

Задача А. Бэкапы

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

Вы управляете IT-компанией, которая создает бэкапы компьютерных данных для больших офисов. Делать бэкапы скучно, и поэтому вы разрабатываете свою систему таким образом, чтобы разные офисы могли создавать бэкапы данных друг друга, пока вы сидите дома и играете в компьютерные игры.

Все офисы ваших клиентов расположены вдоль одной улицы. Вы решили объединить офисы в пары, и для каждой пары офисов вы проложите сетевой кабель между двумя зданиями, чтобы они могли создавать бэкапы данных друг друга.

Однако сетевые кабели стоят дорого. Ваша местная телекоммуникационная компания предоставит вам только k сетевых кабелей, что означает, что вы можете организовать резервное копирование только для k пар офисов, (в общей сложности $2k$ офисов). Ни один офис не может принадлежать более чем одной паре (то есть все эти $2k$ офисов должны быть разными).

Кроме того, телекоммуникационная компания взимает плату за километр. Это означает, что вам нужно выбрать эти k пар офисов, чтобы использовать как можно меньше кабелей. Другими словами, вам нужно выбрать пары таким образом, чтобы при сложении расстояний между двумя офисами в каждой паре общее расстояние было как можно меньше.

Например, предположим, что у вас было пять клиентов, офисы которых расположены как показано ниже: в 1 км, 3 км, 4 км, 6 км и 12 км от начала улицы. Телекоммуникационная компания предоставит вам только $k = 2$ кабелей.

Лучше всего соединить первый и второй офис и третий и четвертый офис. Таким образом вы используете все $k = 2$ кабелей и длина первого кабеля будет $3 - 1 = 2$ км, и длина второго кабеля будет $6 - 4 = 2$ км. Таким образом, суммарная длина кабелей будет 4км. Можно показать, что меньшую длину получить нельзя.

Формат входных данных

Первая строка содержит два числа: n и k , обозначающих количество офисов на улице ($2 \leq n \leq 100\,000$) и количество доступных вам кабелей ($1 \leq k \leq \frac{n}{2}$).

Следующих n строк содержат по одному целому числу: ($0 \leq s \leq 1\,000\,000\,000$), i -ое число означает расстояние от i -ого офиса до начала улицы. Офисы отсортированы по расстоянию до начала улицы, от ближайшего до наиболее далёкого. Не существует двух офисов, которые находятся в одном месте.

Формат выходных данных

Вы должны вывести ровно одно целое положительное число - минимальную длину кабелей, необходимую чтобы соединить $2k$ офисов при помощи k кабелей.

Примеры

стандартный ввод	стандартный вывод
5 2 1 3 4 6 12	4

Замечание

Пример входных данных соответствует примеру разобранным в условии.

Задача В. Новый год и смена хендла

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Наступает новый год. Это значит, что на codeforces пришла пора менять хендлы. Мишке захотелось поменять свой хендл, но таким образом, чтобы люди не забыли, кто он такой.

Чтобы все получилось так, как он хочет, он решил менять только регистры букв. Более формально, за **одну** смену хендла он может выбрать любой отрезок своего хендла $[i; i + l - 1]$ и сделать `tolower` или `toupper` ко всем буквам своего хендла на этом отрезке (более формально, поменять все строчные буквы на отрезке на соответствующие прописные или наоборот). Длина l фиксирована для всех смен.

Так как на codeforces нельзя слишком часто менять хендлы, всего Мишка может сделать не более k таких операций. Какого **минимального** значения может быть $\min(lower, upper)$ (где $lower$ — количество строчных букв, а $upper$ — количество прописных букв) после оптимальной последовательности смен?

Формат входных данных

Первая строка входных данных содержит три целых числа n, k и l ($1 \leq n, k, l \leq 10^6, l \leq n$) — длину хендла Мишки, количество смен и длину отрезка.

Вторая строка входных данных содержит одну строку s , состоящую из n строчных и прописных букв латинского алфавита — хендл Мишки.

Формат выходных данных

Выведите одно целое число — минимальное значение, которого может достигнуть $\min(lower, upper)$ после того, как Мишка сменил свой хендл не более k раз так, как описано в условии задачи.

Примеры

стандартный ввод	стандартный вывод
7 1 4 PikMike	0
15 2 2 AaAaAaaAAAAaaA	2
14 2 6 aBcdEFGHIJkLMn	0
9 2 2 aAaAAAAaA	1

Задача С. Путешествие

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

Маленький Петя очень любит путешествовать. В стране Берляндия, где он живет, есть n городов, расположенных на одной прямой. Петя пронумеровал их числами от 1 до n в порядке увеличения красоты. Петя находится в городе 1 и хочет попасть в город n . Чтобы не портить впечатления о поездке, он может посещать города только в порядке увеличения номеров (а, следовательно, и красоты).

Для перемещения между городами Петя решил воспользоваться услугами единственной авиакомпании страны — Berland Airlines. Стоимость перелёта из города i в город j равна $c_i \cdot |x_i - x_j| + t_j$, где x_i — координата города i , x_j — координата города j , а c_i — стоимость единицы самолётного топлива в городе i , а t_j — стоимость въезда в город j .

Чтобы было о чем рассказать друзьям, Петя хочет потратить как можно больше (да-да, именно больше) денег на эту поездку. Помогите ему в этом. Обратите внимание, что Пете не обязательно бывать во всех городах.

Формат входных данных

Первая строка содержит целое число n — количество городов в Берляндии ($1 \leq n \leq 100\,000$).

Далее следуют n строк. Строка с номером i содержит три целых числа — x_i , c_i и t_i ($-10^6 \leq x_i \leq 10^6$, $1 \leq c_i \leq 10^6$, $1 \leq t_i \leq 10^6$).

Формат выходных данных

Выведите искомое наибольшее количество денег, которые Петя может потратить чтобы добраться из города 1 в город n . Гарантируется, что ответ не превосходит 10^{12} .

Примеры

стандартный ввод	стандартный вывод
4	123
5 10 2	
0 1 10	
15 3 14	
17 2 3	

Задача D. Гонцы

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

Давным-давно в прекрасном молдавском государстве существовало N средневековых городов, пронумерованных от 1 до N . Город с номером 1 был столицей. Города были соединены $N - 1$ двусторонними дорогами. Каждая дорога имела свою длину. Существовал единственный способ проехать между любыми двумя городами (т.е. дороги образовывали дерево).

Когда город подвергался нападению, о ситуации нужно было как можно скорее сообщать в столицу. Послания передавались при помощи гонцов. В каждом городе жил ровно один гонец. Каждый гонец характеризовался временем, необходимым ему для начала путешествия, и постоянной скоростью, с которой он двигался после отбытия.

Сообщение всегда передавалось по уникальному пути от города до столицы. Когда гонец приезжал в город, у него было два варианта дальнейших действий: вести сообщение дальше самому, или передать его другому гонцу. Новый гонец тоже тратил некоторое время, а потом ехал в столицу с постоянной скоростью. Одно сообщение могло быть передано при помощи любого количества гонцов.

Вам необходимо для каждого города узнать за какое минимальное время можно передать сообщение из него в столицу.

Формат входных данных

Первая строка содержит одно целое число N , которое означает количество городов в государстве ($3 \leq N \leq 100\,000$). Каждая из следующих $N - 1$ строк содержит три целых числа u , v и d , описывающих дорогу длиной d километров между городами, с номерами u и v ($1 \leq u, v \leq n$, $0 \leq d \leq 10\,000$).

Затем следуют $N - 1$ пар целых чисел, по одной паре на строку. Пара $i, S_i V_i$, описывает характеристики гонца в $(i + 1)$ -ом городе: S_i - это количество минут для подготовки к путешествию, а V_i - это количество минут, необходимое для прохождения одного километра ($0 \leq S_i \leq 10^9$, $1 \leq V_i \leq 10^9$). В столице нет гонца.

Формат выходных данных

Выходные данные должны состоять ровно из $N - 1$ целых чисел. Число i -это минимальное время в минутах, необходимое для отправки сообщения из города $(i + 1)$ в столицу.

Примеры

стандартный ввод	стандартный вывод
5	206 321 542 328
1 2 20	
2 3 12	
2 4 1	
4 5 3	
26 9	
1 10	
500 2	
2 30	

Задача E. А ты сортируешь запросы?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 1024 мегабайта

Благодарим DittuT за предоставленную задачу.

Вам даётся N запросов, которые бывают трёх типов:

1. Добавить пару чисел (a, b) в набор. $(-10^9 \leq a, b \leq 10^9)$
2. Удалить пару добавленную в запросе $index$. (Запросы нумеруются числами от 1 до N в порядке в котором они идут вводятся в программу.)
3. Вам даётся число A и требуется найти максимум $a_i \cdot A + b_i$ по всем парам в наборе. $(-10^9 \leq A \leq 10^9)$. Гарантируется, что в наборе есть хоть одна пара в этот момент.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 3 \cdot 10^5$) - количество запросов.

Каждая из следующих n строк начинается с целого числа op ($1 \leq op \leq 3$) - типа запроса. Далее, в зависимости от типа запроса, следуют, либо числа a_i и b_i , либо число $index$, либо число A .

Формат выходных данных

Для каждого запроса типа 3 выведите в отдельной строке максимальное значение $a \cdot A + b$. Гарантируется, что в наборе есть хоть одна пара в этот момент.

Примеры

стандартный ввод	стандартный вывод
4	-1000000000000000000
1 -1000000000 0	-999999999999999999
3 1000000000	
1 -1000000000 1	
3 1000000000	

Задача F. Задача без легенды

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

Дан неубывающий массив a размера n , а также стоимостей элементов c . Вам нужно выбрать подпоследовательность (возможно, пустую) с минимальной стоимостью и вывести эту стоимость. Стоимость подпоследовательности из элементов i_1, i_2, \dots, i_k это сумма стоимостей ее элементов плюс сумма «расстояний» между ее соседними элементами. «Расстояние» между элементами i и j , ($i < j$) — это $(7a_j - 4a_i)(5a_j - 2a_i)(4a_j - a_i)$.

Формат входных данных

В первой строке вводится натуральное число n ($1 \leq n \leq 10^6$). Во второй строке вводится массив целых чисел a ($0 \leq a_i \leq 10^6$). В третьей строке вводится массив целых чисел c ($-10^{18} \leq c_i \leq 10^{18}$)

Формат выходных данных

Выведите одно число — минимальную стоимость подпоследовательности.

Примеры

стандартный ввод	стандартный вывод
2 0 1 2 -6	-6
3 1 2 4 -6 5 1	-6
5 1 2 2 2 4 0 -2 -1 -5 2	-5
4 1 1 3 5 7 -10 8 -7	-10
5 0 1 4 4 4 -8 2 10 4 5	-8
2 1 1 -100 -100	-173

Замечание

Так как ответ может быть очень большим, рекомендуется использовать `__int128` или еще что-нибудь.

Задача G. Парные дороги (easy)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 8 секунд
Ограничение по памяти: 512 мегабайт

В некотором государстве есть n городов, между которыми пока что не построено ни одной дороги. В i -м городе живет w_i человек. Так же есть $n - 1$ дорога, которую можно построить. i -я дорога при её строительстве соединит города u_i и v_i , а стоимость её строительства равна s_i .

Известно, что если построить все дороги, то от каждого города можно будет добраться до любого другого, используя только эти дороги. Другими словами, дороги образуют дерево.

В каждый из следующих k дней будет происходить следующее: В i -й день выбирается некоторый город c_i , а также две различные ещё не построенные дороги, соединяющие этот город с какими-то другими городами. После этого эти две дороги строятся, за это платится цена, равная суммарной стоимости строительства этих двух дорог. Город c_i в этом случае является *центральным* в день i .

Через k дней каждый город, являющийся *центральным* в хотя-бы один из k дней, принесёт ровно столько прибыли, сколько человек в нём живёт.

Выгодой назовём разность между прибылью, которую принесут города, и суммарной стоимостью построенных дорог. Найдите максимально возможную *выгоду*.

Формат входных данных

Первая строка содержит три целых числа n , k и t ($3 \leq n \leq 200\,000$, $1 \leq k \leq \frac{n-1}{2}$, $t = 0$) — количество городов, количество пар дорог, которые надо построить, а также число t равное 1, если нужно найти какие именно дороги надо построить, и равное 0 иначе.

Вторая строка содержит n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^8$) — количество жителей в городах.

Следующие $n - 1$ строк содержат по три целых числа u_i, v_i и s_i ($1 \leq u_i, v_i \leq n$, $1 \leq s_i \leq 10^8$) — номера городов, которые соединяет i -я дорога и стоимость её строительства.

Гарантируется, что дороги образуют дерево, и что можно построить k пар дорог, удовлетворяющих условию задачи.

Формат выходных данных

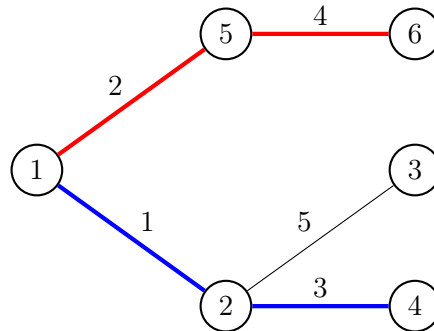
В первой строке выведите единственное целое число — максимальную *выгоду*, которую можно получить после строительства ровно k пар дорог.

Примеры

стандартный ввод	стандартный вывод
<pre>6 2 0 1 2 3 4 5 6 1 2 1 2 3 5 2 4 3 1 5 2 5 6 4</pre>	-3
<pre>8 3 0 4 5 1 2 3 1 3 5 2 1 15 7 1 5 4 8 1 8 5 2 7 8 1 6 7 5 3 7 7</pre>	-13

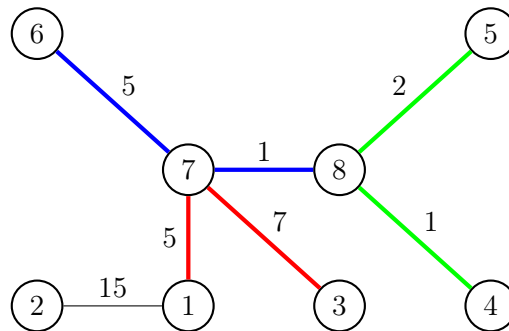
Замечание

В первом тесте из условия предлагается построить следующие дороги (одинаковым цветом отмечены дороги из одной пары):



Суммарная стоимость строительства этих дорог равна $2 + 4 + 1 + 3 = 10$. После строительства, города 2 и 5 принесут по 2 и 5 прибыли соответственно. Таким образом, *выгода* равна $2 + 5 - 10 = -3$. Можно показать, что лучший ответ получить нельзя.

Во втором тесте из условия предлагается построить следующие дороги:



Суммарная стоимость строительства этих дорог равна $5 + 1 + 2 + 1 + 5 + 7 = 21$. После строительства, города 7 и 8 принесут по 3 и 5 прибыли соответственно. Таким образом, *выгода* равна $3 + 5 - 21 = -13$. Можно показать, что лучший ответ получить нельзя.

Задача Н. Парные дороги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	8 секунд
Ограничение по памяти:	512 мегабайт

В некотором государстве есть n городов, между которыми пока что не построено ни одной дороги. В i -м городе живет w_i человек. Так же есть $n - 1$ дорога, которую можно построить. i -я дорога при её строительстве соединит города u_i и v_i , а стоимость её строительства равна s_i .

Известно, что если построить все дороги, то от каждого города можно будет добраться до любого другого, используя только эти дороги. Другими словами, дороги образуют дерево.

В каждый из следующих k дней будет происходить следующее: В i -й день выбирается некоторый город c_i , а также две различные ещё не построенные дороги, соединяющие этот город с какими-то другими городами. После этого эти две дороги строятся, за это платится цена, равная суммарной стоимости строительства этих двух дорог. Город c_i в этом случае является *центральным* в день i .

Через k дней каждый город, являющийся *центральным* в хотя-бы один из k дней, принесёт ровно столько прибыли, сколько человек в нём живёт.

Выгодой назовём разность между прибылью, которую принесут города, и суммарной стоимостью построенных дорог. Найдите максимально возможную *выгоду*.

Формат входных данных

Первая строка содержит три целых числа n , k и t ($3 \leq n \leq 200\,000$, $1 \leq k \leq \frac{n-1}{2}$, $0 \leq t \leq 1$) — количество городов, количество пар дорог, которые надо построить, а также число t равное 1, если нужно найти какие именно дороги надо построить, и равное 0 иначе.

Вторая строка содержит n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^8$) — количество жителей в городах.

Следующие $n - 1$ строк содержат по три целых числа u_i, v_i и s_i ($1 \leq u_i, v_i \leq n$, $1 \leq s_i \leq 10^8$) — номера городов, которые соединяет i -я дорога и стоимость её строительства.

Гарантируется, что дороги образуют дерево, и что можно построить k пар дорог, удовлетворяющих условию задачи.

Формат выходных данных

В первой строке выведите единственное целое число — максимальную *выгоду*, которую можно получить после строительства ровно k пар дорог.

Если $t = 1$, то в следующих k строках выведите k троек чисел c_i, x_i и y_i ($1 \leq c_i, x_i, y_i \leq n$) — описание пар дорог, которые надо построить. Такая тройка задаёт пару из дороги (c_i, x_i) и дороги (c_i, y_i) .

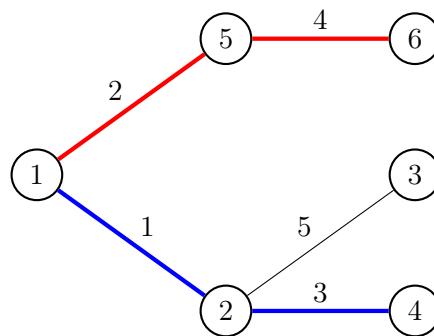
Если существует несколько способов получить максимальную *выгоду*, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
6 2 1 1 2 3 4 5 6 1 2 1 2 3 5 2 4 3 1 5 2 5 6 4	-3 5 6 1 2 4 1
8 3 0 4 5 1 2 3 1 3 5 2 1 15 7 1 5 4 8 1 8 5 2 7 8 1 6 7 5 3 7 7	-13

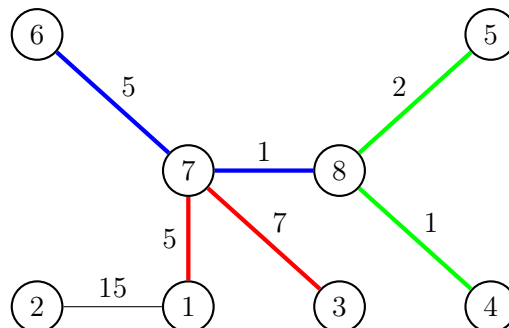
Замечание

В первом тесте из условия предлагается построить следующие дороги (одинаковым цветом отмечены дороги из одной пары):



Суммарная стоимость строительства этих дорог равна $2 + 4 + 1 + 3 = 10$. После строительства, города 2 и 5 принесут по 2 и 5 прибыли соответственно. Таким образом, *выгода* равна $2 + 5 - 10 = -3$. Можно показать, что лучший ответ получить нельзя.

Во втором тесте из условия предлагается построить следующие дороги:



Суммарная стоимость строительства этих дорог равна $5 + 1 + 2 + 1 + 5 + 7 = 21$. После строительства, города 7 и 8 принесут по 3 и 5 прибыли соответственно. Таким образом, *выгода* равна $3 + 5 - 21 = -13$. Можно показать, что лучший ответ получить нельзя.