

Содержание

Задача 4А. 17 стульев [2 сек, 256 mb]	2
Задача 4В. Тримино [2 сек, 256 mb]	3
Задача 4С. Симпатичные узоры наносят ответный удар [3 сек, 256 mb]	4
Задача 4D. Калила и Димна на лесозаготовках [2 сек, 256 mb]	5
Задача 4Е. Коды, сохраняющие порядок [2 сек, 256 mb]	6
Задача 4F. Утилитаризм [5 сек, 256 mb]	7
Задача 4G. Путешествие [1 сек, 256 mb]	8
Задача 4H. Война сладкоежек [2 сек, 256 mb]	9

Задача 4А. 17 стульев [2 сек, 256 mb]

Остап Бендер снова пытается получить причитающиеся драгоценности, но на этот раз они были заперты в шкатулке, для открытия которой необходимо иметь N ключей. По закономерной случайности каждый из ключей был спрятан в одном из N стульев, распроданных на недавнем аукционе. После аукциона эти стулья были развезены в N городов.

И вот теперь Остап решился на новую безумную затею: заехать в каждый из городов и, провернув в каждом из них аферу, выкрасть необходимые ключи. Чтобы избежать конфликтов с недоброжелателями, Остап не хочет больше одного раза появляться в каком-либо городе. Также у Остапа есть список цен за проезд между каждой парой городов. Изначально Остап находится в городе под номером 1 и после посещения всех городов может незаметно скрыться из этой страны.

Помогите Остапу найти порядок посещения городов, при котором ему потребуется потратить как можно меньше средств на странствия, и тогда, возможно, он поделится с Вами добытыми бриллиантами.

Формат входных данных

Первая строка содержит единственное число N — количество городов ($1 \leq N \leq 17$).

Следующие N строк содержат по N целых неотрицательных чисел. j -тое число в i -той строке означает стоимость проезда из города i в город j ($0 \leq a_{ij} \leq 100$). Если $a_{ij} > 0$, то проезд стоит a_{ij} рублей, иначе — это означает, что из города i в j невозможно проехать напрямую.

Формат выходных данных

В первой строке выведите минимальную сумму денег, необходимую для посещения всех городов Остапом. В следующей строке выведите N чисел — порядок посещения городов, при котором эта сумма достигается. Если затею Остапа невозможно вывести, то в единственной строке выходного файла выведите число -1.

Примеры

trader.in или stdin	trader.out или stdout
3 0 3 2 3 0 6 2 6 0	8 1 3 2
5 0 6 4 0 0 6 0 7 0 7 4 7 0 0 0 0 0 0 0 2 0 7 0 2 0	20 1 3 2 5 4

Задача 4В. Тримино [2 сек, 256 mb]

Недавно Глеб рассыпал свои домашние игрушки. Чтобы остаться незамеченным, он решил собрать их обратно в коробку. Коробка имеет размеры n на m . Все игрушки Глеба можно представить в виде уголка из трех клеток, либо в виде трех клеток, расположенных вдоль прямой. Вам требуется выяснить, можно ли собрать все игрушки обратно в коробку так, чтобы вся коробка была покрыта игрушками и никакие две игрушки не пересекались. Дома все игрушки можно переворачивать.

Формат входных данных

В единственной строке записаны 4 целых числа n, m, a, b ($1 \leq n \cdot m \leq 100; 0 \leq a \leq 100; 0 \leq b \leq 100$) — размеры коробки и количество игрушек типа «прямая» и «уголок» соответственно.

Формат выходных данных

Если замостить коробку данным количеством игрушек возможно, в единственной строке выведите «YES» (без кавычек).

Иначе в единственной строке выведите «NO» (без кавычек).

Примеры

trimino.in или stdin	trimino.out или stdout
4 4 1 4	NO
3 4 0 4	YES
6 6 8 4	YES

Задача 4С. Симпатичные узоры наносят ответный удар [3 сек, 256 mb]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся два натуральных числа n и m . $1 \leq n \leq 10^{100}$, $1 \leq m \leq 5$, $1 \leq p \leq 10000$.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$ по модулю p . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Примеры

nice3.in или stdin	nice3.out или stdout
2 2 20	14
3 3 7	0

Задача 4D. Калила и Димна на лесозаготовках [2 сек, 256 mb]

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

lumber.in или stdin	lumber.out или stdout
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 4Е. Коды, сохраняющие порядок [2 сек, 256 mb]

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

codes.in или stdin	codes.out или stdout
5	00
1 8 2 3 1	01
	10
	110
	111

Задача 4F. Утилитаризм [5 sec, 256 mb]

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

korea.in или stdin	korea.out или stdout
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 4G. Путешествие [1 сек, 256 mb]

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

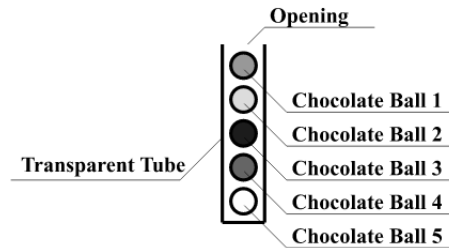
Выведите искомое наибольшее количество денег, которые Петя может потратить чтобы добраться из города 1 в город n . Гарантируется, что ответ не превосходит 10^{12} .

Примеры

travel.in или stdin	travel.out или stdout
4 5 10 2 0 1 10 15 3 14 17 2 3	123
1 709 50 8	0

Задача 4Н. Война сладкоежек [2 сек, 256 mb]

Есть полукоткрытая трубка в которой лежит n шоколадных шариков пронумерованных от 1 до n начиная с открытого конца. С каждым шариком связано два целых неотрицательных числа — *питательность* и *вкусность*.



Алиса и Брианна, принцессы двух стран, играют в игру. И у Алисы, и у Брианны есть текущий уровень энергии — A и B соответственно. Алиса и Брианна ходят по очереди начиная с Алисы. В свой каждая из принцесс делает одно из двух:

- Или пропускает ход, в таком случае её уровень энергии уменьшается на 1. Нельзя пропустить ход если уровень энергии уже на 0.
- Или съедает самый верхний шарик — в таком случае её уровень энергии увеличивается на r_i — питательность этого шарика, а удовольствие повышается на s_i — вкусность этого шарика.

Игра заканчивается когда трубка опустеет.

И Алиса и Брианна стремятся получить как можно больше удовольствия. Найдите сколько удовольствия получит каждая из них, если они обе играют оптимальным образом.

Формат входных данных

Первая строка содержит три целых числа n , A и B ($1 \leq n \leq 150$, $0 \leq A, b \leq 10^9$).

Каждая из следующих n строк содержит два целых числа — r_i и s_i ($0 \leq s_i$, $\sum s_i \leq 150$, $r_i \leq 10^9$) — питательность и вкусность соответствующего шарика.

Формат выходных данных

Выведите два числа — удовольствие которое получит Алиса и Брианна, если они обе играют оптимальным образом.

Примеры

sweet.in или stdin	sweet.out или stdout
2 5 4 5 7 4 8	8 7
3 50 1 49 1 0 10 0 1	10 2
4 3 2 1 5 2 46 92 40 1 31	77 45
5 2 5 56 2 22 73 2 2 1 55 14 18	57 93