

Задача А. И снова!

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

Пете невероятно повезло — к нему приехало такси Жора-3000! Это уникальная система с новейшими реакторам — они представляют из себя n квантовых источников питания, причем все они идеально сбалансированы, поскольку имеют одинаковый заряд.

Несмотря на высокий уровень защиты реакторов, Петя переживал: в последнее время его преследует череда хакерских атак. И переживал он точно не зря — ведь во время полета Жора был взломан!



Взлом нарушил главное свойство Жоры: реакторы перестали быть сбалансированы! Реактор номер i теперь имеет целый положительный заряд d_i . Чтобы снизить риски крушения, Петя должен обеспечить равенство зарядов. За одно действие он может подключиться к одному из реакторов и произвести одну из двух операций:

1. Усилить реактор: если значение заряда реактора составляло d_i , то после оно будет составлять d_i^2 .
2. Ослабить реактор: если значение заряда реактора составляло d_i , то после оно будет составлять $\sqrt{d_i}$. Обратите внимание, что если d_i не является полным квадратом, то выполнять эту операцию строго запрещено — реактор может взорваться.

Теперь Петя хочет узнать, возможно ли восстановить баланс реакторов, и какое минимальное количество действий он должен сделать, если это возможно.

Формат входных данных

В первой строке входных данных находится целое число t ($1 \leq t \leq 1000$) — количество тестовых примеров. Далее следует t тестовых примеров, заданных описанным ниже способом.

В первой строке находится целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество реакторов.

Во второй строке находится n целых чисел d_i ($1 \leq d_i \leq 10^9$) — заряд i -го реактора.

Гарантируется, что сумма n по всем тестовым примерам не превышает $3 \cdot 10^5$.

Формат выходных данных

Для каждого из тестовых примеров выведите в отдельной строке минимальное количество действий, которое Петя должен сделать, чтобы восстановить баланс реакторов, или -1 , если это невозможно сделать.

Система оценки

Пусть sum_n — сумма n по всем тестовым примерам, а w_{one} — количество реакторов с зарядом 1 в тестовом примере.

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$sum_n \leq 10$	12	
2	$sum_n \leq 1000$	9	1
3	$d_i \leq 4$	14	
4	$d_i \leq 10$	18	3
5	$w_{one} \geq 1$	11	
6	$sum_n \leq 10^5$	22	1 - 2
7	Нет дополнительных ограничений	14	1 - 6

Пример

стандартный ввод	стандартный вывод
6	3
4	-1
2 4 4 256	0
3	-1
3 9 27	5
1	6
17	
5	
1 1 2 3 1	
6	
3 3 3 81 81 9	
6	
10000 10 100 10000 100000000 10	

Замечание

В первом тестовом примере необходимо один раз усилить первый реактор, и два раза ослабить четвертый реактор.

Во втором тестовом примере невозможно восстановить баланс реакторов.

Задача В. Киберцпак

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

Такси Пети совершило аварийную посадку прямо на сад Казимира! Комиссия марсианских учёных, прибывшая в сад Казимира Казимировича на срочный вызов, смогла определить, что яблоньеды находились в той самой теплице, на которую приземлился Жора.

На радостях марсианский садовник согласился подвезти Петю на своей новоприобретённой космо-машине — киберцпаке. Ученых также решено было взять с собой, поскольку им тоже надо было попасть на планету Земля для проведения экспериментов.

Это была самая современная машина из всех, на которые у Казимира хватило денег. Она не имеет большого количества удобств, однако славится своим высочайшим уровнем защищённости от хакерских атак — в ней попросту нечего взламывать.



Оказавшись внутри, Казимир продемонстрировал своим пассажирам уникальную игру, которая встроена в киберцпак, чтобы пассажиры могли скоротать время. На панели в ряд расположены k деревянных табличек, на каждой из которых написана одна буква — a , b или c соответственно. За одно действие можно взять две любые деревянные таблички и поменять их местами. Нужно всего-то сделать так, чтобы таблички составляли палиндром. Казимир не хотел бы, чтобы его друзья подумали, что он не умеет решать эту головоломку, поэтому он хочет знать, какое минимальное количество действий для этого ему необходимо совершить.

Формат входных данных

В первой строке находится одно целое число k ($2 \leq k \leq 10^5$) — количество табличек.

Во второй строке находятся k букв, каждая из которых равна a , b или c — описание дощечек.

Формат выходных данных

Если из табличек нельзя сделать палиндром за любое количество действий, то выведите -1 .
Иначе выведите минимальное число ходов, необходимое для достижения цели.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$k \leq 3$	5	
2	$k \leq 10$	9	1
3	$k \leq 1000$	16	1, 2
4	На дощечках написаны только буквы a и b	18	
5	k чётно	28	
6	Нет дополнительных ограничений	24	1 - 5

Примеры

стандартный ввод	стандартный вывод
3 baa	1
3 abc	-1
8 abaccaba	0
9 bbabacccc	3

Замечание

Палиндромом называется последовательность букв, которая читается одинаково как слева направо, так и справа налево. Например, строки `abacaba`, `aba` и `c` являются палиндромами, а строки `abc`, `abac`, `abab` — нет.

В первом тестовом примере мы можем сделать из табличек палиндром за 1 действие — поменяв местами первую и вторую дощечку.

Во втором тестовом примере получить палиндром невозможно.

В третьем тестовом примере ответ 0, так как строка изначально является палиндромом.

Задача С. Блюдо с подвохом

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

В процессе путешествия было принято решение остановиться и перекусить. Пока все выбирали себе блюда, Казимир был уверен в своём выборе: он заказывает кактусный шашлык!



Кактусный шашлык — блюдо, состоящее из кусков шашлыка, пронумерованных целыми числами от 1 до n . Также к блюду прилагаются жареные рёбрышки, каждое из которых соединяет между собой два куска. Если представить куски как вершины, а рёбрышки — как рёбра, то блюдо составляет правильный вершинный кактус.

Вершинным кактусом называется связный неориентированный граф, в котором каждая вершина лежит не более чем на одном простом цикле. Простым циклом называется последовательность различных вершин, в которой каждые две соседние вершины соединены ребром, а первая соединена с последней.

Однако вот незадача — Казимиру принесли не кактусный шашлык, а *деревянный шашлык*! Садовник не мог смириться с таким мошенничеством, поэтому попросил позвать менеджера.

Менеджер пытался объяснить Казимиру, что дерево является частным случаем вершинного кактуса, ведь деревом называется связный граф из n вершин и $n - 1$ рёбер. Но Казимир счел этот аргумент совершенно не убедительным.

Менеджер кафе решил не привлекать лишнего внимания и предложил Казимиру дополнить его блюдо совершенно бесплатно. Однако все не так просто: существуют всего m таких пар (i, j) , что можно добавить рёбрышко между кусочками под номерами i и j . При этом Казимир не может себе позволить, чтобы в результате добавления блюдо перестало быть шашлычным кактусом.

Садовник был очень рад такому исходу событий и теперь хочет определить, какое максимальное количество бесплатных дополнительных рёбрышек он может получить.

Формат входных данных

В первой строке входных данных находится целое число t ($1 \leq t \leq 1000$) — количество тестовых примеров.

Далее следует t тестовых примеров, заданных описанным ниже способом.

В первой строке находится два целых числа n, m ($1 \leq n, m \leq 5 \cdot 10^5$) — количество вершин дерева и количество пар, которые можно будет добавить к блюду.

В каждой из следующих $n - 1$ строк находится по два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) — ребра дерева.

В каждой из следующих m строк находится по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — пара, которую можно добавить к дереву. Гарантируется, что ни одна из дополнительных пар не совпадает с каким-либо ребром дерева.

Обозначим сумму n по всем тестовым примерам как S_n , а сумму m по всем тестовым примерам — как S_m . Тогда гарантируется, что $S_n, S_m \leq 5 \cdot 10^5$.

Формат выходных данных

Для каждого из тестовых примеров выводите ответ в отдельной строке. Каждая строка должна содержать одно число — максимальное количество бесплатных дополнительных рёбрышек, которые может получить Казимир в соответствующем тестовом примере.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n, t \leq 100, m \leq 10$	9	
2	$m \leq 15$	16	1
3	$S_n, S_m \leq 5000, u_i = i, v_i = i + 1$	5	
4	$u_i = i, v_i = i + 1$	12	3
5	a_i является предком b_i в дереве с корнем в 1	17	
6	$S_n, S_m \leq 5000$	19	3
7	Нет дополнительных ограничений	22	1 - 6

Пример

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

Замечание

В первом тестовом примере мы можем взять любую из дополнительных пар, но одновременно их взять невозможно, так как будет нарушено определение кактусного шашлыка.

Во втором тестовом примере мы берем дополнительные пары 1-3, 4-6 и 8-9. Обратите внимание, что невозможно взять одновременно дополнительные пары 1-3 и 3-7, так как будет нарушено определение кактусного шашлыка. Аналогичным образом, невозможно взять одновременно дополнительные пары 4-6 и 4-8.

Задача D. Это же моя машина!!

Имя входного файла: `input1.txt, ..., input25.txt`

Имя выходного файла: `output1.txt, ..., output25.txt`

Новый киберцпак Казимира был уничтожен. Выглядело все так, будто машина подверглась мощному энергетическому удару. Енот, всё это время находившийся в багажнике, отрицает свою причастность, однако лица злоумышленника разглядеть не успел. Пока Казимир выяснял, подлежит ли машина восстановлению, Петя с Енотом занялись осмотром места преступления.

Возле обломков Еноту удалось обнаружить записку с электронным замком и различный мусор, который не давал никакой полезной информации. Петя узнал замок на записке: это прогрессивный нейтринный замок.



Суть таких замков заключается в том, что в них можно ввести любую перестановку. Перестановкой длины n называется массив, содержащий каждое из целых чисел от 1 до n ровно по одному разу. Однако замок откроет содержание записки только если перестановка будет обладать следующими свойствами:

1. Замок крайне сложно устроен, а потому в него можно ввести только перестановку длины от l до r .
2. В перестановке должно содержаться ровно k арифметических прогрессий в качестве подпоследовательностей. Арифметической прогрессией называется любая последовательность длины не менее трех вида $\{a_0, a_0 + c, a_0 + 2 \cdot c, \dots, a_0 + n \cdot c\}$, где a_0 — начальное значение, c — шаг прогрессии, а n — целое положительное число. Подпоследовательностью длины m в перестановке p называется последовательность $p_{i_1}, p_{i_2}, p_{i_m}$ такая, что $1 \leq i_1 < i_2 < \dots < i_m$.

В записке может быть полезная информация, которая поможет вычислить злоумышленника или понять его мотивы, поэтому её надо скорее расшифровать.

Помогите Пете и Еноту найти такую перестановку, которая поможет открыть замок!

Продолжение следует...

Формат входных данных

Это задача с открытыми тестами.

Входные данные находятся в файлах `input1.txt, input2.txt, ..., input25.txt`.

Единственная строка входных данных содержит четыре целых числа t, k, l, r и одно вещественное число w ($0 \leq w \leq 4$) — номер теста (для примера из условия $t = 0$), необходимое количество арифметических прогрессий, наименьшая длина перестановки, наибольшая длина перестановки и особое число для оценивания.

Формат выходных данных

На проверку необходимо сдать выходные файлы с названиями `output1.txt`, `output2.txt`, ..., `output25.txt`, где выходной файл `outputX.txt` должен соответствовать входному файлу `inputX.txt`.

В первой строке выведите целое число n — длину выбранной перестановки.

Во второй строке выведите n целых чисел — саму перестановку.

Система оценки

Если выходной файл не соответствует указанному формату выходных данных, то Вы получите 0 баллов за тест.

Если Вы вывели не перестановку, то Вы также получаете 0 баллов за тест.

Пусть n — длина сгенерированной Вами перестановки, а k^* — количество арифметических прогрессий подпоследовательностей в ней. Тогда Ваш балл за тест равен:

- 4, если $l \leq n \leq r$ и $k = k^*$.
- 0, если $n < l$ или $r < n$.
- $w \cdot \left(\frac{\min(k, k^*) + 1}{\max(k, k^*) + 1}\right)^3$, если $k \neq k^*$.

Баллы за каждый тест округляются вверх до сотых и суммируются. Правила округления таковы, что, например, при округлении числа 10.112 вверх до сотых получаем число 10.12.

Пример

<code>input1.txt, ..., input25.txt</code>	<code>output1.txt, ..., output25.txt</code>
0 10 3 7 2 2	1 4 3 5 6 2 7

Замечание

В примере из условия в ответе восемь подпоследовательностей являются арифметическими прогрессиями:

$\{1, 3, 5\}$, $\{4, 5, 6\}$, $\{4, 3, 2\}$, $\{1, 4, 7\}$, $\{3, 5, 7\}$, $\{1, 3, 5, 7\}$, $\{5, 6, 7\}$, $\{4, 5, 6, 7\}$

А требуется получить десять. Тогда баллы за тест будут равны $2 \cdot \left(\frac{9}{11}\right)^3 \approx 1.0954$, но с учетом округления получаем 1.1.