

## Задача А. 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 В. 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 С. Оптимальное бинарное дерево поиска

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

Рассмотрим множество  $S = \{e_1, e_2, \dots, e_n\}$ , состоящее из  $n$  различных элементов таких, что  $e_1 < e_2 < \dots < e_n$ . Рассмотрим бинарное дерево поиска, состоящее из элементов  $S$ . Чем чаще производится запрос к элементу, тем ближе он должен располагаться к корню. Стоимостью  $cost$  доступа к элементу  $e_i$  из  $S$  в дереве будем называть значение  $cost(e_i)$ , равное числу ребер на пути, который соединяет корень с вершиной, содержащей элемент. Имея частоту запросов к элементам из  $S$  —  $f(e_1), f(e_2), \dots, f(e_n)$  — определим общую стоимость дерева следующим образом:

$$\sum_{i=1}^n f(e_i) \cdot cost(e_i)$$

Дерево, имеющее наименьшую стоимость, считается наилучшим для поиска элементов из  $S$ . Именно поэтому оно называется Оптимальным Бинарным Деревом Поиска. Ваша задача — найти стоимость Оптимального Бинарного Дерева Поиска.

### Формат входных данных

Состоит из нескольких тестов, каждый из которых расположен в отдельной строке. Первое число в строке  $n$  ( $1 \leq n \leq 5000$ ) указывает на размер множества  $S$ . Следующие  $n$  неотрицательных целых чисел описывают частоты запросов элементов из  $S$ :  $f(e_1), f(e_2), \dots, f(e_n)$ . Известно, что  $0 \leq f(e_i) \leq 100$ . Сумма  $n$  по всем тестам не больше 5000.

### Формат выходных данных

Для каждого теста в отдельной строке выведите стоимость Оптимального Бинарного Дерева Поиска.

### Пример

стандартный ввод	стандартный вывод
1 5	0
3 10 10 10	20
3 5 10 20	20
6 1 3 5 10 20 30	63

## Задача D. 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 E. Ciel и гондолы

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

Лиса Ciel 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 Ciel 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 F. 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 Г. 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 Н. Жоские диски

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

У вас есть кластер. В нем есть  $n$  пар жестких дисков (HDD). Каждый диск может быть представлен как точка с целой координатой на бесконечной прямой. В каждой паре один из HDD главный, а второй — запасной.

Вы хотите настроить  $k$  компьютеров, которые тоже представляются как точки с целыми координатами на той же прямой, после чего соединить все диски с компьютерами при помощи проводов. В конечном итоге, основной и запасной диски из одной пары должны быть соединены с одним и тем же компьютером (и только с ним), но каждый компьютер может быть соединен с любым количеством дисков (возможно с нулем). Каждый провод соединяет один HDD с одним компьютером, и длина провода равна расстоянию между соответствующими точками на прямой. Необходимо найти минимальную суммарную длину всех проводов.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $k$  — количество пар дисков и количество компьютеров, соответственно ( $2 \leq k \leq n \leq 100\,000$ ;  $4 \leq k \times n \leq 100\,000$ ). Каждая из следующих  $n$  строк содержат по два целых числа  $a_i$  и  $b_i$  — координаты основного и запасного дисков ( $-10^9 \leq a_i, b_i \leq 10^9$ ).

### Формат выходных данных

Вывод должен содержать ровно одно число — ответ на задачу.

### Примеры

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

### Замечание

В примере оптимальные позиции компьютеров — 0 и 6. Соединим вторую и третью пары дисков с первым компьютером, а остальные со вторым. Тогда суммарная длина проводов, ведущих к первому компьютеру, будет 3, и 10 для второго, что в сумме дает 13.