

A. *Стоимость пирожков*

Пирожок в столовой стоит  $A$  рублей и  $B$  копеек. Определите, сколько рублей и копеек нужно заплатить за  $N$  пирожков.

Программа получает на вход три целых положительных числа:  $A, B, N$  и должна вывести два числа — ответ на вопрос задачи.

Input	Output
10 15 2	20 30

B. *Разность времён*

Даны значения двух моментов времени, принадлежащих одним и тем же суткам: часы, минуты и секунды для каждого из моментов времени. Известно, что второй момент времени наступил не раньше первого.

Программа на вход получает три целых числа: часы, минуты, секунды, задающие первый момент времени и три целых числа, задающих второй момент времени.

Программа должна вывести количество секунд прошло между двумя указанными моментами времени.

Input	Output
1 1 1 2 2 2	3661

C. *Автопробег*

За день машина проезжает  $N$  километров. Сколько дней нужно, чтобы проехать маршрут длиной  $M$  километров?

В первой строке дано число  $N$ , во второй — число  $M$ . Программа должна вывести одно число — ответ на вопрос задачи.

*Указание:* стоит вспомнить, как арифметически решить задачу целочисленного деления с округлением вверх (перед делением что-то надо было прибавить к делимому).

Input	Output
700 750	2

D. *Медленные часы*

В часах села батарейка, и они стали идти вдвое медленнее. Когда на часах было  $x_1$  часов  $y_1$  минут, правильное время было  $a_1$  часов  $b_1$  минут. Сколько времени будет на самом деле, когда часы в следующий раз покажут  $x_2$  часов  $y_2$  минут?

Программа получает на вход числа  $x_1, y_1, a_1, b_1, x_2, y_2$  в указанном порядке. Все числа целые. Числа  $x_1, a_1, x_2$  — от 0 до 23, числа  $y_1, b_1, y_2$  — от 0 до 59.

Выведите два числа  $a_2$  и  $b_2$ , определяющие сколько будет времени на самом деле, когда на часах будет  $x_2$  часов  $y_2$  минут.

Input	Output
12 34 10 34 12 35	10 36

Е. *Клетки шахматной доски*

Заданы две клетки шахматной доски. Если они покрашены в один цвет, то выведите слово YES, а если в разные цвета — то NO.

Input	Output
1	YES
1	
2	
2	

Ф. *Два прямоугольника*

На бесконечном клетчатом листке бумаги по линиям сетки вырезано два прямоугольника. Если на листе бумаги получилось две дырки, вывести NO, в противном случае вывести YES.

Вам даны 8 целых чисел —  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ , где  $(x_1, y_1)$  — координаты левого нижнего угла первого прямоугольника,  $(x_2, y_2)$  — координаты правого верхнего угла первого прямоугольника. Аналогично,  $(x_3, y_3)$  — координаты левого нижнего угла второго прямоугольника,  $(x_4, y_4)$  — координаты правого верхнего угла второго прямоугольника. Стороны обоих прямоугольников параллельны осям координат.

Гарантируется, что данные прямоугольники невырождены ( $x_1 < x_2, y_1 < y_2$  и аналогичные неравенства для второго набора координат), а также то, что ни одна сторона прямоугольника не лежит на прямой, содержащей другую сторону второго прямоугольника. Каждое число по модулю не превосходит  $10^4$ .

*Указание:* в этой задаче гораздо проще написать условие того, что прямоугольники образуют две дырки (т.е. имеют пустое пересечение), а не одну.

“Зафиксируем” первый прямоугольник и проверим, что второй *левее/правее/выше/ниже* первого — как это сделать? Что означают результаты этих проверок?

Input	Output
1	NO
1	
2	
2	
3	
3	
4	
4	

Г. *Лена и конфеты*

На столе стоят три вазы с конфетами. В левой вазе лежат  $A$  конфет, в средней вазе лежат  $B$  конфет, в правой вазе лежат  $C$  конфет. Лена съедает одну конфету из левой вазы, затем — одну конфету из средней вазы, затем из правой, средней, левой, средней, правой, средней и т. д. (слева направо, затем налево, опять направо и т.д.)

Если Лена хочет взять конфету из какой-то вазы, а конфет там нет, она расстраивается и идёт спать. Определите, сколько конфет съест Лена.

Программа получает на вход три целых неотрицательных числа  $A, B, C$  — количество конфет в левой, средней, правой вазе. Сумма трёх данных чисел не превосходит  $2 \times 10^9$ .

*Указание:* Прежде, чем писать в этой программе цикл, моделирующий поедание конфет, попробуйте на небольших значениях “поиграть” за Лену. Может быть удастся придумать более короткое решение.

Собственно, ограничение в условии ( $10^9$ ) ясно указывает на то, что решение с циклом будет работать слишком долго на некоторых тестах.

Input	Output
3	7
3	
3	

### Н. Шахматная доска

Аня разделила доску размера  $m \times n$  на клетки размера  $1 \times 1$  и раскрасила их в черный и белый цвет в шахматном порядке. Васю заинтересовал вопрос: клеток какого цвета получилось больше — черного или белого.

Для того, чтобы выяснить это, он спросил у Ани, в какой цвет она раскрасила  $j$ -ю клетку в  $i$ -м ряду доски (ряды нумеруются с единицы снизу вверх, клетки в ряду нумеруются с единицы слева направо). По этой информации Вася попытался определить, клеток какого цвета на доске больше.

Требуется написать программу, которая по размерам доски и цвету  $j$ -й клетки в  $i$ -м ряду определит, клеток какого цвета на доске больше — черного или белого.

Входной файл содержит пять целых чисел:  $m, n, i, j$  и  $c$  ( $1 \leq m, n \leq 10^9, 1 \leq i \leq m, 1 \leq j \leq n, c = 0$  или  $c = 1$ ). Значение  $c = 0$  означает, что  $j$ -я клетка в  $i$ -м ряду доски раскрашена в чёрный цвет, а значение  $c = 1$  — в белый цвет.

Выходной файл должен содержать одно из трёх слов:

- **black**, если чёрных клеток на доске больше
- **white**, если белых клеток на доске больше
- **equal**, если чёрных и белых клеток на доске поровну

Input	Output
3 5 1 1 0	black

### I. Сумма геометрической прогрессии

Забудьте формулу суммы геометрической прогрессии и вычислите сумму

$$1 + x + x^2 + \dots + x^n$$

Операцией возведения в степень пользоваться нельзя. В решении задачи надо обойтись одним циклом.

Программа получает на вход целое неотрицательное число  $n$  и вещественное число  $x$ .

*Указание:* Не надо вычислять каждое слагаемое  $x^k$  заново, ведь каждое следующее слагаемое отличается от предыдущего только домножением на  $x$ :  $x^{k+1} = x^k \cdot x$ . Так что достаточно помнить предыдущее слагаемое для того, чтобы быстро посчитать очередное.

Input	Output
4 0.1	1.1111

### J. Остатки

Даны целые неотрицательные числа  $a, b, c, d$ , при этом  $0 \leq c < d$ . Выведите в порядке возрастания все числа от  $a$  до  $b$ , которые дают остаток  $c$  при делении на  $d$ .

Если таких чисел в указанном интервале нет, то ничего выводить не нужно.

*Указание:* используйте цикл `for`. Главная сложность в этой задаче: подобрать правильное значение старта цикла. Конец цикла и шаг здесь очевидные. Надо понять, как отрезок  $[a, b]$  может быть устроен относительно последовательности нужных чисел.

Input	Output
2 5 0 2	2 4

К. Автобусы и такси

От школы-интерната Н. на очный тур Очень Открытой олимпиады прошло  $N$  школьников. Для доставки участников на место проведения директор интерната заказывает автобусы и такси. В каждый автобус можно посадить не более 50 школьников, в каждое такси — не более 4 школьников. Почасовая стоимость автобуса составляет  $A$  рублей, такси —  $B$  рублей (разумеется,  $A > B$ ). На олимпиаду все участники из интерната должны приехать одновременно, то есть в заказанном транспорте должно найтись место сразу для всех.

Помогите директору определить, какое количество автобусов и такси нужно заказать, чтобы потратить как можно меньшую сумму денег на дорогу.

Вводятся три целых числа, под одному в строке —  $N, A, B$  ( $1 \leq N \leq 100000, 1 \leq B < A \leq 1000$ ). Выведите два числа, разделённых пробелами — количество автобусов и количество такси для заказа в оптимальном случае. Если возможных ответов несколько, выведите любой.

*Указание:* можно, например, перебрать циклом `for` всевозможные значения количества автобусов (не забыть про 0 и про округление вверх, см. задачу про автопробег). А для каждого фиксированного количества автобусов несложно посчитать количество такси, которое необходимо заказать на оставшихся детей. Ну а пока перебираете, считайте, во что обойдётся каждый вариант (количество автобусов и количество такси) и запоминайте лучший.

Input	Output
4 3 2	0 1

Л. Количество чисел до первого нуля

Дана последовательность целых чисел по одному в строке. Сколько чисел было введено до первого нуля?

Input	Output
5 7 6 0	3

М. Сумма чисел до первого нуля

Дана последовательность чисел. Какова сумма чисел до первого нуля?

Input	Output
5 6 8 0	19

Н. Количество чисел между первыми двумя нулями

Дана последовательность целых чисел. Сколько чисел было введено между первым и вторым нулём? Гарантируется, что в последовательности есть как минимум два нуля.

Input	Output
5 0 6 0 8	1

О. Сумма чисел между первыми двумя нулями

Дана последовательность чисел. Какова сумма чисел между первым и вторым нулём?

Input	Output
0 6 8 0 1	14

P. *Второй максимум*

Последовательность состоит из различных целых чисел и завершается числом 0. Определите значение второго по величине элемента в этой последовательности.

Второй по величине элемент — максимальный элемент в последовательности, полученной из исходной удалением одного элемента, равного её максимуму.

Input	Output
1 7 9 0	7

Q. *Сумма чисел между предпоследней и последней двойками*

Последовательность состоит из натуральных чисел и завершается числом 0. Вычислить сумму всех элементов последовательности натуральных чисел между предпоследней и последней двойками (не включая сами двойки). Если двоек нет или она одна, вывести число  $-1$ .

Input	Output
3 2 1 2 3 2 1 0	3

R. *Сумма делителей числа*

Для данного натурального числа  $N$  ( $1 \leq N < 10^{10}$ ) требуется вычислить сумму его делителей, меньших самого числа.

Решение должно иметь сложность  $O(\sqrt{N})$ .

**ВНИМАНИЕ!**

Вы должны сдать решение, в котором определена функция `sum_divisors(x)`. Имя переменной-аргумента функции может отличаться от указанного.

Функция должна принимать на вход натуральное число и возвращать сумму его делителей, отличных от него самого. Она не должна ничего читать/печатать. В сданном решении не должно быть ничего, кроме описанной функции, в том числе чтения/печати.

Input	Output
12	16

S. *Дружественные числа*

*Дружественные числа* — это два различных натуральных числа, таких, что сумма всех делителей одного числа (меньших самого этого числа) равна другому числу, и наоборот (дружественными являются, например, 220 и 284).

Напишите функцию `is_friends(x, y)`, которая проверяет пару чисел на «дружественность» и возвращает логическое значение `True`, если пара чисел дружественная и `False` в противном случае. Функция `is_friends` должна использовать функцию `sum_divisors` из предыдущей задачи.

**ВНИМАНИЕ!**

Вы должны сдать решение, в котором определены две функции: `sum_divisors(x)` и `is_friends(x, y)`. Имена переменных-аргументов функций могут отличаться от указанных.

Первая функция должна принимать на вход натуральное число и возвращать сумму его делителей, отличных от него самого. Вторая функция принимает на вход два натуральных числа и возвращает `True/False`. Обе функции не должны ничего читать/печатать. В сданном решении не должно быть ничего, кроме описанных функций, в том числе чтения/печати.

Input	Output
220 284	True

T. *Дружественные числа в диапазоне*

*Дружественные числа* — это два натуральных числа, таких, что сумма всех делителей одного числа (меньших самого этого числа) равна другому числу, и наоборот (дружественными являются, например, 220 и 284).

Напишите программу, которая находит все пары не равных друг другу дружественных чисел в заданном диапазоне. Используйте функцию, которая вычисляет сумму делителей числа.

На вход программе подаётся две строки с натуральными числами  $a$  и  $b$  ( $a < b$ ).

Программа должна вывести пары различных дружественных чисел, каждое из которых находится на отрезке  $[a, b]$ .

В каждой паре сначала выводится меньшее число. Пары чисел должны выводиться в порядке возрастания первого числа из пары и разделяться запятой. Каждая пара заключена в скобки.

В случае, если таких пар в указанном диапазоне нет, вывести число 0.

Input	Output
1000 5000	(1184,1210) (2620,2924)

U. *Извлечь цифры*

Дана произвольная строка. Извлеките из этой строки все символы, являющиеся цифрами и составьте из них новую строку.

Программа получает на вход одну строку.

Требуется вывести новую строку, содержащую только цифры данной строки в том же порядке, в котором они в ней следовали.

*Указание:* не стоит понимать слово “извлечь” буквально — не надо *удалять* символы не-цифры из строки, лучше соберите новую из цифр.

Input	Output
2+2=4	224

V. *Капитан Флинт*

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: **North 5**, где первое слово — одно из **North, South, East, West**, а второе число — количество шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось  $OX$  направлена на восток, ось  $OY$  — на север.

Программа получает на вход последовательность строк указанного вида, завершающуюся строкой со словом **Treasure!**

Программа должна вывести два целых числа: координаты клада (сначала абсциссу, потом ординату).

Input	Output
North 5 East 3 South 1 Treasure!	3 4

W. *Буквы по одному разу*

Дана строка, в которой встречаются только символы A, C, G и T.

Вывести буквы этой строки по одному разу в том порядке, в котором они в этой строке появляются в первый раз.

Input	Output
CCCGGGCCAAACCAATTGGAAC	CGAT
AACCGG	ACG

X. *Расстояние Хэмминга - 1*

Расстоянием Хэмминга двух строк одинаковой длины называется количество разных символов, имеющих одинаковые индексы в обеих строках. Например расстояние Хэмминга между строками  $s_1 = \text{'ACCGAGT'}$  и  $s_2 = \text{'ACAGAGG'}$  равно 2, так как  $s_1[2] \neq s_2[2]$  и  $s_1[6] \neq s_2[6]$ , а остальные символы попарно равны.

Расстояние Хэмминга — один из простейших примеров понятия расстояния на множестве строк. Даны две непустые ACGT-строки одинаковой длины. Вычислить расстояние Хэмминга между ними.

Input	Output
ACCGAAT ATCGAGT	2
ACGT TGCA	4

Y. *Расстояние Хэмминга - 2*

Дана ACGT-строка длины  $N$ . Рассмотрим все её циклические сдвиги вправо (на  $1, 2, \dots, N - 1$  символов). Вывести  $N - 1$  число — расстояния Хэмминга между исходной строкой и каждым её циклическим сдвигом.

Input	Output
ACCACT	5 5 2 5 5

Комментарий к примеру: у данной строки следующие циклические сдвиги:

ТАССАС  
СТАССА  
АСТАСС  
САСТАС  
ССАСТА

Соответствующие расстояния Хэмминга равны 5, 5, 2, 5, 5.

Z. *Составить строку*

Даны две ACGT-строки  $S_1$  и  $S_2$ . Представим, что вторая строка составлена из карточек с буквами, которые можно брать в каком угодно порядке.

Сколько экземпляров строки  $S_2$  надо взять, чтобы из всех этих карточек с буквами (возможно, используя только часть из них) можно было составить строку  $S_1$ ?

Вывести одно натуральное число — ответ на вопрос задачи, или число  $-1$ , если сделать это невозможно.

Input	Output
AAAAAAAAAGGG ACCATG	4
ACGACGT ACG	-1

ZA. *Расшифровка*

Широко известный алгоритм шифрования, т.н. шифр подстановки, заключается в том, что каждый символ исходного сообщения заменяется на другой (одинаковые символы переходят в одинаковые). Такой шифр легко взломать, если частоты букв в исходном сообщении заметно различаются (например, частоты букв Е, А, О гораздо больше частот букв Ж или Ю). Один из способов усложнить задачу взломщику — менять правила замены для каждого символа.

Зашифруем фразу ANEFFETETOMATOOFMONTANAOFTEMONFATTOFFEE. Для этого выберем *секретное слово*, например, TRYME. Припишем его к исходной строке слева, и получившуюся строку обрежем справа так, чтобы её длина стала равна длине исходной строки. Мы получили ключ сообщения.

Теперь каждый символ исходной строки циклически сдвинем на величину, соответствующую букве ключа. Буква А ключа означает сдвиг на 0 позиций, буква В ключа означает сдвиг на 1 позицию и т.д. Ниже приведён пример шифрования, изучите его.

ANEFFETETOMATOOFMONTANAOFTEMONFATTOFFEE	(исходный текст)
TRYMEANEFFETETOMATOOFMONTANAOFTEMONFAT	(ключ)
<hr/>	
TECRJEGIYTQTХНСRМВНHFZOVYTRNOYXQBNYEFHVKFXX	(зашифрованное сообщение)

Напишите программу, которая по зашифрованному сообщению и секретному слову восстанавливает исходное сообщение.

Input	Output
TECRJEGIYTQTXHCRMHBHFZOBYTRNOYXQBNEYFHBKFX TRYME	ANEFETETOMATOOFMONTANAOFTENATEMONFATTOFEE