

## Задача А. Калила и Димна на лесозаготовках

Имя входного файла: `lumber.in`  
Имя выходного файла: `lumber.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

*Спонсор сегодняшней задачи — codeforces round 189. Codeforces — мечты сбываются!*

Калила и Димна — два шакала. Они живут в огромных джунглях. Однажды шакалы решили устроиться на завод лесозаготовки и подработать.

Управляющий завода хочет, чтобы они отправились в джунгли и срубили  $n$  деревьев высотой  $a_1, a_2, \dots, a_n$ . Для этого Калила и Димна купили цепную пилу в магазине. Каждый раз, когда они используют пилу на дереве номер  $i$ , они уменьшают высоту этого дерева на единицу. Каждый раз Калила и Димна должны заправить пилу для использования. Цена заправки зависит от того, какие деревья полностью спилены (дерево считается полностью спиленным, если его высота равна 0). Если максимальный идентификатор полностью срубленного дерева равняется  $i$  (первоначально это дерево имело высоту  $a_i$ ), то цена заправки пилы равняется  $b_i$ . Если ни одно дерево не срублено полностью, то заправлять пилу запрещается. Изначально пила заправлена. Известно, что для каждого  $i < j$ ,  $a_i < a_j$  и  $b_i > b_j$ , а также  $b_n = 0$  и  $a_1 = 1$ .

Калила и Димна хотят полностью срубить все деревья с минимальными затратами. Они ждут Вашей помощи! Поможете?

### Формат входных данных

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10^5$ ). Во второй строке записано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ). В третьей строке записано  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^9$ ).

Гарантируется, что  $a_1 = 1$ ,  $b_n = 0$ ,  $a_1 < a_2 < \dots < a_n$  и  $b_1 > b_2 > \dots > b_n$ .

### Формат выходных данных

В единственной строке должна быть записана минимальная стоимость вырубания всех деревьев.

### Примеры

<code>lumber.in</code>	<code>lumber.out</code>
5 1 2 3 4 5 5 4 3 2 0	25
6 1 2 3 10 20 30 6 5 4 3 2 0	138

## Задача В. Сiel и гондолы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Лиса Сiel зашла в парк аттракционов. И вот, она в очереди на колесо обозрения. В очереди стоит  $n$  людей (хотя нет, скорее лис): мы будем считать, что первая лиса стоит в начале очереди, а  $n$ -я лиса стоит в хвосте очереди.

Всего имеется  $k$  гондол, мы распределяем лис по гондолам следующим образом:

- Когда подплывает первая гондола,  $q_1$  лис переходят из начала очереди в подплывшую гондолу.
- Затем, когда подплывает вторая гондола,  $q_2$  лис из начала оставшейся очереди переходит в эту гондолу.
- ...
- Оставшиеся  $q_k$  лис идут с последней ( $k$ -ю) гондолу.

Обратите внимание, что числа  $q_1, q_2, \dots, q_k$  должны быть положительными. Из условия следует, что  $\sum_{i=1}^k q_i = n$  и  $q_i > 0$ .

Вы знаете как лисам не хочется задерживаться в гондолах с незнакомцами. Итак, Ваша задача — найти оптимальный способ размещения (то есть определить оптимальную последовательность  $q$ ), чтобы угодить всем. Для каждой пары лис  $i$  и  $j$  задано значение  $u_{ij}$ , обозначающее степень незнакомости. Можете считать, что  $u_{ij} = u_{ji}$  для всех  $i, j$  ( $1 \leq i, j \leq n$ ) и что  $u_{ii} = 0$  для всех  $i$  ( $1 \leq i \leq n$ ). Тогда значение незнакомости в гондоле определяется как сумма значений незнакомости между всеми парами лис, которые находятся в этой гондоле. Общее значение незнакомости определяется как сумма значений незнакомости по всем гондолам.

Помогите лисе Сiel найти минимальное возможное значение общей незнакомости при некотором оптимальном распределении лис по гондолам.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 4000$  and  $1 \leq k \leq \min(n, 800)$ ) — количество лис в очереди и количество гондол. В следующих  $n$  строках записано по  $n$  целых чисел — матрица  $u$ , ( $0 \leq u_{ij} \leq 9$ ,  $u_{ij} = u_{ji}$  и  $u_{ii} = 0$ ).

Пожалуйста, используйте методы быстрого чтения (например, для Java используйте `BufferedReader` вместо `Scanner`).

### Формат выходных данных

Выведите целое число — минимальное возможное значение общей незнакомости.

## Примеры

стандартный ввод	стандартный вывод
5 2 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0	0
8 3 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0	7

## Замечание

В первом примере можно распределить лис вот так: 1, 2 идут в одну гондолу, 3, 4, 5 идут в другую гондолу.

Во втором примере оптимальное распределение таково: 1, 2, 3 | 4, 5, 6 | 7, 8.

## 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. Утилитаризм

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 7 секунд  
Ограничение по памяти: 256 мегабайт

В британском королевстве есть  $n$  городов, пронумерованных от 1 до  $n$ . Некоторые города связаны двусторонними дорогами. Всего есть ровно  $n-1$  дорога, и между любыми двумя городами есть ровно один путь. Также с каждой дорогой связана некоторая ценность.

Сегодня, чтобы почтить отцов-основателей (их было, кстати,  $k$ ) королевства, нынешний король Лелуш решил выбрать  $k$  различных дорог и подарить по одной дороге каждому из основателей. Чтобы избежать ненужных конфликтов, требуется, чтобы дороги не касались своими концами.

На самом деле, Лелушу не так-то и важно, кто получит какую дорогу. С другой стороны, его волнует суммарная ценность подарочных дорог. Вам нужно помочь Лелушу и выбрать  $k$  дорог, максимизируя эту суммарную ценность.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $2 \leq n \leq 250\,000$ ,  $1 \leq k \leq n-1$ ) — количество городов и количество дорог, которые нужно подарить.

Каждая из следующих  $n-1$  содержит три целых числа  $v_i, u_i, c_i$  ( $1 \leq v_i, u_i \leq n$ ,  $-10^6 \leq c_i \leq 10^6$ ) — концы очередной дороги и её ценность.

### Формат выходных данных

Если раздать дороги нельзя, выведите «Impossible».

Иначе выведите одно число — максимальную суммарную цену  $k$  дорог.

### Примеры

стандартный ввод	стандартный вывод
5 1 1 2 2 2 3 3 2 4 10 4 5 6	10
5 2 1 2 2 2 3 3 2 4 10 4 5 6	9
5 3 1 2 2 2 3 3 2 4 10 4 5 6	Impossible

## Задача Е. Транспортировка кошек

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

Zxc960115 содержит большое хозяйство. Он кормит  $m$  милых кошечек и держит у себя  $p$  кормильщиков. Через ферму проходит прямая дорога, а вдоль дороги расположено  $n$  холмов, пронумерованных от 1 до  $n$ , слева направо. Расстояние от холма  $i$  до  $i-1$  равняется  $d_i$  метров. Кормильщики живут на холме 1.

Однажды кошечкам захотелось порезвиться и они разбежались. Кошка  $i$  пошла к холму  $h_i$ , дошла до него в момент времени  $t_i$ , а затем стала ждать кормильщика на холме  $h_i$ . Кормильщики должны собрать всех разбежавшихся кошек. Каждый кормильщик идет прямо от холма номер 1 до холма номер  $n$ , не останавливаясь у какого-либо холма, и собирает всех кошек, **ожидающих** на каждом холме. Кормильщики двигаются со скоростью 1 в единицу времени и достаточно сильны, чтобы собрать сколько угодно кошек.

Например, пусть имеется два холма ( $d_2 = 1$ ) и одна кошечка, которая дошла до холма 2 ( $h_1 = 2$ ) в момент времени 3. Тогда, если кормильщик отправится за кошками от холма 1 в момент времени 2 или 3, то он сможет забрать эту кошку. Но если он отправится от холма 1 в момент времени 1, то он не сможет этого сделать. Если кормильщик отправится за кошкой в момент времени 2, то кошка будет ждать его 0 единиц времени, если же он отправится в момент времени 3, то кошка будет ждать его 1 единицу времени.

Ваша задача — составить расписание отправки от холма 1 для кормильщиков так, чтобы общее время ожидания кошек до того как их заберут было минимальным.

### Формат входных данных

В первой строке входных данных содержится три целых числа  $n, m, p$  ( $2 \leq n \leq 10^5, 1 \leq m \leq 10^5, 1 \leq p \leq 100$ ).

Во второй строке содержится  $n - 1$  положительных целых чисел  $d_2, d_3, \dots, d_n$  ( $1 \leq d_i < 10^4$ ).

В каждой из следующих  $m$  строк содержится по два целых числа  $h_i$  и  $t_i$  ( $1 \leq h_i \leq n, 0 \leq t_i \leq 10^9$ ).

### Формат выходных данных

Выведите целое число, минимальную сумму времен ожидания всех кошек.

### Примеры

стандартный ввод	стандартный вывод
4 6 2 1 3 5 1 0 2 1 4 9 1 10 2 10 3 12	3

## Задача F. Петя и прямоугольники

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

Маленький Петя очень любит прямоугольники. Петя дал маме список прямоугольников, которые он хочет получить в подарок на Новый Год. Каждый прямоугольник характеризуется  $w$  и высотой  $h$ . Мама хочет сделать Пете приятное и купить все прямоугольники из его списка. Мама отправилась в магазин и узнала, что цена одного прямоугольника равна его площади. К ее счастью, в магазине действует предновогодняя акция, позволяющая покупать прямоугольники не по одному, а сразу наборами. Стоимость одного набора равна ширине самого широкого прямоугольника, умноженной на высоту самого высокого прямоугольника из этого набора. Обратите внимание, что поворачивать прямоугольники (тем самым меняя местами ширину и высоту) нельзя. Помогите маме Пети купить все прямоугольники из списка ее сына, потратив на это наименьшее количество денег.

### Формат входных данных

В первой строке записано число  $N$  ( $1 \leq N \leq 200\,000$ ) — количество прямоугольников в списке Пети. В каждой из следующих  $N$  строк записаны по 2 целых положительных числа, не превышающих  $10^6$  — ширина и высота очередного прямоугольника.

### Формат выходных данных

Выведите одно число — наименьшее количество денег, которое может потратить мама чтобы купить Пете все прямоугольники из его списка.

### Примеры

стандартный ввод	стандартный вывод
4 100 1 15 15 20 5 1 100	500
5 1 10 2 20 3 30 4 40 10 1	170

## Задача G. Коды, сохраняющие порядок

Имя входного файла: `codes.in`  
Имя выходного файла: `codes.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Двоичный код — это код, где каждому символу сопоставляется последовательность из единиц и нулей. Код называется префиксным, если ни одно кодовое слово не является префиксом другого. Код называется сохраняющим порядок, если лексикографический порядок кодовых слов совпадает с алфавитным порядком символов.

Рассмотрим текст над алфавитом, содержащим  $n$  символов, в котором  $a_1$  раз встречается первый символ,  $a_2$  раз встречается второй символ,  $\dots$ ,  $a_n$  раз встречается  $n$ -й символ. Длина текста после кодирования его префиксным кодом, где первому символу сопоставлена строка длины  $l_1$ , второму — строка длины  $l_2$ , и т. д., будет равна  $a_1 \cdot l_1 + a_2 \cdot l_2 + \dots + a_n \cdot l_n$ .

Требуется найти сохраняющий порядок префиксный код, минимизирующий длину закодированного текста.

### Формат входных данных

Первая строка содержит число  $n$  — число символов в алфавите ( $2 \leq n \leq 2000$ ). Следующая строка содержит  $n$  целых чисел — сколько раз каждый символ встречается в тексте:  $a_1, a_2, \dots, a_n$ . Числа положительные и не превосходят  $10^9$ .

### Формат выходных данных

Выведите  $n$  двоичных последовательностей — искомый код.

### Примеры

<code>codes.in</code>	<code>codes.out</code>
5	00
1 8 2 3 1	01
	10
	110
	111

## Задача Н. Реконструкция дорог

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

Байтазар — король Байтландии, любит улучшать дорожную инфраструктуру в своей стране. В Байтландии  $n$  городов, пронумерованных от 1 до  $n$ . Байттаун, столица Байтландии, имеет номер 1. Города соединены сетью из  $m$  двусторонних проселочных дорог. До каждого города можно добраться от столицы.

По плану Байтазара следует построить несколько качественных шоссе, выполнив реконструкцию некоторых проселочных дорог. После строительства шоссе жители Байтландии должны иметь возможность добраться от любого города до любого другого по шоссе.

Для каждой дороги известна стоимость ее реконструкции, и Байтазар хотел бы минимизировать стоимость шоссе. Но жители столицы воспротивились строительству шоссе: ведь пробки в столице существенно увеличатся. Чтобы контролировать ситуацию, решено было, что из дорог, выходящих непосредственно из столицы, реконструировано будет не более чем  $d$ .

Помогите Байтазару понять, как организовать строительство.

### Формат входных данных

Первая строка ввода содержит три числа:  $n$ ,  $m$  и  $d$  ( $1 \leq d \leq n \leq 2000$ ,  $0 \leq m \leq n(n-1)/2$ ).

Далее следует описание существующих дорог. Каждое описание состоит из трех чисел:  $a$ ,  $b$ ,  $c$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ,  $1 \leq c \leq 10^9$ ) — номера городов, соединенных дорогой, и стоимость реконструкции этой дороги. Между двумя городами проложено не более одной дороги.

### Формат выходных данных

Выведите одно число — минимальную стоимость реконструкции.

Гарантируется, что хотя бы один способ осуществить реконструкцию существует.

### Примеры

стандартный ввод	стандартный вывод
5 6 2 1 2 2 1 3 1 1 4 2 1 5 1 4 5 10 2 3 4	16