

Задача А. Двоичные числа

Имя входного файла: `binary.in`
Имя выходного файла: `binary.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Пусть задано положительное целое число x . Рассмотрим строку S , которая является записью его представления в двоичной системе счисления. Необходимо вычислить сумму всех различных чисел, двоичные представления которых являются подстроками строки S .

Например, пусть $x = 5$. Тогда $S = 101$, множество подстрок S таково: $\{1, 0, 1, 10, 01, 101\}$. Если интерпретировать эти строки как двоичные числа и удалить повторяющиеся, получается множество чисел: $\{0, 1, 2, 5\}$. Их сумма равна 8.

Формат входного файла

Входной файл содержит целое число x ($1 \leq x \leq 10^{12}$). Оно задано в десятичной системе счисления.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

<code>binary.in</code>	<code>binary.out</code>
5	8

Задача В. Галактическая конференция

Имя входного файла: `conferen.in`
Имя выходного файла: `conferen.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вот уже которую тысячу лет идет межгалактическая война между Республикой и Федерацией. Долгое время противники не соглашались ни на какие уступки, но теперь готовы сесть за переговоры.

Полным ходом идут приготовления. На одной из отдаленных планет нейтральной галактики уже построен зал, в котором будут проходить переговоры. План зала — это клетчатый прямоугольник шириной W и длиной L . Планируется, что справа будут находиться представители Федерации, а слева — делегаты Республики.

Столы, за которыми сидят дипломаты, имеют форму прямоугольника. Представители Федерации обычно замкнуты в себе, поэтому за одним столом их собирается немного, а республиканцы, наоборот, чрезвычайно общительны, и каждой группе необходим большой стол. На планете наличествуют столы двух размеров — 1×2 и 2×2 . Поэтому столы первого вида определили для Федерации, а второго — для Республики.

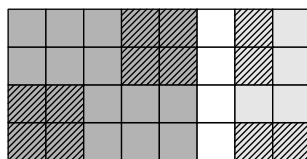
В зале столы могут располагаться только согласно сетке плана. То есть, каждая половинка стола 1×2 должна занимать ровно одну клетку на плане зала. Аналогичное правило действует и для столов другого вида.

Согласно тысячелетним традициям, столы для переговоров не должны соприкасаться, пусть даже и углами.

При расстановке столов возникают так называемые *зоны влияния* сторон. Каждая зона влияния представляет собой непрерывную (т.е. без промежутков) совокупность столбцов. Зона влияния Федерации распространяется начиная с самой правой клетки зала и заканчивая самой левой клеткой зала из тех, на которых стоят столы, принадлежащие Федерации. Аналогично (с точностью до перемены сторон местами) определяется зона влияния Республики.

В целях безопасности, расстановка столов должна удовлетворять следующему правилу: зоны влияния противников должны быть разделены не менее чем одним свободным столбцом. Если кто-то из противников отсутствует, то столы присутствующего противника можно ставить по всей площади зала, не нарушая при этом другие правила расстановки.

На рисунке заштрихованы клетки, занятые столами, светло-серым цветом показана зона влияния Федерации, темно-серым — зона влияния Республики.



Так как переговоры еще не начались, и неясно, какая из сторон будет сильнее к моменту начала переговоров, то вероятно любое соотношение числа столов Федерации и Республики. Даже возможно, что какое-либо из этих чисел, или даже оба, будут равны 0, так как одна из сторон может выиграть войну, бойкотировать или даже вовсе отменить переговоры.

На данный момент организаторов интересует количество способов расставить столы так, чтобы они удовлетворяли данным требованиям. Помогите им.

Формат входного файла

В первой строке входного файла содержатся числа W, L, P , $1 \leq W \leq 5$, $1 \leq L \leq 1000$, $2 \leq P \leq 10^4$.

Формат выходного файла

Выведите ответ на задачу по модулю P .

Примеры

conferen.in	conferen.out
2 4 1000	22

Задача С. Киоски с мороженым

Имя входного файла: `icecream.in`
Имя выходного файла: `icecream.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В маленьком провинциальном городке есть маленькая школа, в которой учатся не совсем большие дети. После занятий они бегут на автобусную остановку, откуда автобус развозит их по домам.

По дороге от школы до остановки есть N перекрестков, соединенных улицами. Школьники с улицы на улицу переходят только на перекрестках.

Все школьники, как известно, любят мороженое. Известная компания *Cold-N-Icy*, производящая мороженое, решила воспользоваться этим. Она хочет разместить киоски с мороженым на некоторых перекрестках таким образом, чтобы любой путь школьника от школы до остановки проходил хотя бы через один перекресток, на котором установлен киоск.

Так как установка и содержание киоска — дорогое дело, то компания решила привлечь Вас для того, чтобы определить минимальное число киосков, которое необходимо установить.

Помогите компании *Cold-N-Icy* найти это минимальное число.

Формат входного файла

В первой строке входного файла находится число перекрестков N ($1 \leq N \leq 100$).

В каждой из последующих N строк находится информация о перекрестках, соединенных улицами между собой. Перекрестки нумеруются, начиная с единицы. В начале i -той строки находится число K_i — количество мест (перекрестков, школы или остановки), соединенных улицами с i -тым перекрестком. Далее идет K_i мест, разделенных пробелами. Для обозначения перекрестков используются их номера, школа обозначается как `school`, остановка обозначается как `station`.

Если перекресток i находится в списке перекрестка j , то обратное также верно.

Гарантируется, что от школы до остановки всегда существует путь.

Формат выходного файла

Выведите одно число — минимальное число киосков, которые планируется установить.

Пример

<code>icecream.in</code>	<code>icecream.out</code>
2 2 school station 2 station school	2
3 3 school 2 3 3 school 1 3 3 station 1 2	1

Задача D. Прямые в пространстве

Имя входного файла: `lines3d.in`
Имя выходного файла: `lines3d.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В трехмерном пространстве заданы две прямые: l_1 и l_2 . Найдите расстояние между ними.

Формат входного файла

Каждая прямая задана двумя различными точками, лежащими на ней: на l_1 лежат точки A и B , на l_2 — точки C и D .

В первой строке входного файла заданы координаты точки A : x_A, y_A, z_A . Во второй, третьей и четвертой строках заданы координаты точек B, C и D соответственно.

Все координаты — целые числа, не превосходящие 10^3 по абсолютному значению.

Формат выходного файла

Выведите в выходной файл расстояние между заданными прямыми. Ваш ответ будет принят, если относительная или абсолютная погрешность не будет превышать 10^{-5} .

Пример

<code>lines3d.in</code>	<code>lines3d.out</code>
0 0 0 1 1 1 0 0 0 2 3 9	0.0
0 0 0 1 1 1 0 0 1 1 1 2	0.8164965809277260327324280249
0 0 0 1 1 0 0 0 1 1 0 1	1.0

Задача Е. Замок на холодильнике

Имя входного файла: lock.in
Имя выходного файла: lock.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вася очень любит покушать. Но его мама считает, что столько кушать нельзя. Поэтому она поставила на холодильник замок, представляющий собой прямоугольную сенсорную панель из n строк и m столбцов, занумерованных по порядку числами от 1 до n и от 1 до m соответственно. Изначально все клетки замка покрашены в некоторые цвета. Известно, что панель может отображать не более, чем $\min(n, m) - 1$ цветов, занумерованных натуральными числами от 1 до $\min(n, m) - 1$. Панель можно перекрашивать: если выбрать в одной строке (в одном столбце) две клетки одного цвета, то вся эта строка (весь этот столбец) перекрасится в цвет выбранных клеток. Чтобы открыть холодильник, необходимо не более, чем за $2 \cdot (n + m)$ операций, перекрасить всю панель в какой-нибудь один цвет.

Вам требуется помочь Васе взломать мамин замок. Напишите программу, которая по заданной раскраске панели будет выводить последовательность длиной не более $2 \cdot (n + m)$ операций перекрашивания, приводящую к открытию замка.

Формат входного файла

В первой строке находятся числа n, m ($3 \leq n, m \leq 100$). Затем задается начальное состояние панели замка: в n строках по m натуральных чисел из диапазона 1 до $\min(n, m) - 1$.

Формат выходного файла

В первой строке выведите целое число A — количество операций перекрашивания. В следующих A строках опишите предлагаемые операции перекрашивания. Строка описания содержит в начале символ «V», если соответствующая операция перекрашивает столбец, и «H», если строку. Затем в строке через пробел идут четыре натуральных числа x_1, y_1, x_2, y_2 — координаты двух выбранных клеток. Цвета в этих клетках на начало операции перекрашивания должны быть одинаковыми.

Примеры

lock.in	lock.out
3 3 1 1 2 2 2 1 2 2 2	3 V 2 1 3 1 V 2 2 3 2 V 1 3 3 3
4 5 1 2 3 2 1 3 2 1 3 2 1 1 1 1 1 2 2 2 2 2	6 H 1 1 1 5 V 1 1 3 1 V 1 2 3 2 V 1 3 3 3 V 1 4 3 4 V 1 5 3 5

Задача F. Числа-подстроки

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

За один шаг к числу X разрешается прибавить или вычесть любое положительное число Y , десятичная запись которого является подстрокой десятичной записи числа X . Стоимость такой операции равна сумме цифр числа Y .

Необходимо за минимальную стоимость получить из числа a число b , при этом все промежуточные числа должны быть положительными и не должны превышать n .

Формат входного файла

Входной файл содержит три целых числа: n, a, b ($1 \leq a, b \leq n \leq 5000$).

Формат выходного файла

Если из числа a нельзя получить число b , выведите в выходной файл одно число — -1 .

Если такая последовательность преобразований существует, в первой строке выходного файла выведите минимальную стоимость требуемого преобразования. Во второй строке выходного файла выведите число k — количество шагов в преобразовании. В последующих k строках выведите сами шаги преобразования — по одному в строке. Каждая строка должна иметь вид $+число$ или $-число$, в зависимости от того, прибавляется или вычитается очередное число.

Примеры

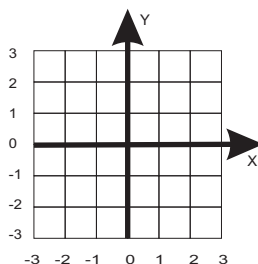
<code>numbers.in</code>	<code>numbers.out</code>
20 12 18	5 3 -2 +10 -2
100 5 43	29 8 +5 +1 +1 +1 +13 +26 -5 -4
50 5 43	-1

Задача G. Городской парк

Имя входного файла: park.in
Имя выходного файла: park.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В Санкт-Петербурге было решено открыть новый городской парк. Власти города нарисовали план парка на карте и теперь хотят приступить к закупке растительности и постройке забора. Закупкой и рассадкой зелени пусть занимаются ботаники, а забор предстоит строить Вам.

Санкт-Петербург, как почти любой современный город представляет собой правильную прямоугольную сетку единичных квадратов (кварталов). Введем систему с центром в одном из узлов квадратной сетки, осью OX направленной с запада на восток и осью OY направленной с юга на север. Парк будет представлять собой связную область состоящую из единичных квадратов.



Вам необходимо сделать заказ на резные решетки, которые производит сталепромышленный завод «Решет & Ко». Для этого, Вам необходимо узнать периметр парка.

Формат входного файла

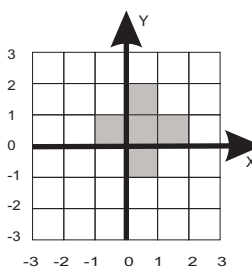
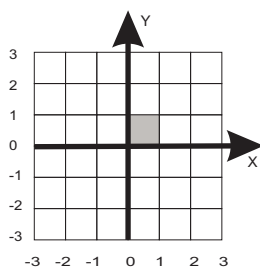
В первой строке входного файла указано число n ($1 \leq n \leq 100000$) — количество единичных квадратов, отведенных под парк. В следующих N строках описываются эти кварталы. Каждый квартал задается координатами своего нижнего левого угла. Гарантируется, что парк представляет собой связную фигуру. Координаты не превышают 10^9 по абсолютной величине.

Формат выходного файла

В единственную строку входного файла запишите одно число — периметр парка.

Пример

park.in	park.out
1 0 0	4
5 0 0 0 -1 -1 0 1 0 0 1	12



Задача Н. Шаблон-палиндром

Имя входного файла: `pattern.in`
Имя выходного файла: `pattern.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Шаблон размера n назовем строку длины n , каждый из символов которой входит в множество $\{0, 1, ?\}$. Шаблоны преобразуются в строки из нулей и единиц по следующим правилам:

- символы 0 и 1 могут быть преобразованы только сами в себя;
- символ ? может быть преобразован либо в 0, либо в 1;

Палиндромом называется строка, одинаково читающаяся с обеих сторон. Например, строка `abba` является палиндромом, а строка `abc` — нет.

Необходимо найти наименьшую лексикографически являющуюся палиндромом строку, в которую может быть преобразован заданный шаблон p .

Формат входного файла

Первая строка входного файла содержит шаблон p . Его размер положителен и не превосходит 1000.

Формат выходного файла

В выходной файл выведите искомый палиндром или `NO SOLUTION`, если ни одного палиндрома из заданного шаблона получено быть не может.

Примеры

<code>pattern.in</code>	<code>pattern.out</code>
<code>01????0</code>	<code>0100010</code>
<code>010?</code>	<code>NO SOLUTION</code>

Задача I. Восстановление строки

Имя входного файла: `restore.in`
Имя выходного файла: `restore.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Напомним, что строка $B = b_1b_2b_3\dots b_m$, является *подпоследовательностью* строки $A = a_1a_2a_3\dots a_n$, если существует строго возрастающая последовательность $\{i_1, i_2, i_3, \dots, i_m\}$ индексов A , такая, что для всех $j \in [1, m]$, выполняется $A_{i_j} = B_j$. Например, $B = "aba"$ является подпоследовательностью строки $A = "abacaba"$. Последовательность индексов в этом случае может быть такой: $\{1, 2, 3\}$.

Задано число n . Необходимо найти строку вида $a^i b^j c^k$, в которой abc встречается ровно n раз как подпоследовательность, причем длина найденной строки должна быть минимальна.

Строки вида $a^i b^j c^k$ — это строки, в которых вначале идут i символов a , затем j символов b , затем k символов c . Например, $a^2 b^3 c^4 = aabbbcccc$.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^{10}$).

Формат выходного файла

В выходной файл выведите три целых числа: i, j, k .

Примеры

	<code>restore.in</code>	<code>restore.out</code>
1	1	1 1 1
2	2	1 1 2

Задача J. Таблица

Имя входного файла: `table.in`
Имя выходного файла: `table.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Заданы две последовательности чисел: a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m . Рассмотрим таблицу $c_{i,j} = a_i + b_j$. Необходимо найти, какие числа и сколько раз встречаются в этой таблице.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n . Все a_i удовлетворяют условию $0 \leq a_i < 2^{15}$.

Третья строка входного файла содержит целое число m ($1 \leq m \leq 10^5$). Четвертая строка входного файла содержит m целых чисел b_1, b_2, \dots, b_m . Все b_i удовлетворяют условию $0 \leq b_i < 2^{15}$.

Формат выходного файла

В выходной файл выведите информацию о том, какие числа и сколько раз содержатся в описанной таблице. Для каждого числа выведите одну строку, имеющую формат **число - количество вхождений этого числа**. Числа должны быть упорядочены по возрастанию. Тире должно отделяться от чисел ровно одним пробелом.

Примеры

table.in	table.out
3	1 - 1
1 2 3	2 - 2
2	3 - 2
0 1	4 - 1
3	1 - 2
1 2 3	2 - 2
2	3 - 2
0 0	