Задача A. Эффективное тестирование

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

Начиная с $20xx$ года все организаторы всех школьных олимпиад по программированию договорились проводить соревнования исключительно по интернету, для чего было создано общество с ограниченной ответственностью «Организация онлайн-олимпиад» (ООО «ООО»). Разумеется, такая серьёзная организация не может обойтись без собственной тестирующей системы, поэтому для её создания были наняты эффективные менеджеры, закуплены доски и подготовлена синяя изолента.

Для повышения эффективности процесса тестирования была разработана следующая архитектура. Сначала все m тестов задачи располагаются в порядке от 1 к m в очереди тестирования. Затем модуль планирования последовательно выполняет n действий. Действие i состоит в том, чтобы выбрать отрезок очереди с позиции l_i по r_i включительно (в нумерации с единицы) и проверить решение на каждом втором тесте на этом отрезке, а именно на тестах на позициях $l_i, l_i+2, l_i+4, \ldots, r_i$ очереди (при этом гарантируется, что l_i и r_i имеют одинаковую чётность). После этого те тесты, на которых было проведено тестирование, удаляются из очереди, а все оставшиеся тесты сдвигаются по очереди таким образом, чтобы пустых мест не осталось. Например, если в очереди находились тесты с исходными номерами $2, 3, 4, 5, 10, 12, 13, 20$ и была применена операция с $l_i = 3$, $r_i = 7$, то посылка будет протестирована на тестах с позиций 3, 5 и 7, которые исходно имели номера 4, 10 и 13. После выполнения данной операции очередь тестирования будет состоять из тестов с исходными номерами $2, 3, 5, 12, 20$.

Вам поручено реализовать модуль, который для каждого из n описанных выше действий будет определять минимальный и максимальный номер теста в изначальной нумерации из тех, на которых на этом шаге проверялось решение.

Формат входных данных

В первой строке входных данных находятся два числа n и m ($1 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 10^{18}$) — количество действий модуля планирования и количество тестов в задаче.

В каждой из последующих n строк записаны два целых числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant m$) — параметры i-го действия модуля планирования. Гарантируется, что перед началом выполнения действия i в очереди тестирования находятся хотя бы r_i тестов и что числа l_i и r_i имеют одинаковую чётность.

Формат выходных данных

Для каждого из n действий модуля планирования выведите два целых числа — минимальный и максимальный номер теста в исходной нумерации из тех, на которых проверялось решение на соответствующем шаге.

Примеры

стандартный ввод	стандартный вывод
2 10 2 8 1 3	2 8 1 5
4 6 1 1 1 1 1 1 2 2	1 1 2 2 3 3 5 5

Замечание

Рассмотрим, как изменяется очередь тестирования в первом примере.

0. Изначально в очереди тестирования находятся все тесты от 1 до 10, то есть очередь имеет вид $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

1. При выполнении первого запроса будут удалены тесты $2, 4, 6, 8$, и очередь примет вид $1, 3, 5, 7, 9, 10$.

2. При выполнении второго запроса будут удалены тесты 1 и 5, очередь примет вид $3, 7, 9, 10$.

Система оценки

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

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	m		
0	0	–	–	–	Тесты из условия
1	10	$n \leqslant 100$	$m \leqslant 100$	0	
2	9	$n \leqslant 10\,000$	$m \leqslant 10\,000$	0, 1	
3	13	–	$m \leqslant 1\,000\,000$	$0 - 2$	
4	15	$n \leqslant 1000$	–	0, 1	
5	17	$n \leqslant 10\,000$	–	$0 - 2$, 4	
6	36	–	–	$0 - 5$	**Offline-проверка**

Задача B. Застройка мегаполиса

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

В $20yy$ году Москва оказалась застроена до такой степени, что в черте города совсем не осталось территории, пригодной для постройки новых зданий. В поисках новых источников дохода правительство города приняло план, согласно которому все железнодорожные пути в черте города реконструируются и заменяются на подземные, а высвобожденная поверхность используется для коммерческой аренды.

Планирование будущей застройки было начато с участка Октябрьской железной дороги длиной в k метров. Так как строить здания поверх образованного туннеля — долгий и сложный процесс, было принято решение закрепить новый участок за наиболее популярными передвижными точками общепита, продающими мороженное, хот-доги, кофе и тому подобное.

Участок для строительства разделён на k сегментов одинаковой длины, последовательно пронумерованных целыми числами от 1 до k. Из n поступивших в правительство заявок на получение территории i-я претендует на сегменты с l_i по r_i, причём соответствующая точка общепита будет оказывать давление величиной p_i на соответствующий отрезок поверхности. Каждую заявку правительство либо отклонит, либо полностью удовлетворит, предоставив точке все запрошенные сегменты.

Правительство города заинтересовано в том, чтобы сдать каждый сегмент новообразованной территории в аренду хотя бы одной точке общепита. При этом, чтобы уменьшить риск обвала туннеля, было принято решение минимизировать максимальное из оказываемых давлений на каком-либо из сегментов. Обратите внимание, что не запрещается сдавать в аренду один сегмент сразу нескольким точкам общепита, но в таком случае давление, оказываемое ими на данный сегмент поверхности, суммируется.

Помогите правительству одобрить такой набор заявок, чтобы каждый сегмент был сдан в аренду хотя бы одной передвижной точке, но максимальное давление, оказываемое на туннель, было как можно меньше.

Формат входных данных

В первой строке входных данных находятся два целых числа n и k ($1 \leqslant n \leqslant 100\,000$, $1 \leqslant k \leqslant 10^9$) — количество заявок на открытие точек общепита и количество сегментов поверхности.

В последующих n строках описаны заявки, каждая из которых задаётся тремя целыми числами l_i, r_i, p_i ($1 \leqslant l_i \leqslant r_i \leqslant 10^9$, $1 \leqslant p_i \leqslant 10^9$), соответственно границами предприятия и давлением, которое оно оказывает на поверхность туннеля.

Формат выходных данных

Выведите наименьшее возможное максимальное давление, оказываемое на поверхность туннеля точками общественного питания, при условии, что все сегменты поверхности туннеля сданы в аренду. Если не существует способа покрыть весь участок, выведите -1.

Примеры

стандартный ввод	стандартный вывод
3 4 1 3 1 3 4 2 1 4 5	3
1 3 1 2 1	-1
4 5 1 4 3 4 5 5 1 1 3 1 2 1	8
4 5 1 4 1 4 5 1 3 4 1 5 5 1	1

Замечание

В первом тесте из условия оптимальное решение — принять первые две заявки, тогда максимальное давление, равное 3, будет достигаться на третьем сегменте.

Во втором тесте из условия невозможно сдать в аренду третий сегмент.

В третьем тесте из условия одним из оптимальных решений будет удовлетворить все заявки, тогда максимальное давление, равное 8, будет достигаться на четвёртом сегменте. Обратите внимание, что минимизировать или максимизировать количество удовлетворённых заявок не требуется.

В четвёртом тесте из условия оптимальное решение — удовлетворить первую и четвёртую заявки, тогда на все сегменты будет оказываться одинаковое давление, равное 1.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за группу ставятся только при прохождении всех тестов группы и всех групп, от которых зависит данная группа (см. таблицу с системой оценивания). **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. В данной задаче решение не обязано проходить тесты из условия, чтобы быть принятым на проверку.

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	p_i		
0	0	—	—	–	Тесты из условия
1	8	$n \leqslant 10$	$p_i \leqslant 10^9$	–	
2	15	$n \leqslant 3000$	$p_i = 1$	–	
3	21	$n \leqslant 3000$	$p_i \leqslant 10^9$	1	
4	16	$n \leqslant 100\,000$	$p_i = 1$	2	
5	40	–	–	1 – 4	**Offline-проверка**

Задача C. Двоичные карты

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Никогда не поздно научиться играть в увлекательную игру под названием «Двоичные карты»!

В игре участвует неограниченное число игральных карт разных положительных и отрицательных достоинств. Абсолютная величина, написанная на любой карте, является степенью двойки, то есть на карте может быть написано либо 2^k, либо -2^k для некоторого целого $k \geqslant 0$. Карты любого достоинства имеются в неограниченном количестве.

В начале игрок выбирает себе колоду — некоторое множество (возможно пустое) игральных карт. При этом в колоду разрешается включить произвольное количество карт каждого из достоинств, но уровень игрока оценивают по тому, насколько мало карт он включает в свою колоду. Игра состоит из n конов. На i-м кону жюри называет игроку целое число a_i. После этого игрок обязан выложить на стол такое подмножество карт из своей колоды, что сумма достоинств этих карт в точности равна a_i (можно не выкладывать никакие карты вообще, тогда сумма считается равной нулю). Если игрок не может выложить нужный набор карт, он считается проигравшим, а игра заканчивается. В противном случае игрок возвращает выложенные карты назад в свою колоду и переходит к следующему кону. Игрок считается победителем, если он на каждом кону смог выложить необходимые карты.

До вас дошли слухи, какие числа a_i жюри собирается назвать на каждом кону. Теперь вы хотите выбрать колоду с минимальным количеством карт, которая позволит вам выиграть в «Двоичные карты».

Формат входных данных

В первой строке находится целое число n ($1 \leqslant n \leqslant 100\,000$) — количество названных чисел в игре.

Во второй строке находятся n целых чисел $a_1, a_2, \ldots, a_n$ ($-100\,000 \leqslant a_i \leqslant 100\,000$) — числа, называемые на каждом кону.

Формат выходных данных

В первой строке выведите число k ($0 \leqslant k \leqslant 100\,000$) — минимальное количество карт, которые нужно выбрать в колоду, чтобы выиграть в «Двоичные карты».

Во второй строке выведите k целых чисел $b_1, b_2, \ldots, b_k$ ($-2^{20} \leqslant b_i \leqslant 2^{20}$, $|b_i|$ является степенью двойки) — достоинства карт, составляющих колоду. Достоинства можно выводить в любом порядке. Если существует несколько колод оптимального размера, разрешается вывести любую из них.

Гарантируется, что существует колода карт минимального размера, удовлетворяющая требованиям на величины чисел выше.

Примеры

стандартный ввод	стандартный вывод
1 9	2 1 8
5 -1 3 0 4 7	3 -1 -4 8
4 2 -2 14 18	3 -2 4 16

Замечание

В первом тесте из условия на единственном кону можно положить обе карты из колоды. Обратите внимание, этот тест из условия — единственный из подходящих под ограничения первой группы тестов.

Во втором тесте из условия в первый кон можно выложить единственную карту -1, во второй кон — карты 4 и -1, в третий кон можно не выкладывать ничего, в четвёртый кон — только карту 4, а в пятый кон — всю колоду.

Система оценки

Тесты к этой задаче состоят из четырнадцати групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех предыдущих групп. Обратите внимание, прохождение тестов из условия **не требуется** для принятия на проверку. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	$\lvert a_i \rvert$	
0	0	–	–	Тесты из условия
1	8	$n = 1$	$\lvert a_i \rvert \leqslant 10$	–
2	7	$n \leqslant 10$	$\lvert a_i \rvert \leqslant 10$	–
3	7	$n \leqslant 30$	$\lvert a_i \rvert \leqslant 30$	–
4	7	$n \leqslant 50$	$\lvert a_i \rvert \leqslant 50$	–
5	7	$n \leqslant 100$	$\lvert a_i \rvert \leqslant 100$	–
6	7	$n \leqslant 300$	$\lvert a_i \rvert \leqslant 300$	–
7	8	$n \leqslant 500$	$\lvert a_i \rvert \leqslant 500$	–
8	7	$n \leqslant 1000$	$\lvert a_i \rvert \leqslant 1000$	**Offline-проверка**
9	6	$n \leqslant 3000$	$\lvert a_i \rvert \leqslant 3000$	**Offline-проверка**
10	7	$n \leqslant 5000$	$\lvert a_i \rvert \leqslant 5000$	**Offline-проверка**
11	6	$n \leqslant 10\,000$	$\lvert a_i \rvert \leqslant 10\,000$	**Offline-проверка**
12	7	$n \leqslant 30\,000$	$\lvert a_i \rvert \leqslant 30\,000$	**Offline-проверка**
13	7	$n \leqslant 50\,000$	$\lvert a_i \rvert \leqslant 50\,000$	**Offline-проверка**
14	9	–	–	**Offline-проверка**

Задача D. Отбой

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

Преподаватели Летней Какой-нибудь Школы укладывают учащихся спать.

Домик состоит из n комнат, в каждой из которых должны жить ровно b школьников. Однако к моменту отбоя оказалось, что далеко не каждый школьник находится в своей комнате. Комнаты расположены в ряд и пронумерованы по порядку от 1 до n, изначально в i-й комнате находятся a_i школьников. При этом, в домике находятся все проживающие в нём школьники и только они, то есть, $a_1 + a_2 + \ldots + a_n = nb$. В домике также живут p преподавателей, причём $p \leqslant 2$.

Процесс укладывания школьников спать устроен следующим образом. Один преподаватель встаёт у комнаты 1 и движется в направлении комнаты n. Если есть второй преподаватель, то он встаёт у комнаты n и движется в сторону комнаты 1. Закончив укладывать очередную комнату, преподаватель переходит к следующей, причём если преподавателей двое, то они входят и выходят из своих комнат одновременно. Если n нечётно, а преподавателей двое, то среднюю комнату укладывает спать только первый преподаватель. Как только все школьники уложены спать, процесс заканчивается.

Когда преподаватель заходит в комнату, он считает количество школьников внутри, затем тушит свет и закрывает комнату. Дополнительно, если количество школьников внутри не совпадает с b, этот преподаватель записывает номер комнаты в свою записную книжку и всё равно тушит свет и закрывает комнату. Преподаватели очень торопятся составить учебный план на завтра, поэтому игнорируют, кто именно находится в комнате, считая только количество школьников.

Пока преподаватели находятся в комнатах, школьники могут перебегать между ещё не закрытыми и не укладываемыми прямо сейчас комнатами, причём школьник успевает перебежать не более чем на d комнат (то есть, перемещается в комнату с номером, который отличается не более чем на d). Также, после (или вместо) перемещения любой школьник может спрятаться под кровать в той комнате, в которой находится, тогда преподаватель его не заметит при подсчёте. В любой комнате под кроватями могут спрятаться сколько угодно школьников одновременно.

Формально говоря, происходит следующий процесс:

- Объявляется отбой, к этому моменту в комнате i находится a_i школьников.
- Каждый школьник может переместиться в другую комнату, но не далее, чем на d от изначального положения, либо остаться в текущей комнате. После этого желающие спрятаться под кровать прячутся.
- В комнату 1 заходит преподаватель (а также в комнату n, если преподавателей двое), укладывает спать всех находящимся там школьников и запирает комнату (после этого нельзя ни войти в комнату, ни выйти из неё).
- Школьники из комнат с номерами от 2 до n (или до $n-1$, если преподавателей двое) могут переместиться в другие комнаты, убегая не далее, чем на d комнат от своего **текущего** местоположения, либо остаться в текущей комнате. Желающие спрятаться под кровать прячутся.
- Из комнаты 1 в комнату 2 (и, возможно, из комнаты n в комнату $n-1$) переходит преподаватель.
- Процесс продолжается, пока во всех комнатах школьники не будут уложены спать.

Обозначим за x_i количество комнат с видимым нарушением численности, которые были записаны i-м преподавателем. Школьники знают, что директор смены будет слушать жалобы на безобразие во время отбоя не более, чем от одного преподавателя домика, поэтому хотят действовать таким образом, чтобы максимальное из чисел x_i было как можно меньше. Помогите им определить, какое минимальное значение эта величина может принимать при оптимальных действиях школьников.

Формат входных данных

В первой строке записаны четыре целых числа p, n, d и b ($1 \leqslant p \leqslant 2$, $2 \leqslant n \leqslant 100\,000$, $1 \leqslant d \leqslant n-1$, $1 \leqslant b \leqslant 10\,000$) — количество преподавателей, количество комнат в домике, максимальное количество комнат, которые успевает пробежать школьник, пока преподавателя нет в коридоре, и требуемое количество школьников в одной комнате.

Во второй находятся n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$), i-е из которых соответствует количеству школьников в i-й комнате перед объявлением отбоя.

Гарантируется, что $a_1 + a_2 + \ldots + a_n = nb$.

Формат выходных данных

Выведите одно число — минимальное возможное значение максимального x_i.

Примеры

стандартный ввод	стандартный вывод
1 5 3 1 0 0 0 5 0	0
1 5 3 10 5 1 1 1 42	1
2 5 1 1 1 0 0 0 4	1
2 6 1 2 3 8 0 1 0 0	2

Замечание

В первом примере школьники достаточно быстро бегают, чтобы разбежаться по своим комнатам ещё до того, как преподаватель войдёт в первую, тем самым ответ 0.

Во втором примере преподаватель запишет минимум одну комнату в свою записную книжку. Одной из оптимальных стратегий является следующая: перед тем, как преподаватель войдёт в первую комнату, из пятой комнаты по 10 школьников должны перебежать в комнаты 2, 3 и 4. Далее в комнатах 2, 3, 4 по одному школьнику прячется под кровать, в комнате 5 под кровать прячутся два школьника, после чего школьники не производят никаких действий. При такой последовательности действий школьников преподаватель занесёт в записную книжку только комнату 1.

В третьем примере первые три комнаты посетит первый преподаватель, а последние две — второй. Одним из оптимальных решений является следующее: на первом шаге три школьника должны перебежать из комнаты 5 в комнату 4, на втором шаге два из них должны перебежать в комнату 3 и одному из них следует там спрятаться. Тем самым недовольство первого преподавателя вызовет только комната 2, а второй преподаватель и вовсе не встретит никаких нарушений.

В четвёртом примере одной из оптимальных стратегий является следующая: на первом шаге все школьники из первой комнаты прячутся, а все школьники из второй комнаты перебегают в третью. На втором шаге один школьник перебегает из третьей комнаты в четвёртую, а ещё 5 прячутся. Тем самым первый преподаватель будет недоволен комнатами 1 и 2, а второй — комнатами 5 и 6.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия **не требуется** для принятия на проверку. **Offline**-проверка означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		p	n	b		
0	0	–	–	–	–	Тесты из условия
1	10	$p = 1$	$n \leqslant 1000$	$b = 1$	–	–
2	10	$p = 1$	$n \leqslant 1000$	—	1	–
3	10	—	$n \leqslant 100$	$b = 1$	–	–
4	15	—	$n \leqslant 100$	$b \leqslant 30$	0, 3	–
5	20	—	$n \leqslant 1000$	$b \leqslant 30$	0, 1, 3, 4	–
6	10	$p = 1$	—	—	1, 2	**Offline-проверка**
7	25	—	—	—	0–6	**Offline-проверка**

Задача E. Открытая олимпиада по дизайну

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Если вы вдруг не знаете, то в подготовке соревнований по программированию обычно участвуют люди, которые в прошлом сами являлись участниками подобных соревнований. Это очень удобно: бывшие олимпиадники-программисты часто разбираются не только в написании программ, но и в вёрстке условий, и в настройке тестирующей системы, и ещё в множестве разных интересных (и не очень) дел. А что если бы олимпиаду готовили, например, дизайнеры?

В одной воображаемой вселенной проходит Открытая Олимпиада школьников по Дизайну, состоящая из n задач, которую готовят n дизайнеров условий, один дизайнер названий и один дизайнер шрифтов. Каждый из n дизайнеров условий уже сверстал условие своей задачи, но, так как дизайнеры люди творческие и работают поодиночке, каждый из них оставил разное количество места под название задачи. А именно, макет i-й задачи предполагает l_i букв в названии задачи.

Согласно регламенту олимпиады, требуется, чтобы названия задач состояли из символов Юникода, были **различны** и шли в лексикографическом порядке (см. замечание). Дизайнер шрифтов договорился с дизайнером названий, чтобы тот подобрал такие названия для задач, которые используют как можно меньше различных букв, и дизайнеру шрифтов пришлось бы работать над меньшим количеством изображений символов.

Пока дизайнер названий придумывает названия, дизайнер шрифтов решил поинтересоваться у вас, какое минимальное количество различных букв ему придётся нарисовать, чтобы из них можно было составить различные названия для всех задач и чтобы они шли в лексикографическом порядке. Считайте, что символов в Юникоде достаточно большое количество, чтобы это всегда можно было сделать для любых входных данных, удовлетворяющих ограничениям задачи. Обратите внимание, **не разрешается** менять порядок задач.

Формат входных данных

В первой строке входных данных находится целое число n ($1 \leqslant n \leqslant 100\,000$) — количество задач в олимпиаде.

Во второй строке находится последовательность целых чисел $l_1, l_2, \ldots, l_n$ ($1 \leqslant l_i \leqslant 10^9$) — длины названий задач.

Формат выходных данных

Выведите единственное целое число — минимальное количество различных символов, которое потребуется использовать для составления названий для всех задач.

Примеры

стандартный ввод	стандартный вывод
5 2 2 2 2 2	3
4 3 1 2 2	2

Замечание

Строка $x_1x_2\ldots x_a$ длины a *лексикографически меньше* строки $y_1y_2\ldots y_b$ длины b, если выполнено одно из двух условий:

- либо в первой позиции i, такой что $x_i \neq y_i$, в первой строке стоит меньший символ, чем во второй, то есть $x_1 = y_1$, $x_2 = y_2$, $\ldots$, $x_{i-1} = y_{i-1}$, $x_i < y_i$;
- либо первая строка является строгим префиксом второй строки, то есть $a < b$ и $x_1 = y_1$, $x_2 = y_2$, $\ldots$, $x_a = y_a$.

В первом тесте из условия для составления названий задач можно воспользоваться символами «a» < «o» < «x» и составить названия «aa», «ao», «ax», «ox», «xx».

Во втором тесте из условия можно воспользоваться двумя символами «l» < «o» и составить названия «lol», «o», «ol» и «oo».

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 кроме тестов из условия. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования. В данной задаче решение не обязано проходить тесты из условия, чтобы быть принятым на проверку.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	l_i	
0	0	–	–	Тесты из условия
1	11	$n \leqslant 10$	$l_i \leqslant 5$	Все l_i одинаковые
2	7	$n \leqslant 10$	$l_i \leqslant 5$	
3	20	$n \leqslant 300$	$l_i \leqslant 300$	
4	20	$n \leqslant 5000$	$l_i \leqslant 5000$	
5	21	–	$l_i \leqslant 200\,000$	**Offline-проверка**
6	21	–	–	**Offline-проверка**

Задача F. ООШП

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

Открытое Объединение Шиномонтажников-Перфекционистов (ООШП) — организация, объединяющая n шиномонтажников города N. Все шиномонтажники этой организации пронумерованы последовательными целыми числами от 1 до n в порядке вступления в ООШП и объединены в одну древовидную иерархию таким образом, что шиномонтажник номер 1 является руководителем организации, а у любого другого шиномонтажника i есть *непосредственный начальник* p_i, обязательно имеющий меньший номер. Будем говорить, что шиномонтажник v является *начальником* шиномонтажника u, если v встречается в цепочке непосредственных начальников от u до 1, то есть в последовательности p_u, p_{p_u} и так далее. Шиномонтажник u в таком случае называется *подчинённым* шиномонтажника v.

Поскольку все члены ООШП являются ещё и перфекционистами, то в ходе работы у них часто возникают споры. Будем считать, что споры могут возникать только в паре шиномонтажников, ни один из которых не является подчинённым другого. Для разрешения спора они идут к своему *ближайшему общему начальнику*, то есть шиномонтажнику с максимальным номером, который является начальником для каждого из спорящих. Каждый шиномонтажник кроме первого обладает некоторым *уровнем перфекционизма*, выражающимся целым числом c_i. *Накалом* спора называется сумма уровней перфекционизма двух участвующих в нём шиномонтажников. Наконец, *конфликтностью* рабочего дня называется сумма накалов всех споров, возникших в течение этого дня.

По окончании рабочего дня шиномонтажник v считает себя *эффективным руководителем*, если в течение дня он помог разрешить хотя бы один спор каждому своему подчиненному. Формально говоря, это значит, что для каждого шиномонтажника u, который является подчиненным v, существует такой шиномонтажник w, что u и w конфликтовали в течение дня, а v оказался ближайшим общим начальником u и w. В частности, любой шиномонтажник, у которого нет подчинённых, автоматически считает себя эффективным руководителем.

Вы работаете программистом в ООШП и знакомы со всеми шиномонтажниками в организации. Уходя сегодня с работы, каждый шиномонтажник в компании по секрету сообщил вам, что по итогам сегодняшнего дня считает себя эффективным руководителем. Вы знаете иерархию шиномонтажников ООШП, но не знаете, какие именно споры возникали в течение дня. Теперь вам интересно, какое минимальное значение могла принять конфликтность сегодняшнего дня при условии, что каждый шиномонтажник действительно сегодня был эффективным руководителем.

Формат входных данных

В первой строке находится число n ($3 \leqslant n \leqslant 200\,000$) — количество шиномонтажников в ООШП. Во второй строке находятся $n-1$ целых чисел $p_2, p_3, \ldots, p_n$ ($1 \leqslant p_i < i$), где p_i соответствует номеру шиномонтажника, являющегося начальником шиномонтажника номер i. В третьей строке находятся $n-1$ целых чисел $c_2, c_3, \ldots, c_n$ ($1 \leqslant c_i \leqslant 10^6$), где c_i — уровень перфекционизма i-го шиномонтажника.

Гарантируется, что при заданной структуре иерархии могла случиться такая ситуация, что к концу рабочего дня каждый шиномонтажник будет считать себя эффективным руководителем.

Формат выходных данных

Выведите минимальное возможное значение конфликтности сегодняшнего дня.

Примеры

стандартный ввод	стандартный вывод
5 1 2 2 1 1 1 1 1	8
6 1 1 1 4 4 1 2 3 4 5	25

Замечание

Рассмотрим первый тест из условия. Чтобы достичь указанной величины конфликтности дня, необходимо, чтобы в течение дня возникли споры в парах шиномонтажников $(2, 5)$, $(3, 4)$, $(3, 5)$ и $(4, 5)$.

- Шиномонтажники 3, 4 и 5 автоматически считают себя эффективными руководителями, так как у них нет подчинённых.
- Шиномонтажник 2 считает себя эффективным руководителем, так как он помог шиномонтажнику 3 в споре с шиномонтажником 4, а шиномонтажнику 4 в споре с шиномонтажником 3.
- Шиномонтажник 1 считает себя эффективным руководителем, так как он помог шиномонтажникам 2, 3 и 4 уладить их конфликты с шиномонтажником 5, а шиномонтажнику 5 он помог уладить целых три конфликта.

Накал каждого из споров равен $2 = 1 + 1$, поэтому величина конфликтности в данный день составляет 8.

Во втором примере (который подходит под ограничения второй и четвёртой, но не первой и третьей групп) оптимальное решение можно получить спорами в парах $(2, 5)$, $(3, 6)$, $(4, 5)$ и $(5, 6)$. Значение конфликтности дня при таком сценарии составит $(1 + 3) + (1 + 4) + (2 + 5) + (4 + 5) = 25$. Указанный набор пар не является единственным способом получить минимальное значение конфликтности.

Система оценки

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

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	c_i		
0	0	–	–	–	Тесты из условия
1	25	$n \leqslant 2000$	$c_i = 1$	–	–
2	26	$n \leqslant 2000$	–	1	–
3	25	–	$c_i = 1$	1	**Offline-проверка**
4	24	–	–	0–3	**Offline-проверка**

Задача G. Часовой механизм

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Часовой механизм

Меня зовут Джим ди Гриз, я самый ловкий мошенник и авантюрист во всей галактике. По мотивам моих похождений написано множество книг, а ограблениям, совершённым мною, нет числа. Однако вы смогли застать меня в весьма неприятной ситуации.

Не обнаружив себя перед камерами, перехитрив десяток охранников и обойдя множество ловушек, я смог добраться до желанного ящика с сокровищами. Отворив его крышку, я активировал бомбу с часовым механизмом, который уже отсчитывает секунды до неминуемого взрыва! К счастью, мне уже приходилось сталкиваться с бомбами подобной модели, и я знаю, что часовой механизм можно остановить, если правильно соединить проводами контакты на панели управления.

Передо мной n контактов, соединённых $n-1$ проводами. Контакты пронумерованы целыми числами от 1 до n. Бомба устроена таким образом, что если некоторый набор из $k \geqslant 2$ контактов $c_1, c_2, \ldots, c_k$ соединён по циклу, то есть между парами контактов c_1 и c_2, c_2 и c_3, …, c_k и c_1 есть k **различных** проводов, то срабатывает проверка безопасности, и заряд мгновенно детонирует, не оставляя от неудачливого взломщика и следа. В том числе, если два контакта соединены более чем одним проводом, то на них образуется цикл длины 2, и бомба также взрывается. Соединять контакт с самим собой запрещается.

С другой стороны, если я отсоединю одновременно более чем один провод (иными словами, в какой-то момент времени будет подключено менее $n-2$ проводов), то сработает другая проверка безопасности, которая приведёт к такому же плачевному результату. Таким образом, всё, что мне остаётся, это последовательно вытаскивать провод и вставлять его в новое место, следя, чтобы не образовалось цикла, связывающего контакты.

Я знаю, как надо расположить провода, чтобы остановить часовой механизм. Но у меня остаётся всё меньше и меньше времени на это! Помогите мне выбраться из передряги: найдите кратчайшую последовательность безопасных операций, каждая из которых представляет собой отключение определённого провода и его подключение в новое место, а также выстраивает провода требуемым образом.

Формат входных данных

В первой строке входных данных находится число n $(2 \leqslant n \leqslant 500\,000)$ — количество контактов.

В каждой из последующих $n-1$ строках записана пара целых чисел x_i, y_i $(1 \leqslant x_i, y_i \leqslant n,\ x_i \neq y_i)$, обозначающих номера контактов, соединённых очередным проводом в данный момент времени.

В последних $n-1$ строках в аналогичном формате задаётся схема подключения проводов, останавливающая часовой механизм.

Формат выходных данных

В первой строке выведите число k $(k \geqslant 0)$ — минимальное количество проводов, которые потребуется переподключить.

В последующих k строках выведите четвёрки чисел a_i, b_i, c_i, d_i, означающих, что на i-м шаге нужно отсоединить провод, соединяющий контакты a_i и b_i, и соединить им контакты c_i и d_i. Разумеется, к этому моменту времени провод между контактами a_i и b_i должен присутствовать на схеме.

Если оптимальных последовательностей несколько, то выведите любую из них.

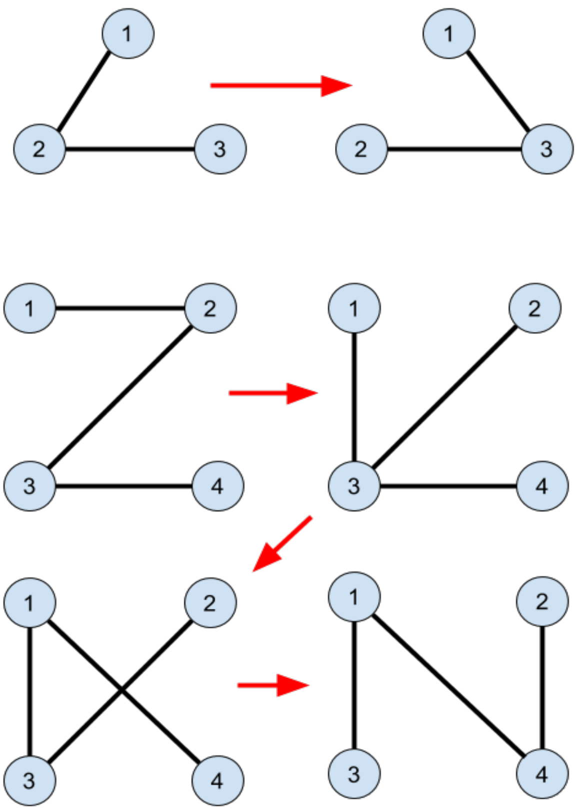
Если требуемой последовательности операций не существует, выведите одно число `-1`.

Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3 1 3 3 2	1 1 2 1 3
4 1 2 2 3 3 4 2 4 4 1 1 3	3 1 2 1 3 4 3 4 1 2 3 2 4

Замечание

Картинка с пояснением к тестам из условия:



Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп, за исключением, возможно, тестов из условия.

Группа	Тесты	Баллы	Ограничения	Комментарий
			n	
0	1 – 2	0	—	Тесты из условия
1	3 – 18	20	$n \leqslant 50$	Гарантируется, что ответ существует и требует не более чем одну операцию
2	19 – 37	20	$n \leqslant 50$	
3	38 – 53	20	$n \leqslant 5000$	
4	54 – 69	20	$n \leqslant 100\,000$	
5	–	20	–	**Offline-проверка**