

## Problem A. Злые свинки или Anti Angry Birds

Input file: pigs.in  
Output file: pigs.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Вы никогда не задумывались, почему в Angry Birds у птиц нет крыльев? Тем же вопросом задались разработчики новой игры. В их версии смысл игры прямо противоположен Angry Birds: злая свинка стреляет по неподвижно висящим в воздухе птицам из лазерного ружья (завязка явно не хуже исходной игры).

Птицы в игре представляются точками на плоскости. Выстрел сбивает всех птиц, находящихся на одной прямой. При этом сбита птица, падая, сбивает всех птиц, находящихся ровно под ней. По заданному расположению птиц необходимо определить, какое минимальное количество выстрелов необходимо, чтобы все птицы были сбиты.

### Input

Первая строка входного файла содержит единственное целое число  $N$   $1 \leq N \leq 100000$  — количество птиц.

Следующие  $N$  строк содержат по два натуральных числа каждая  $x_i, y_i$  — координаты  $i$ -ой птицы ( $0 < x, y \leq 10^9$ ). Свинка находится в точке с координатами  $(0, 0)$ .

### Output

Единственная строка выходного файла должна содержать одно целое число — минимальное количество выстрелов, необходимое для того, чтобы сбить всех птиц.

### Examples

pigs.in	pigs.out
6 1 1 2 2 3 3 2 1 3 2 3 1	1
6 1 1 2 2 3 3 2 1 3 2 3 4	2

## Problem B. Народный IPO

Input file: ipo.in  
Output file: ipo.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Компания «Рога и копыта» решила провести народный IPO (первоначальное публичное предложение акций компании на продажу широкому кругу лиц). Как обычно у Остапа Бендера не обошлось без аферы.

Потенциальные покупатели прислали заявки на покупку. Каждая заявка содержит два числа: процент акций компании, который хотел бы купить этот человек, а также цену, которую он готов заплатить за 1 процент акций.

Суть аферы заключается в следующем: Остап Бендер решил удовлетворить все заявки, но продавать каждому покупателю не процент от общего количества акций компании, а процент от еще не распроданной части. Остап Бендер чит уголовный кодекс и деньги с покупателя может брать только за определенное количество проданных акций (а не за абстрактные доли от остатка и другие изобретенные им вещи).

Тем не менее, изменяя порядок удовлетворения заявок покупателей можно влиять на количество вырученных денег. Помогите Остапу Бендеру наиболее выгодно продать «Рога и копыта». Акции у компании так много, что Остап может продать любой процент акций, в том числе дробный.

### Input

В первой строке задано количество людей, подавших заявки  $N$  ( $1 \leq N \leq 1000$ ). Далее следуют  $N$  строк по два числа в каждой: натуральное число, не превосходящее 100 — часть компании в процентах, которую хочет купить очередной человек и натуральное число не превосходящее  $10^9$  — количество денег, которое он заплатит за 1 процент от общего количества акций компании.

### Output

Выведите  $N$  чисел — номера людей в порядке, в котором должны удовлетворяться их заявки. Номера людей соответствуют порядку их описания во входных данных. Нумерация начинается с 1.

### Examples

ipo.in	ipo.out
2 25 5 25 10	2 1
3 10 30 20 15 30 10	1 2 3

### Note

В первом тесте выгоднее сначала удовлетворить заявку второго человека, в результате чего будет продано 25% акций компании. Первому человеку будет продано 25% от оставшей-

ся части (т.е. 18,75% от общего числа акций). Суммарная выручка в таком случае составит  $25 \times 10 + 18,75 \times 5 = 343,75$  рублей.

### Problem C. Домино

Input file: domino.in  
Output file: domino.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Проводя генеральную уборку на дачном чердаке, Саша нашел в комодке кучу доминошек из разных наборов. Каждая доминошка представляет собой прямоугольник, разделенный на две половинки. На каждой из половинок нарисовано от 0 до 6 точек. Ориентации доминошки не имеют — их можно как угодно поворачивать.

В совсем раннем детстве Саша видел, как играют в домино: суть игры заключается в том, что надо брать доминошку и как можно громче колотить ее об стол, крича при этом «рыба!». Услышав доносящийся с чердака грохот, наверх поднялся Сашин дедушка. Он смог объяснить Саше настоящие правила игры в домино: игроки составляют длинную цепочку, в которой соседние доминошки касаются половинками с одинаковым числом точек.

Саше решил называть «дружными доминошками» пару доминошек, которые можно поставить в игре рядом (т.е. доминошки в паре соприкасаются половинками с равными числами) в том или ином порядке. Играть в домино ему не с кем, поэтому Саша развлекается тем, что всевозможными способами составляет пары и считает количество «дружных доминошек».

По заданному набору доминошек определите, сколько пар «дружных доминошек» можно составить из него. Пары, отличающиеся хотя бы одной доминошкой, считаются различными. По-разному составленная пара из одних и тех же доминошек считается один раз.

#### Input

В первой строке входного файла содержится натуральное число  $N$  — количество доминошек ( $1 \leq N \leq 100\,000$ ).

В каждой из последующих строк содержится описание доминошки: два целых числа  $X$  и  $Y$  ( $0 \leq X, Y \leq 6$ ) — количество точек на каждой из половинок доминошки.

#### Output

Выведите одно число — количество пар «дружных доминошек».

### Examples

domino.in	domino.out
2 1 2 2 1	1
5 1 2 2 3 3 1 4 3 4 3	8

### Note

Во втором тесте дружными являются следующие пары:

1-2 2-3

1-2 3-1

2-3 3-1

2-3 4-3

2-3 4-3

3-1 4-3

3-1 4-3

4-3 4-3

В этой задаче для хранения ответа необходим 64-битный тип данных (int64 на языке Pascal и long long на языках C/C++).

### Problem D. Фибоначчиева система счисления

Input file: fibonacci.in  
Output file: fibonacci.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Напомним, что числами Фибоначчи называется последовательность чисел, получаемая по следующему правилу:  $f_0 = f_1 = 1$ ,  $f_k = f_{k-1} + f_{k-2}$ , где  $k > 1$ .

Фибоначчиева система счисления (ФСС) — это позиционная система счисления с алфавитом, состоящим из двух цифр: 0 и 1, а ее базисом является последовательность чисел Фибоначчи 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... ( $f_0 = 1$  в базис не включается). В фибоначчией системе, как и во всех позиционных системах счисления, «вес» каждого разряда определяется соответствующим элементом базиса этой системы. Так,  $10011_{fib} = 1 \times 8 + 0 \times 5 + 0 \times 3 + 1 \times 2 + 1 \times 1 = 11$ . Если не наложить дополнительных ограничений, то представление чисел в такой системе счисления оказывается неоднозначным.

Например,  $11_{10} = 1111_{fib} = 10011_{fib} = 10100_{fib}$

Требуется написать программу, которая для натурального числа  $N$  будет выводить все его пред-

ставления в ФСС. Если полученные представления рассматривать как двоичные числа, то вывести их надо в порядке возрастания — от меньшего числа к большему.

### Input

Во входном файле записано единственное число  $N$  ( $1 \leq N \leq 2 \times 10^9$ ).

### Output

В выходной файл требуется вывести все представления числа  $N$  в ФСС по одному на каждой строке, упорядоченные по возрастанию значений соответствующих этим представлениям двоичных чисел.

### Examples

fibonacci.in	fibonacci.out
5	110 1000
11	1111 10011 10100

### Problem E. Палиндромика

Input file: palin.in  
Output file: palin.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Игра «Палиндромика» набирает все большую популярность в казино Рулеттенбурга. Правила «Палиндромики» довольно просты: в начале игры на листок записывается строка и игроки по очереди стирают первый или последний символ. Побеждает игрок, *перед* ходом которого строка представляет собой палиндром. Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево.

Алексей Иванович — азартный игрок, однако вместо участия в игре предпочитает делать ставки. Ему удалось узнать, какая строка будет предложена для игры. Алексею Ивановичу предсказать исход игры при оптимальных действиях обоих игроков не под силу. За помощью он обратился к вам.

### Input

В единственной строке входного файла содержится строка, предложенная игрокам. Строка состоит из маленьких латинских букв. Длина строки не превышает 50000 символов.

### Output

Выведите номер игрока, который победит в игре (число 1 или 2) при оптимальной игре каждого из игроков.

### Examples

palin.in	palin.out
3 uho	1
6 ababab	2

### Problem F. Сдвиг

Input file: shift.in  
Output file: shift.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Петя и Вася придумали систему шифровки для обмена записками. Суть ее заключается в следующем. Дана исходная строка  $S$ .  $S'$  — циклический сдвиг строки влево (первый символ становится последним, а остальные перемещаются на одну позицию влево),  $S''$  — циклический сдвиг строки  $S'$  и т.д. Петя с Васей выписывают на листок бесконечную последовательность символов  $SS'S''S''' \dots$ . Если им необходимо зашифровать символ  $C$ , то они ищут какое-либо вхождение этого символа в выписанную последовательность и записывают его порядковый номер  $k$ . Нумерацию символов они ведут с единицы.

Злоумышленник Коля перехватил сообщение и выкрал исходную строку  $S$ . Однако он не может определить, какой символ стоит в последовательности  $SS'S''S''' \dots$  на  $k$ -ом месте. Помогите злоумышленнику Коле узнать, какой символ соответствует числу  $k$ .

### Input

Первая строка входного файла содержит строку, состоящую только из строчных латинских букв. Длина строки не превышает 100000 символов. Вторая строка входного файла содержит единственное целое число  $1 \leq k \leq 2 \times 10^9$ .

### Output

Единственная строка выходного файла должна содержать символ, который окажется на  $k$ -ом месте сформированной строки.

### Examples

shift.in	shift.out
abcd 5	b
abcd 17	a

## Problem G. Туризм

Input file: `tourism.in`  
Output file: `tourism.out`  
Time limit: 2 seconds  
Memory limit: 64 megabytes

Александр недавно увлекся горным туризмом. Ему уже надоело покорять отдельные горные пики, и он собирается покорить самую настоящую горную цепь!

Напомним, что Александр живет в плоском мире. Горная цепь состоит из отрезков, соединяющих точки на плоскости, каждая из которых находится строго правее предыдущей ( $x$ -координата следующей точки больше, чем у предыдущей). Трассой на горной цепи называется её часть между двумя фиксированными точками цепи, которые не обязательно являются концами отрезков.

Участок, на котором при движении по трассе координата  $y$  (высота) всегда возрастает, называется подъемом, величиной подъема называется разность высот между начальной и конечной точками участка. Ниже изображена горная цепь из 7 точек, на ней задана трасса, которая состоит из четырех участков (трасса выделена жирным). Если проходить трассу слева направо, то один из участков является подъемом.



Туристическая компания предлагает на выбор несколько трасс на одной горной цепи. Александр из-за финансовых трудностей может выбрать для поездки только одну из этих трасс. Вы решили помочь ему с выбором. Александру важно для каждой трассы определить суммарную высоту подъемов на ней.

### Input

В первой строке входного файла содержится единственное число  $N$  — количество точек ломаной, задающей горную цепь ( $1 \leq N \leq 100\,000$ ). Далее в  $N$  строках содержатся описания точек, каждое из которых состоит из двух целых чисел,  $x_i$  и  $y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ).

В следующей строке находится число  $M$  — количество трасс ( $1 \leq M \leq 100\,000$ ).

Далее в  $M$  строках содержатся описания трасс, каждое описание представляет собой два целых числа,  $s_i$  и  $f_i$ , они обозначают  $x$ -координаты начала и конца трассы, соответственно. Начало и конец трассы могут совпадать.

Гарантируется, что во входном файле задана именно горная цепь, и ни одна из трасс не выходит за ее пределы.

### Output

Для каждой трассы выведите одно число — суммарную высоту подъемов на данной трассе. Ваш ответ будет считаться правильным, его абсолютная или относительная погрешность не будет превышать  $10^{-6}$ .

### Examples

tourism.in	tourism.out
7 2 1 4 5 7 4 8 2 9 6 11 3 15 3 1 5 10	4.0000000000
6 1 1 3 2 5 6 7 2 10 4 11 1 3 9 11 3 10 7 3	0.6666666667 6.0000000000 4.0000000000

## Problem H. Башня

Input file: `tower.in`  
Output file: `tower.out`  
Time limit: 1 second  
Memory limit: 64 megabytes

Петя в очередной раз купил себе набор из кубиков. На этот раз он выстроил из них настоящую крепость — последовательность из  $N$  столбиков, высота каждого столбика составляет  $A_i$  кубиков.

Вскоре ему стало интересно, насколько его крепость защищена от жуликов и воров. Для этого он ввел понятия башни. Башней называется любая последовательность из  $K$  столбиков подряд (где  $K$  — любимое число Пети). Защищенность башни определяется как суммарная высота всех столбиков этой башни (чем она больше, тем громаднее и ужаснее она кажется), умноженная на минимум высоты столбиков башни (т.к. враги, очевидно, будут пытаться проникнуть через самое слабое место башни). Непроступность крепости определяется как сумма защищенностей каждой из башен.

Петя решил как можно скорее посчитать, какова же непроступность его крепости. Однако вскорее он понял, что недостаточно знать высоту каждого из столбиков. В зависимости от того, как

сгруппировать столбики в башни, получится разный результат. В различных вариантах группировки часть столбиков могут не принадлежать ни одной из башен. Разумеется, Петя выберет то разбиение на башни, при котором неприступность будет максимальной.

Петя успешно справился со своей задачей, но теперь Правительство Флатландии решило защитить свой горный курорт. Правительство уже построило крепость из кубиков (просто кубики были побольше). Теперь вы должны помочь Правительству посчитать неприступность этой крепости. Единственная трудность состоит в том, что у Правительства было очень много денег, и поэтому крепость была построена очень длинная.

### Input

В первой строке входного файла содержится число  $N$  — количество столбиков в крепости и число  $K$  — любимое число Пети ( $1 \leq K \leq N \leq 100\,000$ ). Далее на следующей строке содержатся  $N$  целых чисел, обозначающих  $A_i$  ( $1 \leq A_i \leq 10^6$ ).

### Output

На первой строке выведите число  $Q$  — количество башен в оптимальном разбиении. Далее выведите  $Q$  чисел — номера первых столбиков каждой башни.

### Examples

tower.in	tower.out
1 1 1	1 1
2 1 1 1000000	2 1 2
8 3 1 2 3 4 1 6 7 8	2 2 6

## Problem I. Лучшие друзья девушек - это фуллерены

Input file: fuller.in  
Output file: fuller.out  
Time limit: 1 second  
Memory limit: 64 megabytes

Развитие химической науки привело к тому, что высшие фуллерены (сложные молекулы углерода в виде шарика или продолговатой трубки) стали недорогими в производстве. Благодаря своим уникальным оптическим свойствам они нашли свое место и в ювелирной промышленности. Ювелирный дом «Кёрл, Крото и Смолли» выпустил уникальную коллекцию украшений из фуллеренов. Украшение продается в виде набора трубок-фуллеренов различной длины, из которых можно составить украшение самостоятельно.

Норма Джин очень любит сложные углеродные соединения и купила себе набор фуллеренов для составления украшений. Ее фирменный стиль состоит в том, чтобы носить украшения, составленные ровно из трех трубок фуллерена, причем в результате должен получаться тупоугольный треугольник. Норма Джин — объект постоянной охоты папарацци, поэтому не может позволить себе дважды появиться на людях с одним и тем же украшением.

Помогите Норме Джин узнать, сколько вечеров она сможет посетить с имеющимся у нее набором фуллереновых трубок. Фуллереновые трубки одинаковой длины считаются различными. Треугольники считаются различными, если они отличаются хотя бы одной трубкой. Треугольники, состоящие из одних и тех же трубок, считаются одинаковыми независимо от порядка трубок.

### Input

Первая строка входного файла содержит одно число  $N$  ( $1 \leq N \leq 5000$ ) — количество фуллереновых трубок в наборе Нормы Джин.

Вторая строка содержит  $N$  упорядоченных по возрастанию целых чисел  $L_i$  ( $1 \leq L_i \leq 2 \times 10^9$ ).

### Output

Выведите одно целое число — количество вечеров, на которые сможет сходить Норма Джин.

### Examples

fuller.in	fuller.out
4 2 2 3 4	3

## Problem J. Магнитные подушки

Input file: city.in  
Output file: city.out  
Time limit: 2 seconds  
Memory limit: 64 megabytes

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добираться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

### Input

В первой строке входного файла находятся числа  $N$  и  $M$  — количество небоскребов в городе и количество работающих магнитных подушек соответственно ( $3 \leq N \leq 100\,000$ ,  $1 \leq M \leq 100\,000$ ). В

каждой из следующих  $M$  строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до  $N$ . Гарантируется, что имеющиеся воздушные подушки позволяют перемещаться от одного небоскреба до любого другого.

### Output

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

### Examples

city.in	city.out
3 1 1 2 3	1 1
3 2 1 2 3 3 2 1	0
5 4 1 2 3 2 4 3 1 2 4 3 5 1	1 4