

## Задача А. Петя и прямоугольники

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
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  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — 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.

### Пример

стандартный ввод	стандартный вывод
4 100 1 15 15 20 5 1 100	500

## Задача В. Коды, сохраняющие порядок

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

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.

### Пример

стандартный ввод	стандартный вывод
5	00
1 8 2 3 1	01
	10
	110
	111

## Задача С. Автобусные остановки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

В деревне есть  $n$  домов, расположенных вдоль главной дороги, которую можно воспринимать как числовую прямую.  $i$ -й дом имеет координату  $x_i$ .

Жители предпочитают автобусные остановки рядом с их домом, и чем дальше автобусная остановка, тем более несчастливы они. *Недовольство* дома определяется как квадрат расстояния между домом и ближайшей к нему автобусной остановкой. Ваша задача — построить  $k$  автобусных остановок вдоль главной дороги так, чтобы сумма недовольств домов была минимальна.

*Обратите внимание:* остановка может быть построена в любой точке числовой прямой, необязательно совпадающей с точкой какого-то из домов.

Формально, пусть ближайшая остановка к  $i$ -му дому находится в точке  $p_i$ . Тогда вы хотите минимизировать:

$$\sum_{i=1}^n |x_i - p_i|^2$$

### Формат входных данных

Первая строка содержит два целых числа  $n, k$  ( $1 \leq n \leq 5 \cdot 10^4$ ,  $1 \leq k \leq \min(n, 100)$ ).

Вторая строка содержит  $n$  целых чисел  $x_i$  ( $1 \leq x_i \leq 10^5$ ).

### Формат выходных данных

Выведите одно число — ответ с относительной или абсолютной погрешностью не более  $10^{-6}$ .

### Пример

стандартный ввод	стандартный вывод
3 2 1 2 4	0.5000000000000000

### Замечание

Пусть построили автобусные остановки в координатах 1.5 и 4.0. Тогда:

- Недовольство первого дома:  $(x_1 - p_1)^2 = (1.0 - 1.5)^2 = 0.25$
- Недовольство второго дома:  $(x_2 - p_1)^2 = (2.0 - 1.5)^2 = 0.25$
- Недовольство третьего дома:  $(x_3 - p_2)^2 = (4.0 - 4.0)^2 = 0.00$

Таким образом, суммарное недовольство равно  $0.25 + 0.25 + 0.00 = 0.5$

## Задача D. Задача «ИЛИ»

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Дан массив  $a$  из  $n$  целых чисел. Стоимость отрезка массива — побитовое «ИЛИ» его элементов. Пусть вы разбили массив на  $k$  непустых отрезков и посчитали сумму их стоимостей. Какое максимальное значение вы можете получить?

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 2 \cdot 10^5$ ).

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 2^{30}$ ) — элементы массива  $a$ .

### Формат выходных данных

Выведите одно целое число — ответ на задачу.

### Пример

стандартный ввод	стандартный вывод
5 2 1 4 3 4 8	20

## Задача Е. Честный дележ

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

У вас есть массив неотрицательных целых чисел  $a_1, a_2, \dots, a_n$ .

Вам нужно разделить его на  $k$  непустых подотрезков:  $[1; b_1], [b_1 + 1; b_2], \dots, [b_{k-1} + 1; n]$ .

Обозначим сумму на  $i$ -м отрезке как  $s_i$  и максимум на  $i$ -м отрезке как  $m_i$ . Ваша задача сделать  $|s_i - s_{i+1}| \leq \max(m_i, m_{i+1})$  для всех  $1 \leq i \leq k - 1$ .

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $k$ : размер массива и необходимое количество отрезков ( $3 \leq k \leq n \leq 100\,000$ ).

В следующей строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$ : данный массив ( $0 \leq a_i \leq 50\,000$ ).

### Формат выходных данных

Если разделить массив указанным образом возможно, выведите “Yes” на первой строке и  $k - 1$  целых чисел  $b_1, b_2, \dots, b_{k-1}$ , разделенных пробелами на второй строке.

Числа должны удовлетворять  $1 \leq b_1 < b_2 < \dots < b_{k-1} < n$ .

Также неравенства  $|s_i - s_{i+1}| \leq \max(m_i, m_{i+1})$  должны быть выполнены для всех  $1 \leq i \leq k - 1$ .

Если существует несколько возможных решений, выведите любое.

Если разделение невозможно, выведите “No” в единственной строке.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 3	Yes
17 18 17 30 35	2 4

## Задача F. 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 G. Жесткие диски

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

### Пример

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.



## Задача Н. Гладкие числа

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

Назовем число гладким, если его цифры, начиная со старшего разряда, образуют неубывающую последовательность. Упорядочим все такие числа в возрастающем порядке и присвоим каждому номер. Вам требуется по номеру  $N$  вывести  $N$ -ое гладкое число.

### Формат входных данных

На вход программы поступает номер  $N$  ( $1 \leq N \leq 2147483647$ ).

### Формат выходных данных

Выведите соответствующее номеру  $N$  гладкое число.

### Примеры

стандартный ввод	стандартный вывод
3	3
11	12

## Задача I. Не выделяйся

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

Дан массив целых неотрицательных чисел  $a$  длины  $n$  и число  $k$ . За одну операцию можно изменить любой элемент массива на 1. Необходимо сделать так, чтобы разница всех соседних элементов массива  $a$  была не больше  $k$  по абсолютному значению, за минимальное количество операций. При этом делать элементы массива отрицательными запрещено.

### Формат входных данных

В первой строке входных данных даны два целых неотрицательных числа  $n$  и  $k$  ( $1 \leq n \leq 200\,000$ ,  $0 \leq k \leq 10^9$ ) — длина массива  $a$  и максимальная допустимая разность соседних элементов.

Во второй строке входных данных даны  $n$  целых неотрицательных чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — элементы массива  $a$ .

### Формат выходных данных

Выведите одно число — минимальное количество операций, которые надо сделать, чтобы абсолютное значение разницы любых соседних элементов массива было не больше  $k$ .

### Примеры

стандартный ввод	стандартный вывод
6 1 2 10 0 2 4 3	10
6 3 2 10 2 6 4 3	6
4 1 1 4 1 4	4
10 1 10 9 8 7 6 5 4 3 2 1	0
3 0 1 1 3	2

## Задача J. Транспортировка кошек

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

Zxr960115 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.

### Пример

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

## Задача К. Красные и синие лампочки

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

Есть  $n$  лампочек с номерами  $1, 2, \dots, n$ . Надо ровно  $m$  из них сделать красными, а все остальные — синими, чтобы максимизировать следующую величину: по всем  $i \in [1, n - 1]$ : если лампочки  $i$  и  $i + 1$  разного цвета, то добавить к ответу  $a_i$ .

### Формат входных данных

Первая строка содержит два числа —  $n$  и  $m$  ( $2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq n - 1$ ).

Вторая строка содержит  $n - 1$  число  $a_1, a_2, \dots, a_{n-1}$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите максимальную сумму, которую можно получить, если ровно  $m$  лампочек красные.

### Примеры

стандартный ввод	стандартный вывод
6 2 3 1 4 1 5	11
7 6 2 7 1 8 2 8	10
11 7 12345 678 90123 45678901 234567 89012 3456 78901 23456 7890	46207983

## Задача L. Две лесопилки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
Ограничение по памяти: 64 мегабайта

От вершины до подножья холма растет  $N$  старых деревьев. Районная администрация решила в санитарных целях срубить эти деревья, а чтобы снизить стоимость мероприятия перевезти все древесину на лесопилки. Деревья могут быть перевезены только в одном направлении – вниз. У подножья холма находится лесопилка, а также две дополнительные лесопилки могут быть построены на холме вдоль дороги. Вам предстоит определить, где наиболее выгодно построить эти лесопилки, чтобы минимизировать стоимость транспортировки древесины. Перевозка 1 килограмма древесины на 1 метр стоит 1 копейку.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $N$  – количество деревьев ( $1 \leq N \leq 20\,000$ ). Деревья занумерованы от 1 до  $N$  начиная с вершины холма. Следующие  $N$  линий содержат по два целых числа  $w_i$  и  $d_i$  ( $1 \leq w_i, d_i \leq 10\,000$ ) – вес дерева номер  $i$  и расстояние между деревьями  $i$  и  $i+1$ . Последнее из этих чисел ( $d_n$ ) задает расстояние от нижнего дерева до лесопилки.

### Формат выходных данных

Выведите единственное число – минимальную стоимость сплава деревьев вниз по реке.

### Пример

стандартный ввод	стандартный вывод
9	26
1 2	
2 1	
3 3	
1 1	
3 2	
1 6	
2 1	
1 2	
1 1	

### Замечание

В примере выгодно поставить лесопилки у деревьев с номерами 3 и 6.

## Задача M. Civil Voting

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

Мэр Байтсбурга планирует построить трамвайную линию вдоль главной улицы города.

Вдоль улицы были выделены  $N$  мест, и мэр должен выбрать некоторые из данных мест для строительства остановок. Уже было решено, что первое и  $N$ -е место точно будут выбраны для строительства. Для каждого места известны два параметра:  $x_i$  — расстояние от  $i$ -го места до начала улицы, а  $c_i$  — уровень неудовлетворенности жителей города, если в данном месте будет построена остановка.

Для выбора мест строительства остановок были выбраны  $m$  жителей города, среди которых был произведен опрос. Психологи выяснили, что  $i$ -й житель крайне не любит число  $d_i$ , так как считает данное число слишком банальным, поэтому:

- уровень удовлетворенности жителя равен сумме уровней удовлетворенности для каждых двух соседних построенных остановок;
- уровень удовлетворенности для соседних остановок, находящихся на расстоянии  $d$  друг от друга, равен  $|d - d_i|$ .

Мэр вычислил суммарную удовлетворенность горожан, просуммировав удовлетворенность по всем выбранным жителям и вычел из данного числа неудовлетворенность для всех мест, в которых было решено построить остановку. Выведите максимальную суммарную удовлетворенность, которую можно достичь, оптимально выбрав места для постройки остановок.

### Формат входных данных

В первой строке записаны два числа  $n$  ( $2 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^5$ ) — количество мест для строительства станций и выбранных горожан, соответственно.

Во второй строке записаны  $m$  чисел  $d_i$  ( $0 \leq d_i \leq 10^7$ ) — параметры выбранных жителей.

В каждой из следующих  $n$  строк записаны числа  $x_i$  ( $0 \leq x_i \leq 10^7$ ) и  $c_i$  ( $0 \leq |c_i| \leq 10^{12}$ ).

Гарантируется, что  $x_1 < x_2 < \dots < x_n$ .

### Формат выходных данных

Выведите одно число — максимальную суммарную удовлетворенность жителей города.

## Примеры

стандартный ввод	стандартный вывод
2 1 10 0 5 20 3	2
3 3 3 7 10 2 20 5 4 10 -3	-1
9 5 30 64 2 93 67 0 81 1 256 6 251 13 256 23 180 52 256 72 94 77 256 97 12	137

## Задача N. Dynamic Skyline

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Для того, чтобы не сразу испугать вас, рассмотрим сначала классическую задачу. Вам даны  $n$  функций  $f_1(x), f_2(x), \dots, f_n(x)$ , где  $f_i(x) = a_i \cdot x + b_i$ . Теперь вы хотите найти минимальное значение  $f_i(x)$  для всех  $i$  при некотором фиксированном значении  $x$ . Данная задача легко решается при помощи Convex Hull Trick.

Совсем не страшно, правда? А теперь перейдем к задаче. Вам даны  $n$  функций  $f_1(x), f_2(x), \dots, f_n(x)$ , где  $f_i(x) = (x - a_i)^4 + b_i$ .

От вас требуется ответить на  $m$  запросов. Каждый запрос бывает одного из трех типов:

- «1  $a$   $b$ » ( $1 \leq a \leq 50\,000$ ,  $1 \leq b \leq 10^{18}$ ) — добавить новую функцию  $f_{n+1}(x) = (x - a)^4 + b$ , после чего заменить значение  $n$  на  $n + 1$ .
- «2  $t$ » ( $1 \leq t \leq n$ ) — удалить функцию  $f_t(x)$ . Гарантируется, что каждая функция будет удалена не более одного раза.
- «3  $x$ » ( $1 \leq x \leq 50\,000$ ) — вычислить минимальное значение функций  $f_i(x)$ , где  $1 \leq i \leq n$  и функция  $f_i(x)$  не была удалена.

### Формат входных данных

В первой строке записано число  $T$  ( $1 \leq T \leq 5$ ) — количество тестовых случаев. Далее следует их описание.

В первой строке записаны два числа  $n$  и  $m$  ( $1 \leq n, m \leq 100\,000$ ) — количество функций и запросов.

В каждой из следующих  $n$  строк записаны целые числа  $a_i$  и  $b_i$  ( $1 \leq a_i \leq 50\,000$ ,  $1 \leq b_i \leq 10^{18}$ ) — коэффициенты функции  $f_i(x)$ .

В каждой из следующих  $m$  строк следует описание запроса в формате, описанном выше.

### Формат выходных данных

Для каждого запроса третьего типа выведите целое число — минимальное значение функций  $f_i(x)$ . Если в данный момент в множестве нет неудаленных функций, выведите «-1».

### Пример

стандартный ввод	стандартный вывод
1	10
2 8	116
3 9	82
6 100	-1
3 4	
2 1	
3 4	
1 1 1	
3 4	
2 2	
2 3	
3 4	



## Задача О. Сбежать через лист

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.3 секунд  
Ограничение по памяти: 256 мегабайт

Дано дерево на  $n$  вершинах (пронумерованных от 1 до  $n$ ) с корнем в вершине 1. В вершине  $i$  записаны два числа:  $a_i$  и  $b_i$ .

Вы можете прыгнуть из вершины в любую вершину в её поддереве. Стоимость такого прыжка из вершины  $x$  в вершину  $y$  равна произведению  $a_x$  и  $b_y$ . Суммарная стоимость пути между вершинами, состоящего из нескольких прыжков равна сумме стоимостей прыжков в нём. Для каждой вершины посчитайте минимальную стоимость пути от неё до какого-либо листа. Обратите внимание, что корень дерева не является листом, даже если имеет степень 1.

Учтите, что нельзя совершать прыжок из вершины в ту же вершину.

### Формат входных данных

В первой строке содержится целое число  $n$  ( $2 \leq n \leq 10^5$ ) — количество вершин в дереве.

Во второй строке через пробел заданы  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^5 \leq a_i \leq 10^5$ ).

Во третьей строке через пробел заданы  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $-10^5 \leq b_i \leq 10^5$ ).

В следующих  $n - 1$  строках содержатся пары целых чисел  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), разделённых пробелом, обозначающие ребро между вершинами  $u_i$  и  $v_i$  в дереве.

### Формат выходных данных

Выведите  $n$  целых чисел через пробел,  $i$ -е из которых обозначает минимальную стоимость, чтобы добраться от вершины с номером  $i$  до какого-либо листа.

### Примеры

стандартный ввод	стандартный вывод
3 2 10 -1 7 -7 5 2 3 2 1	10 50 0
4 5 -10 5 7 -8 -80 -3 -10 2 1 2 4 1 3	-300 100 0 0

### Замечание

В первом тестовом примере вершина 3 сама является листом, поэтому ответ равен 0. Для вершины 2 прыжок в вершину 3 стоит  $a_2 \times b_3 = 50$ . Для вершины 1 прыжок в вершину 3 стоит  $a_1 \times b_3 = 10$ .

Во втором тестовом примере вершины 3 и 4 являются листьями, поэтому ответ для них равен 0. Для вершины 2 прыжок в вершину 4 стоит  $a_2 \times b_4 = 100$ . Для вершины 1 необходимо сначала прыгнуть в вершину 2 прыжком стоимостью  $a_1 \times b_2 = -400$ , а затем прыгнуть из 2 в 4 за  $a_2 \times b_4 = 100$ .