командам имеется естественное требование: количество пар людей, участвующих в тренинге в оба дня в одной и той же команде, должно быть как можно меньше.

Оказалось, что распределить людей требуемым образом — не такая простая задача, как кажется на первый взгляд. Помогите организаторам тренингов определить минимальное количество пар сотрудников, которые окажутся в одной команде в оба дня.

Формат входных данных

В единственной строке входных данных находятся два числа n и k ($4 \leq n \leq 10^9$, $2 \leq k < n$, n делится на k) — количество людей в компании и количество людей в одной команде в оба дня тренинга соответственно.

Формат выходных данных

Выведите минимальное количество пар сотрудников, которые окажутся в одной команде в оба дня тренингов.

Примеры

ВВОД	ВЫВОД
9 3	0
8 4	4

Пояснение

Пронумеруем сотрудников компании числами от 1 до n .

В первом тесте из условия можно в первый день разбить людей на тройки как (2, 4, 9), (1, 3, 8), (5, 6, 7), а во второй — как (2, 5, 8), (3, 4, 7) и (1, 6, 9). При таком разбиении ни одна пара людей не окажется в одной команде в оба дня.

Во втором тесте из условия можно в первый день разбить людей на две команды как (1, 3, 5, 7) и (2, 4, 6, 8), а во второй день — как (1, 2, 7, 8) и (3, 4, 5, 6). Тогда четыре пары людей (1 и 7, 2 и 8, 3 и 5, 4 и 6) окажутся в оба дня в одной и той же команде. Можно показать, что решения лучше не существует.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			n	
0	1 – 2	0	–	Тесты из условия.
1	3 – 8	25	$n \leq 10$	
2	9 – 21	25	$n \leq 500$	
3	22 – 35	25	$n \leq 100\,000$	
4	36 – 47	25	$n \leq 10^9$	

Задача В. Большие планы

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Для путешествий по странам шенгенской зоны турист, который не является гражданином входящих в неё государств, должен получить шенгенскую визу. Она позволяет не только совершать путешествия в любую страну шенгенской зоны, но и свободно перемещаться между ними. Если же турист получает мультивизу — он может совершать произвольное количество путешествий в страны шенгенской зоны, разумеется, если все поездки полностью укладываются в сроки действия визы.

Геннадий только что получил шенгенскую мультивизу на a дней. Он уже запланировал n поездок, когда узнал о наличии дополнительного требования миграционной полиции: для любых последовательных b дней он может находиться в шенгенской зоне не более, чем c дней. При этом день приезда и день отъезда считаются полностью проведёнными в шенгенской зоне днями. Теперь Геннадий просит вас определить, соответствует ли его план требованиям миграционной полиции, или ему лучше сразу отказаться от части поездок.

Формат входных данных

Первая строка ввода содержит четыре целых числа n , a , b и c ($1 \leq n \leq 100\,000$, $1 \leq c \leq b \leq a \leq 1\,000\,000$) — количество поездок, запланированных Геннадием, продолжительность действия мультивизы в днях и параметры требования миграционной полиции соответственно.

Следующие n строк описывают сами предполагаемые поездки. Каждое описание содержит два числа l_i и r_i ($1 \leq l_i \leq r_i \leq a$) — предполагаемые день отъезда и день приезда для i -го путешествия соответственно.

Гарантируется, что поездки не перекрываются, то есть каждый из a дней является частью не более чем одной поездки. Поездки перечислены в произвольном порядке, не обязательно в хронологическом.

Формат выходных данных

Выведите “Yes”, если текущий план Геннадия не нарушает правил миграционной полиции по пребыванию на территории стран шенгенской зоны, в противном случае выведите “No” .

Примеры

ВВОД	ВЫВОД
1 1000 180 90 2 91	Yes
1 1000 180 90 1 91	No
2 1000 100 20 3 21 100 110	No
2 300 10 10 181 270 1 90	Yes

Пояснение

Во втором тесте из примера 180-дневное окно с дня 1 по день 180 (включительно) содержит 91 день предполагаемого пребывания Геннадия в шенгенской зоне, а следовательно не попадает под правила миграционной полиции.

В третьем тесте из примера **одним из** 100-дневных окон, нарушающих правило, будет окно с дня 4 по день 103 (включительно). В этот период Геннадий планирует пробыть на территории стран шенгенской зоны с дня 4 по день 21 (18 дней) и ещё с дня 100 по день 103 (4 дня), что в сумме даст 22 дня.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			n	a	
0	1 – 4	0	–	–	Тесты из условия.
1	5 – 40	50	$n \leq 1000$	$a \leq 1000$	
2	41 – ∞	50	–	–	Offline-проверка.

Задача С. Чистые носки

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Бомбослав только что забрал чистые вещи из прачечной и разложил их по полочкам в своём шкафу. Теперь у Бомбослава в ящике лежат n чистых носков, цвет i -го из них выражается целым неотрицательным числом c_i , определяющим некоторый оттенок серого цвета. Чем больше значение c_i , тем светлее носок, в частности, $c_i = 0$ означает, что носок полностью чёрный.

Каждое утро Бомбослав достаёт из ящика два носка и надевает их, а вечером кладёт их в корзину с грязным бельём и **больше не использует**, пока не сходит снова в прачечную. Бомбослав опасается полиции моды, поэтому никогда не наденет два носка, если их оттенки серого отличаются более чем на d . Формально говоря, Бомбослав может одновременно надеть носки i и j (разумеется, один носок нельзя надеть на две ноги, то есть $i \neq j$), если $|c_i - c_j| \leq d$. Известно, что Бомбослав использует ровно одну пару носков в день.

Бомбослав очень занятой человек, он старается оптимизировать своё время, поэтому его интересует максимальное количество дней, через которое ему всё-таки придётся нести корзину с грязным бельём в прачечную, при условии, что каждое утро он выбирает пару носков оптимально.

Формат входных данных

Первая строка ввода содержит два целых числа n и d ($1 \leq n \leq 200\,000$, $0 \leq d \leq 10^9$) — количество чистых носков в ящике Бомбослава, имеющихся после предыдущего визита в прачечную, и максимально возможная разница в оттенке серого для двух носков в один день соответственно.

Следующая строка содержит n целых чисел c_i ($0 \leq c_i \leq 10^9$), i -е число соответствует оттенку серого носка номер i .

Формат выходных данных

Выведите единственное число — максимальное количество дней, в течение которых Бомбослав может надевать чистые носки, которые отличаются по оттенку серого не более чем на d , и при этом не ходить в прачечную.

Примеры

ВВОД	ВЫВОД
3 1 1 3 4	1
6 0 3 1 5 1 1 1	2

Пояснение

В первом примере есть только одна пара носков, которую может надеть Бомбослав, — это пара из второго и третьего носка: $|c_2 - c_3| = 1 \leq d$.

Во втором примере Бомбослав может надеть только носки одинаковых оттенков серого, поэтому имеющихся шести носков хватит не более чем на два дня: в оба дня Бомбослав наденет пару носков оттенка 1.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. Обратите внимание, что тесты из условия не подходят под ограничения группы 1, но они всё равно должны быть пройдены, чтобы ваше решение было принято на проверку. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения			Комментарий
			n	d	c_i	
0	1 – 2	0	–	–	–	Тесты из условия.
1	3 – 24	30	$n \leq 1000$	$d = 0$	$c_i \leq 1000$	
2	25 – 61	30	$n \leq 200\,000$	$d \leq 200\,000$	$c_i \leq 200\,000$	
3	62 – ∞	40	–	–	–	Offline-проверка.

Задача D. Клавиатура и вирус

Имя входного файла:	<code>input.txt</code> или стандартный поток ввода
Имя выходного файла:	<code>output.txt</code> или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Коля считает себя очень крутым программистом. При этом Коля еще и хипстер — он не хочет программировать на какой-то стандартной клавиатуре с резиной под клавишами, и, как истинный ценитель, купил себе модную винтажную механическую клавиатуру, в которой под клавишами расположены настоящие пружины. Клавиатура оказалась настолько старой, что все символы, написанные на её клавишах, давно стёрлись, но это нисколько не смущает Колю — ведь он очень крутой программист!

С помощью своей клавиатуры Коля печатает на n различных языках. По счастливому стечению обстоятельств количество букв в каждом из этих языков совпадает с количеством клавиш на Колиной винтажной клавиатуре и равняется m . Все символы всех языков присутствуют в используемой Колей кодировке и, следовательно, могут быть представлены как числа от 1 до s . Один и тот же символ может присутствовать в произвольном количестве языков, но для двух различных языков обязательно найдётся хотя бы один символ, который есть в одном из них и отсутствует в другом. Некоторые числа от 1 до s могут быть не заняты ни одним известным Коле символом.

Друзьям Коли порядком надоело, что он везде таскает с собой эту новую клавиатуру и раздражает окружающих, громко клацая её механическими клавишами. Они решили подшутить над ним и написали компьютерный вирус, который при каждом переключении языка делает две пакости:

- Вместо переключения на какой-то определённый язык, он меняется на совершенно случайно выбранный из n языков, используемых Колей (при этом язык может и не измениться).
- Раскладка клавиатуры для данного языка перемешивается произвольным образом, выполняется лишь одно условие — разным клавишам соответствуют разные символы.

Столкнувшись со зловредным вирусом, Коля сначала запаниковал, но вскоре понял, что всё не так уж и плохо. Нажав по разу на каждую клавишу, Коля может выяснить, какой символ ей соответствует, а по этой информации уже определить, какой язык сейчас включен. Однако ему кажется, что определить текущий язык можно и за меньшее количество нажатий. Помогите Коле показать друзьям, что он действительно крутой программист, вычислив минимальное количество нажатий на клавиши винтажной клавиатуры, которое придётся сделать Коле для определения текущего языка в самом худшем случае.

Формат входных данных

Первая строка входных данных содержит два числа n и m — количество языков, используемых Колей, и количество клавиш на его новой клавиатуре (а заодно и букв во всех языках) соответственно.

Следующие n строк описывают языки. Каждая из них содержит описание одного языка, состоящее из m номеров символов в Колиной кодировке. Номера даны в **произвольном** порядке. Все номера — целые числа от 1 до s . Параметр s не даётся в тесте, но известен для каждой группы тестов.

Гарантируется, что n , m и s положительные.

Формат выходных данных

Выведите одно число — минимальное количество нажатий на клавиши клавиатуры, которое потребуется сделать Коле, чтобы определить текущий язык в худшем случае.

Примеры

ВВОД	ВЫВОД
2 3 1 2 3 1 3 4	3
3 2 1 2 3 4 5 6	1

Пояснение

В первом тесте из условия, если Коле не повезёт и он нажмёт на клавиши, соответствующие 1 и 3, то он не сможет понять, на каком языке пишет.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую из групп 1, 2, 3, 4 и 5 ставятся только при прохождении всех тестов группы. Обратите внимание: прохождение всех тестов предыдущих групп не требуется. В группе 6 каждый тест **оценивается отдельно**.

Группа	Тесты	Баллы	Ограничения			Комментарий
			n	m	c	
0	1 – 2	0	–	–	–	Тесты из условия.
1	3 – 23	10	$n = 2$	$m \leq 1000$	$c = 10^9$	
2	24 – 35	10	$n = 2$	$m \leq 10^5$	$c = 10^5$	
3	36 – 55	10	$n = 2$	$m \leq 10^5$	$c = 10^9$	
4	56 – 80	20	$n \leq 50$	$m \leq 50$	$c = 10^9$	
5	81 – 89	10	$n \leq 10\,000$	$m \leq 200$	$c = 200$	
6	90 – 129	40	$n \leq 2000$	$m \leq 1000$	$c = 10^9$	n и m возрастают с номером теста.

Задача E. Dungeons & Dragons

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Аня и Боря играют в известную игру Dungeons & Dragons. Сейчас их герой находится в подземелье и сражается с драконом. К несчастью, силы неравны, и у героя друзей осталась лишь одна жизнь, а значит, следующая атака принесёт дракону победу, и игра завершится. Осталась только одна надежда на спасение — нанести критический удар, мгновенно убивающий дракона.

Согласно правилам той версии игры Dungeons & Dragons, в которую играют Аня и Боря, чтобы нанести критический удар, они должны одновременно бросить свои кубики. Они смогут нанести критический удар, только если сумма выпавших на кубиках значений в точности равна n . Кубик Ани устроен таким образом, что на нём может выпасть любое целое число от a до b включительно. Аналогично на кубике Бори может выпасть любое целое число от c до d включительно.

Ребята очень хотят поиграть подольше, поэтому они просят вас посчитать количество возможных исходов, при которых их герой нанесёт критический удар. Два варианта считаются различными, если в них отличаются значения, выпавшие хотя бы на одном из кубиков. Например, вариант, когда на первом кубике выпало значение 1, а на втором выпало значение 2, отличается от варианта, когда на первом кубике выпало значение 2, а на втором — значение 1.

Формат входных данных

В первой строке входных данных содержится единственное целое число n ($0 \leq n \leq 10^{18}$) — сумма очков на двух кубиках, при которой вымышленный герой Ани и Бори наносит критический удар.

Следующие две строки содержат целые числа a и b ($0 \leq a \leq b \leq 10^{18}$), определяющие диапазон возможных значений для первого кубика. Каждое число записано в отдельной строке.

Последние две строки входных данных содержат целые числа c и d ($0 \leq c \leq d \leq 10^{18}$), определяющие диапазон возможных значений для второго кубика. Как и для первого кубика, каждое число записано в отдельной строке.

Формат выходных данных

Выведите единственное целое число — количество вариантов выпавших значений на кубиках, при которых герой Ани и Бори нанесёт критический удар.

Обратите внимание, в четвертой группе тестов во входных и выходных данных будут фигурировать числа, для хранения которых вам может понадобиться 64-битный целочисленный тип данных (`long long` в языках C/C++, `long` в языке Java, `int64` в языке Pascal).

Примеры

ввод	вывод
4 0 4 0 4	5
19 3 7 1 13	2
3 0 10 2 2	1

Пояснение

В первом примере требуемая сумма получится при выпадении на кубиках любой из следующих пар значений: (0, 4), (1, 3), (2, 2), (3, 1) и (4, 0) (первое число соответствует значению, выпавшему на кубике Ани, а второе число — значению на кубике Бори).

Во втором примере подходят только пары (6, 13) и (7, 12).

В третьем примере критический удар будет нанесён только при выпадении пары (1, 2).

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			n, b, d	
0	1 – 3	0	–	Тесты из условия.
1	4 – 39	30	$n, b, d \leq 100$	
2	40 – 57	30	$n, b, d \leq 300\,000$	
3	58 – 84	40	$n, b, d \leq 10^{18}$	

Задача F. Дураки и дороги

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как известно, в Берляндии ровно две проблемы, и дороги — одна из них.

Из курса школьной географии вам должно быть знакомо, что в Берляндии ровно n городов и m дорог с двусторонним движением. Некоторые дороги, будем честны, находятся в плачевном состоянии.

Для поддержания качества дорог правительство объявило некоторые из них платными. Каждая платная дорога обслуживается одной из k компаний, которая и обеспечивает своевременный ремонт дороги (она же и взимает плату за проезд по ней).

В Берляндии не только две проблемы, но и две столицы. Они находятся на разных широтах, поэтому одну называют Северной, а другую — Южной. Споры о том, какая столица главнее, длятся уже много лет, но для компаний важно не кто главнее, а то, что именно между этими двумя городами сосредоточен основной автомобильный трафик.

Берляндская антимонопольная служба заподозрила, что дороги были распределены нечестно, а именно, что существует путь между Северной столицей и Южной, такой что какая-то из компаний не владеет ни одной из дорог этого пути. По мнению представителей службы, это создает нездоровую конкуренцию, и таких ситуаций необходимо избегать, но для начала необходимо выявить все компании, страдающие от подобной несправедливости. Эту нелёгкую задачу антимонопольная служба поручила вам.

Назовем компанию *обделённой*, если существует какой-нибудь путь между двумя столицами, на котором нет ни одной дороги, обслуживаемой этой компанией. Выведите номера всех обделённых компаний.

Формат входных данных

В первой строке входных данных записаны три числа n , m и k ($2 \leq n, k \leq 100\,000, 1 \leq m \leq 100\,000$) — количество городов, дорог и компаний соответственно.

Далее следуют m строк, описывающих дороги. В i -й из них записаны три целых числа u_i , v_i и c_i ($1 \leq u_i, v_i \leq n, 0 \leq c_i \leq k$) — номера городов, соединенных i -й дорогой, и номер компании, которая эту дорогу обслуживает. При этом $c_i = 0$ означает, что дорога осталась бесплатной и не принадлежит ни одной компании.

В последней строке записаны два числа a и b ($1 \leq a, b \leq n, a \neq b$) — номера городов, являющихся Северной и Южной столицами соответственно.

Гарантируется, что никакая дорога не соединяет город сам с собой и между каждой парой городов проходит не более одной дороги.

Формат выходных данных

В первой строке выведите количество обделённых компаний.

Во второй строке выведите номера всех обделённых компаний в порядке возрастания.

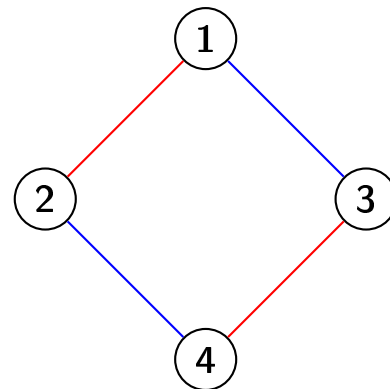
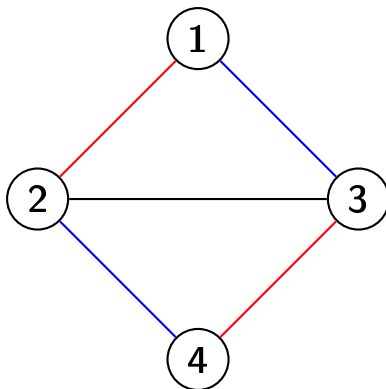
Примеры

ВВОД	ВЫВОД
4 5 2 1 2 1 1 3 2 2 4 2 3 4 1 2 3 0 1 4	2 1 2
4 4 2 1 2 1 1 3 2 2 4 2 3 4 1 1 4	0

Пояснение

В первом примере есть путь 1-2-3-4, на котором дороги только первой компании (на рисунке красные) и бесплатные (черные) и нет дорог второй компании, а также путь 1-3-2-4, на котором только дороги второй компании (синие) и бесплатные.

Во втором примере существует всего 2 пути из 1 города в 4: 1-2-4 и 1-3-4. На обоих присутствуют дороги обеих компаний.



Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения			Комментарий
			n	m	k	
0	1 – 2	0	–	–	–	Тесты из условия.
1	3 – 32	30	$n \leq 1000$	$m \leq 1000$	$k = 2$	
2	33 – 57	30	$n \leq 1000$	$m \leq 1000$	$k \leq 1000$	
3	58 – ∞	40	–	–	–	Offline-проверка

Задача G. Очередь в столовой

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Главный распорядитель столовой Галактической Школы Добра Иннокентий очень любит порядок. Но каждый день на Очень Большой Перемене, когда ученики направляются на обед, в его владениях воцаряется хаос.

Начинается всё вполне безобидно — двое самых проворных школьников встают в очередь. Далее очередь расширяется в k этапов. На i -м этапе ($1 \leq i \leq k$) в каждый промежуток между соседними школьниками, уже стоящими в очереди, вклинивается по a_i человек. Например, в случае $k = 2$, $a_1 = 3$, $a_2 = 1$ после первого этапа расширения в очереди оказывается 5 человек, а после второго — 9.

Несмотря на название учебного заведения, такие метаморфозы очереди не проходят без ссор и потасовок. Уставший от бардака Иннокентий твёрдо решил бороться с этим безобразием. Для того чтобы железной рукой наводить порядок, он хочет научиться выяснять, как происходил процесс расширения очереди, зная только итоговое число n учеников в ней. Понимая, что по n процесс не восстанавливается однозначно, Иннокентий хочет найти максимально возможное число этапов расширения очереди k , а также соответствующий ему набор чисел a_i ($1 \leq i \leq k$), обозначающих количества школьников, которые вклинивались между каждыми двумя соседями в очереди на каждом из этих этапов.

Количество воспитанников Школы, которые могут прийти в столовую, поистине огромно, поэтому за помощь в этом нелёгком деле Иннокентий обратился к вам.

Формат входных данных

На вход программе подаётся одно целое число n ($3 \leq n \leq 2^{64} - 1$) — итоговое число учеников в очереди.

Формат выходных данных

В первой строке выведите одно целое положительное число k — максимальное количество этапов расширения очереди. Во второй строке выведите через пробел k целых положительных чисел a_i ($1 \leq i \leq k$). В случае, если удовлетворяющих условию последовательностей a_i максимальной длины несколько, выведите любую из них.

Примеры

ВВОД	ВЫВОД
4	1 2
9	3 1 1 1

Пояснение

В первом примере, очевидно, есть только одна возможность — на первом шаге вклинивается два школьника.

Во втором примере процесс определён неоднозначно: один вариант развития событий с $k = 2$ приведён в условии, однако максимально возможное число этапов расширения очереди равно трём.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			n	
0	1 – 2	0	–	Тесты из условия.
1	3 – 18	10	$n \leq 20$	
2	19 – 33	15	$n \leq 100$	
3	34 – 47	15	$n \leq 10\,000$	
4	48 – 60	20	$n \leq 10^6$	
5	61 – 81	20	$n \leq 10^{12}$	
6	82 – 100	20	$n \leq 2^{64} - 1$	

Задача Н. Путник

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ежедневно огромное количество людей пользуется электричками. В данной задаче мы рассмотрим направление от станции «Жилые Мытищи» (далее ЖМ) до станции «Большая Центральная» (далее БЦ). Всего вдоль этого направления находится $n + 1$ станция, при этом станция ЖМ имеет номер n , а станция БЦ имеет номер 0. Все пары соседних станций (то есть пары станций с номерами i и $i + 1$) соединены перегонами, при этом никакие другие пары перегонами не соединены.

Каждая электричка стартует в ЖМ и едет в БЦ, при этом на станции номер i ежедневно находятся c_i пассажиров, которые хотят попасть на станцию БЦ. Других пассажиров не бывает, ведь кому придёт в голову ехать не до конечной станции? Для каждого перегона известно количество минут e_i , за которое его проезжает обычная электричка. Посадка пассажиров на электричку происходит моментально.

Совет директоров компании Российские Стальные Дороги решил пустить экспресс из депо станции «Жилые Мытищи» до станции «Большая Центральная». После продолжительных совещаний было решено назвать новый поезд «Путник». На каждом перегоне была построена отдельная колея для нового поезда, и время его проезда от станции i до станции $i - 1$ составляет p_i . Как это часто бывает на практике, p_i может быть как меньше e_i , так и больше.

Поскольку «Путник» — не просто электричка, то ему требуется одна минута на остановку, чтобы бортпроводники провели инструктаж по технике безопасности при поездках на высокоскоростных поездах.

Все пассажиры руководствуются весьма простой логикой: они будут пользоваться новым экспрессом вместо старой электрички, только если это сделает время в пути от их станции до станции 0 строго меньше. Считается, что пассажир садится в «Путник» в самом начале его остановки, то есть тратит минуту на приветствие бортпроводника и прослушивание инструктажа.

Совет директоров хочет максимизировать популярность нового поезда, поэтому они поручили вам выбрать, на каких именно станциях должен останавливаться экспресс, чтобы как можно больше людей им воспользовались. Обратите внимание, что, хотя «Путник» стартует из депо станции «Жилые Мытищи», он не обязан остановиться на этой станции.

Формат входных данных

В первой строке входных данных записано единственное число n ($1 \leq n \leq 1000$) — количество перегонов на пути от ЖМ до БЦ.

Во второй строке записаны n целых чисел e_i ($1 \leq e_i \leq 10^9$) — время в минутах, за которое электричка проезжает путь от станции i до станции $i - 1$.

В третьей строке содержатся n целых чисел c_i ($1 \leq c_i \leq 10^9$) — количество людей, ежедневно пытающихся попасть со станции номер i на станцию 0.

В четвертой строке написано n чисел p_i ($1 \leq p_i \leq 10^9$) — время в минутах, за которое «Путник» проезжает перегон между станциями i и $i - 1$.

Формат выходных данных

В единственной строке выведите одно целое число — максимально возможное количество пассажиров нового экспресса.

Примеры

ВВОД	ВЫВОД
2 1 2 2 2 1 1	0
3 3 2 30 10 100 1 1 2 10	101

Пояснение

В первом примере «Путник» доезжает от ЖМ до БЦ на одну минуту быстрее, чем обычная электричка, если не будет делать никаких остановок. Однако если он где-нибудь остановится, то это преимущество пропадёт, и никто не поедет на новом поезде.

Во втором примере экспресс должен делать остановки на станциях 2 и 3. Остановиться на станции 1 он не может, так как это сделает невыгодными поездки со станции 2.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			n	e_i, c_i, p_i	
0	1 – 2	0	–	–	Тесты из условия.
1	3 – 29	30	$1 \leq n \leq 15$	$1 \leq e_i, c_i, p_i \leq 20$	
2	30 – 49	30	$1 \leq n \leq 300$	$1 \leq e_i, c_i, p_i \leq 20$	
3	50 – ∞	40	–	–	Offline-проверка.

Задача I. Рефераты

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Флорен и её друзьям на уроке английского поручили подготовить рефераты по предоставленным статьям — каждому ученику была выдана отдельная статья, по которой он должен сделать реферат. В рамках данной задачи *статьёй* называется непустая строка, состоящая из строчных символов английского алфавита. *Рефератом* статьи называется непустая строка, являющаяся подстрокой данной статьи. Все ученики должны будут представить свои рефераты у доски. Чтобы их было интереснее слушать, учитель потребовал, чтобы каждый из представленных рефератов не встречался как подстрока ни в одном тексте статьи, кроме той, из текста которой он был образован.

Так как Флорен и её друзья не любят тратить много времени на выполнение домашнего задания, они попросили вас помочь им выбрать для каждой статьи реферат минимальной длины, который не является подстрокой никакой другой статьи, либо сообщить, что выбрать реферат, удовлетворяющий требованию учителя, невозможно. Если для какого-то ученика возможно образовать несколько кратчайших рефератов, из них требуется выбрать лексикографически минимальный. Вы, конечно же, согласились им помочь.

Формат входных данных

В первой строке ввода записано одно число n ($1 \leq n \leq 100\,000$) — количество статей.

Следующие n строк содержат описания статей. Каждая статья представляет собой последовательность строчных символов английского алфавита s_i ($1 \leq |s_i| \leq 500\,000$).

Обозначим через L суммарную длину всех статей, то есть $L = \sum_{i=1}^n |s_i|$. Гарантируется, что во всех тестах $L \leq 500\,000$.

Формат выходных данных

Выведите n строк — рефераты, соответствующие указанным статьям, в том же порядке, что и во входных данных, или символ '?', если для статьи не существует подходящего реферата.

Примеры

ВВОД	ВЫВОД
4 bear deer read beard	? de ad rd
3 xerox roxwill williams	e xw a

Пояснение

Пояснение к первому тесту из примеров.

Любая подстрока первой статьи полностью содержится в четвёртой, поэтому для первой статьи нет подходящего реферата. Во второй статье самыми короткими подходящими рефератами являются 'de', 'ee' и 'er', но лексикографически наименьшим из них является 'de'. В третьей статье самыми короткими подходящими рефератами являются 're' и 'ad', но лексикографически наименьшим из них является 'ad'. В четвёртой статье есть только один самый короткий подходящий реферат — 'rd'.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			n	L	
0	1–2	0	–	–	Тесты из условия.
1	3–22	15	$n \leq 10$	$L \leq 50$	
2	23–45	15	$n \leq 10$	$L \leq 5\,000$	
3	46–68	15	$n \leq 10$	$L \leq 20\,000$	
4	69–108	15	$n \leq 20\,000$	$L \leq 20\,000$	
5	109–148	20	$n \leq 100\,000$	$L \leq 100\,000$	
6	149–196	20	–	–	

Задача J. Новогодний и прямоугольный

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Это **интерактивная** задача.

На Новый год Дед Мороз подарил Глебу то, о чём он уже давно мечтал — клетчатый квадрат размером $n \times n$. Подарок этот не простой, а с сюрпризом — внутри квадрата Дед Мороз выбрал некоторый непустой прямоугольник, и в каждую клетку этого прямоугольника он положил по мандарину.

Теперь, чтобы получить желанный подарок, Глебу нужно сыграть с Дедом Морозом в очень интересную игру. Глеб должен отгадать, в каком именно прямоугольнике находятся все мандаринки, подаренные Дедом Морозом. Будем считать, что строки и столбцы занумерованы числами от 1 до n снизу вверх и слева направо. Глеб может производить два типа запросов:

- ? $x_1 y_1 x_2 y_2$ ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq n$) — в ответ на этот запрос Дед Мороз говорит, сколько мандаринок находится в прямоугольнике, левым нижним углом которого является клетка (x_1, y_1) , а правым верхним — клетка (x_2, y_2) ;
- ! $x_1 y_1 x_2 y_2$ ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq n$) — когда Глеб уверен, что он точно знает, где находятся мандаринки, он должен сделать запрос такого вида, чтобы сообщить свой ответ. При этом (x_1, y_1) соответствует предполагаемому расположению левого нижнего угла, а (x_2, y_2) — правого верхнего.

Формат входных данных

При запуске решения на вход вашей программе подается одно число n ($1 \leq n \leq 2 \cdot 10^9$) — размер квадрата.

Затем на каждый запрос типа “?” вам будет выдаваться количество мандаринок, находящихся в указанном вами прямоугольнике.

Формат выходных данных

Вы должны выводить корректные запросы в формате, описанном выше. Последним должен следовать единственный запрос вида “!”, после чего ваша программа должна немедленно завершиться. Ваша программа должна произвести не больше q (параметр зависит от номера группы) запросов типа “?”. Обратите внимание, что последний запрос, выводящий ответ, не входит в данные q запросов.

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или `Delphi`, `fflush(stdout)` или `cout.flush()` в `C/C++`, `sys.stdout.flush()` на языке `Python`, `System.out.flush()` на языке `Java`.

Примеры

ввод	вывод
4	? 1 1 4 4
6	? 1 3 4 4
6	? 2 3 4 4
4	! 1 3 3 4

Пояснение

Пример в условии иллюстрирует взаимодействие с проверяющей программой. Для прохождения первого теста не обязательно производить такие же запросы, как в примере.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			n	q	
0	1 – 1	0	$n = 4$	$q = 1000$	Тест из условия.
1	2 – 12	10	$n \leq 10$	$q = 10\,000$	
2	13 – 23	20	$n \leq 100$	$q = 10\,000$	
3	24 – 34	20	$n \leq 10\,000$	$q = 20\,000$	
4	35 – 45	25	$n \leq 2 \cdot 10^9$	$q = 128$	
5	46 – ∞	25	$n \leq 2 \cdot 10^9$	$q = 64$	Offline-проверка.

Задача К. Михаил наносит ответный удар

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 5 секунд
Ограничение по памяти: 4 мегабайта

Обратите внимание на нестандартное ограничение по памяти в данной задаче.

Вася очень любит участвовать в олимпиадах по информатике. Он уже давно понял, что больше всего ему нравится решать задачи, связанные со строками. Особенное впечатление на него произвела недавно изобретенная структура данных — дерево палиндромов. Оказалось, что практически любую задачу на строки можно решить с помощью этой поистине всемогущей структуры, ведь она умеет находить палиндромы даже там, где их нет и в помине!

Васе никак не удается заставить дерево палиндромов не искать, а создавать палиндромы из строк. А ведь так хочется! При этом, так как Вася очень ценит встречающиеся ему строки, он хочет лишь добавлять в них символы в произвольных местах, но не удалять и не изменять их. Помогите ему найти способ превращать обычные строки в палиндромы!

Формат входных данных

В первой и единственной строке входных данных содержится непустая строка s , состоящая из строчных английских букв. Её длина, обозначаемая $|s|$, не превышает 15 000 символов.

Формат выходных данных

В первой строке выведите единственное число a — минимальное количество символов, которые необходимо добавить в строку s , чтобы получился палиндром. Во второй строке выведите строку t , являющуюся палиндромом и получающуюся из s добавлением указанного Вами числа символов (иными словами, должно быть верно $|s| + a = |t|$).

Примеры

ввод	вывод
ab	1 bab
abba	0 abba

Пояснение

Палиндромом называется строка, одинаково читающаяся в обоих направлениях. Так, например, строки “a”, “abraarba” являются палиндромами, а “ab” и “abracadabra” нет.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			$ s $	
0	1 – 2	0	–	Тесты из условия.
1	3 – 12	20	$ s \leq 4$	Строка состоит из символов ‘a’ и ‘b’
2	13 – 38	15	$ s \leq 300$	
3	39 – 51	15	$ s \leq 1000$	
4	52 – 61	20	$ s \leq 5000$	
5	62 – 72	15	$ s \leq 10\,000$	
6	73 – 82	15	$ s \leq 15\,000$	

Задача L. Концентрические окружности

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Космические захватчики в очередной раз отправили экспедицию на Землю. Для устрашения местного населения ими было решено рисовать круги на полях.

Космические захватчики выбрали на поле точку (x_1, y_1) и нарисовали с центром в этой точке n окружностей с радиусами a_1, a_2, \dots, a_n , после чего с центром в другой точке (x_2, y_2) они нарисовали m окружностей с радиусами b_1, b_2, \dots, b_m .

Не то что бы жители Земли хоть сколько-нибудь настороженно относятся ко всяким пришельцам из Космоса, но на всякий случай они хотят знать, сколько на поле образовалось областей, ограниченных линиями нарисованных захватчиками окружностей.

Формат входных данных

В первой строке входных данных находятся два целых числа x_1 и y_1 ($-10^9 \leq x_1, y_1 \leq 10^9$).

Во второй строке расположено число n ($1 \leq n \leq 100\,000$), после которого в строго возрастающем порядке расположены n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^9$).

В третьей строке входных данных находятся два целых числа x_2 и y_2 ($-10^9 \leq x_2, y_2 \leq 10^9$). Гарантируется, что точки (x_1, y_1) и (x_2, y_2) не совпадают.

В четвёртой строке расположено число m ($1 \leq m \leq 100\,000$), после которого в строго возрастающем порядке расположены m целых чисел b_1, b_2, \dots, b_m ($1 \leq b_i \leq 2 \cdot 10^9$).

Формат выходных данных

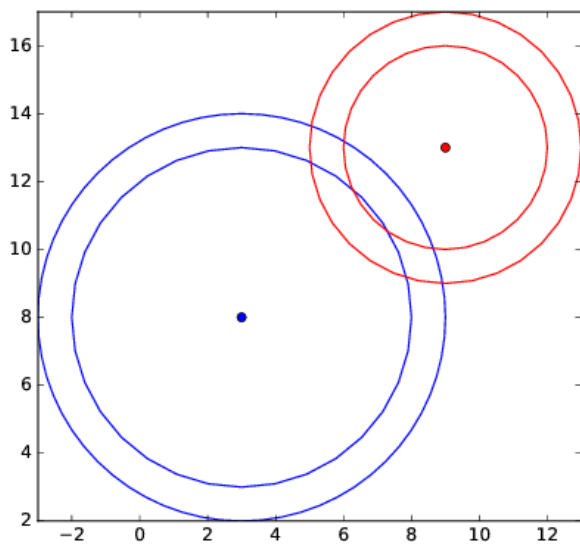
Выведите одно число — количество областей, образованных окружностями, без учёта внешней области.

Примеры

ввод	вывод
3 8 2 5 6 9 13 2 3 4	9
1 1 1 1 5 4 2 4 6	3
-1000 2345 1 1 934 -1344 4 4 5 6 10	5

Пояснение

В первом примере области выглядят следующим образом:



Система оценки

В данной задаче 100 тестов, каждый из которых оценивается независимо и стоит 1 балл. Задача является полностью **Offline**, то есть результаты проверки вашего решения станут доступны уже после окончания соревнования.