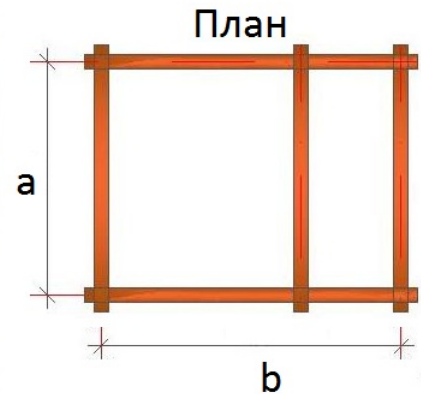


Задача 1. Пятистенок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Изба-пятистенка или пятистенок — жилая деревянная прямоугольная постройка, разделенная внутренней поперечной стеной на две неравные части: избу (горницу) и сени. Пятая стена связывает между собой две длинные стены и делает конструкцию более прочной — не даст разъехаться связанным стенам.



2100 год. Схема сборки избы осталась прежней, а вот дерево заменено более стойким к внешним воздействиям полимерным материалом. Строители из длинной заготовки длины c отрезают бревна нужной длины и укладывают их друг на друга. На фундамент кладут два длинных бревна длины b , на них — три коротких длины a , снова два длинных, опять три коротких, и так далее. Самый верхний ряд всегда делают из трех коротких бревен.

По данным значениям a , b и c определите максимальную высоту избы, которую можно построить из одной заготовки. Каждые пять уложенных брёвен (два длинных и три коротких) увеличивают высоту дома на 1.

Формат входных данных

Программа получает на вход три целых числа a , b и c — длины брёвен и заготовки ($1 \leq a < b < c \leq 10^{18}$), записанных в отдельных строках.

Обратите внимание, что для считывания данных необходимо использовать 64-битный тип данных, например `long long` в C++, `int64` в Free Pascal, `long` в Java.

Формат выходных данных

Программа должна вывести одно неотрицательное целое число — максимальную высоту избы, которую можно построить из заготовки.

Система оценивания

Решение, правильно работающее только для случаев, когда входные числа не превосходят 10^5 , будет оцениваться в 50 баллов.

Примеры

стандартный ввод	стандартный вывод
3 5 29	1
1 2 100	14

Замечание

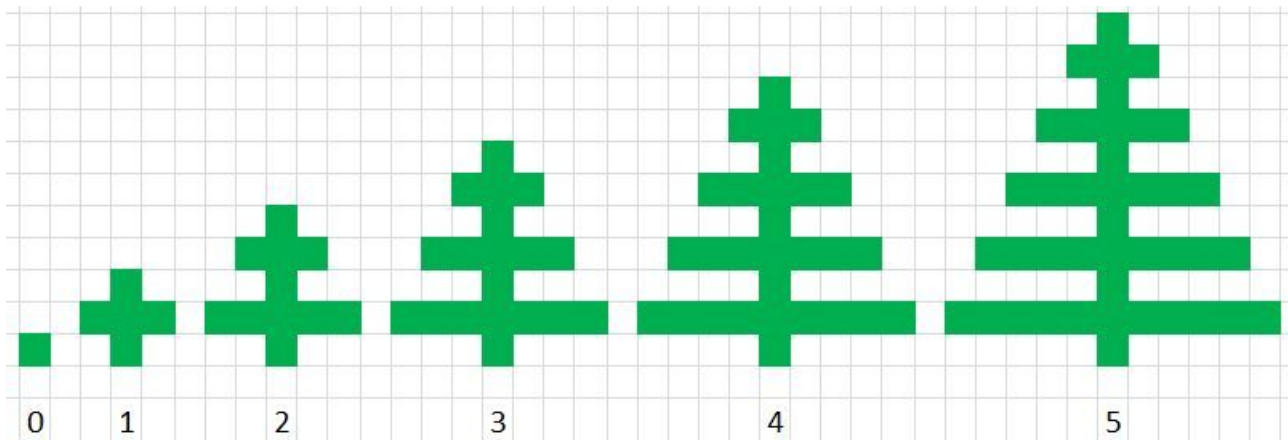
В первом примере строители уложат в первый ряд два продольных бревна, отрезав от заготовки длиной 29 ровно 10 единиц длины. Потом уложат три поперечных бревна, отрезав от заготовки еще 9 единиц длины. Уложено 5 бревен, высота избы 1. От заготовки осталось 10 единиц длины, их как раз хватит на ряд из длинных бревен, но на следующий ряд заготовки уже не хватит.

Задача 2. Ёлочки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	512 мегабайт

На сегодняшнем уроке ИЗО весь класс рисует зимний лес. К сожалению, с передачей художественных образов изобразительными методами дела у Тимофея обстоят из рук вон плохо. Но хоть что-то нарисовать нужно, поэтому Тимофей рисует елочки по клеточкам.

Каждая елочка имеет свою **красоту**, равную количеству ветвей с одной стороны ствола и (так уж совпало) длине самой нижней ветви. Каждая следующая верхняя ветка на одну клетку короче предыдущей. Между ветвями, а также под самой нижней и над самой верхней ветвями находится ствол дерева шириной ровно в одну клетку. На рисунке вы видите елки кисти Тимофея красотой от 0 до 5 включительно.



Поскольку с математическими формулами Тимофей дружит гораздо сильнее, чем с акварельными красками, его заинтересовал вопрос, какую площадь занимает клетчатая елка определенной красоты. Тимофей без труда решил эту задачу. А вы сможете?

Формат входных данных

Программа получает на вход одно целое число n — красоту ёлки ($0 \leq n \leq 2 \cdot 10^9$).

Обратите внимание, что при заданных ограничениях для хранения ответа необходимо использовать 64-битный тип данных, например **long long** в C++, **int64** в Free Pascal, **long** в Java.

Формат выходных данных

Программа должна вывести одно целое число — площадь елки красоты n .

Система оценивания

Решение, правильно работающее в случае, когда $n \leq 100$, будет оцениваться 60 баллов.

Пример

стандартный ввод	стандартный вывод
5	41

Задача 3. Соревнование делимости

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 512 мегабайт

Кате нравятся целые числа, которые делятся без остатка на число K , а Маше — целые числа, которые делятся без остатка на число M . Сегодня подруги решили утроить соревнование и выяснить, чьи любимые числа лучше.

Для начала они выписали на лист бумаги все целые числа от A до B включительно. Затем Катя посчитала, сколько чисел среди выписанных делятся на число K без остатка, а Маша посчитала, сколько чисел делятся на число M без остатка.

В соревновании победит тот из них, чьих любимых чисел окажется больше. Если же количества любимых чисел Кати и Маши совпадут, объявляется ничья. Для того, чтобы определить победителя, девочки попросили вас вычислить разность количества любимых чисел Кати и Маши.

Формат входных данных

Программа получает на вход четыре целых положительных числа, записанных в отдельных строках: K , M , A и B . Числа не превосходят 2×10^9 .

Формат выходных данных

Программа должна вывести одно целое число — разность количества любимых чисел Кати и количества любимых чисел Маши.

Система оценивания

Решения, правильно работающие только для случаев, когда входные числа не превосходят 100, будут оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
2 3 2 9	1
3 3 6 6	0
10 2 1 5	-2

Замечание

В первом примере выписаны числа 2, 3, 4, 5, 6, 7, 8, 9. Среди них есть четыре числа, которые делятся на 2: 2, 4, 6, 8, и три числа, которые делятся на 3: 3, 6, 9. Ответ: $4 - 3 = 1$.

Во втором примере выписано одно число 6 и оно является любимым числом как Кати, так и Маши.

В третьем примере среди чисел 1, 2, 3, 4, 5 нет ни одного любимого числа Кати, а у Маши любимыми являются 2 и 4.

Задача 4. Задача из ЕГЭ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 512 мегабайт

Тимофей готовится к ЕГЭ. Для отработки навыка скорости и точности поиска ответов на задания по теме «Системы счисления» ему часто приходится решать примеры типа «сколько значащих нулей (или единиц) содержит двоичная запись значения выражения $2^a + 2^b - 2^c$?». Значащими называются все цифры, кроме нулей в начале числа (которые обычно и не записываются). Например, десятичное число 20 в двоичной системе счисления записывается как 10100, и в этой записи две значащие цифры «1» и три значащие цифры «0».

Помогите Тимофею по известным a , b и c узнать ответ на задачу.

Формат входных данных

Программа получает на вход четыре целых неотрицательных числа: a , b , c и d . Числа a , b и c соответствуют показателям степеней двоек в задании ($0 \leq a, b, c \leq 10^9$). При этом **гарантируется**, что $2^a + 2^b - 2^c > 0$ и $a \neq b$.

Число d равно либо 0, либо 1 — цифра, количество которых в значении выражения нужно узнать.

Формат выходных данных

Программа должна вывести одно неотрицательное целое число — ответ на задачу.

Система оценивания

Решение, правильно работающее в случае, когда $a \leq 12$, наберет не менее 40 баллов.

Пример

стандартный ввод	стандартный вывод
4 3 2 1	2

Замечание

В примере нужно узнать количество единиц в двоичной записи значения выражения $2^4 + 2^3 - 2^2$. Вычислим: $16 + 8 - 4 = 20$. $20_{10} = 10100_2$. Всего две единицы.

Такой же результат можно получить, выполнив действия в столбик, не переводя числа в десятичную систему счисления.

$$\begin{array}{r} + 1\ 0\ 0\ 0\ 0 \\ \quad 1\ 0\ 0\ 0 \\ \hline 1\ 1\ 0\ 0\ 0 \end{array} \qquad \begin{array}{r} - 1\ 1\ 0\ 0\ 0 \\ \quad \quad 1\ 0\ 0 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

Задача 5. Бизнесмен Василий

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Бизнесмен Василий готовится к уплате налогов за квартал (3 месяца). Действующая налоговая система в государстве, в котором Василий ведет свой бизнес, устроена таким образом, что величина налога зависит от прибыли в конце каждого месяца. Чистая прибыль бизнесмена определяется как разница между доходом и расходом. Разумеется, если бизнес идет не очень удачно, прибыль бизнесмена может быть отрицательной — в этом случае речь идет об убытке.

Все доходы и расходы Василий записывал в журнал в виде целых чисел. Теперь Василий должен подать налоговую декларацию с суммой доходов на конец каждого месяца, другими словами ему необходимо поделить последовательность записей в журнале на **три непустые** части.

При этом Василий хочет сделать это таким образом, чтобы **прибыль в каждой части была одинаковой** (возможно даже отрицательной) — в этом случае сумма налога будет минимальной. Менять записи в журнале местами нельзя.

По имеющимся данным определите количество способов выполнить такое разбиение.

Формат входных данных

В первой строке входных данных содержится единственное натуральное число N — количество записей в журнале Василия ($3 \leq N \leq 10^5$).

В следующих N строках записаны целые числа a_i , соответствующие записям в журнале ($-10^8 \leq a_i \leq 10^8$).

Формат выходных данных

Выведите единственное целое число - количество способов выполнить необходимое разбиение

Система оценивания

Решение, верно работающее при $n \leq 200$, будет оцениваться в 40 баллов.

Решение, верно работающее при $n \leq 1000$, будет оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
6 4 3 -3 5 -1 4	2
3 0 0 0	1
4 3 -2 3 1	0

Замечание

В первом примере в журнале записано 6 чисел $[4, 3, -3, 5, -1, 4]$ из них можно получить два разбиения: $[4], [3, -3, 5, -1], [4]$ и $[4, 3, -3], [5, -1], [4]$.

Во втором примере в журнале записаны три нуля — имеется единственное возможное разбиение $[0], [0], [0]$, потому что все части должны быть непустыми.

В третьем примере выполнить подходящее разбиение невозможно.

Задача 1. Пятистенок

Для строительства одного ряда нужно три бревна длиной a и два бревна длиной b , поэтому для нахождения ответа нужно нацело поделить n на $3a + 2b$.

```
a = int(input())
b = int(input())
c = int(input())
ans = c // (3 * a + 2 * b)
print(ans)
```

Задача 2. Ёлочки

Ёлочка красоты n состоит из $2n$ ветвей длины 1, 2, ..., n и ствола длины $2n + 1$, поэтому нужно посчитать сумму $s = 1 + 2 + \dots + n$ и вывести значение $2s + 2n + 1$.

Если для нахождения суммы чисел от 1 до n использовать цикл, то решение будет набирать 60 баллов. т.к. для больших n будет работать долго. Пример такого решения

```
n = int(input())
s = 0
for i in range(1, n + 1):
    s += i
print(2 * s + 2 * n + 1)
```

Для того, чтобы набрать 100 баллов, заметим, что $2s = n(n + 1)$, поэтому программа может просто вывести значение $n(n + 1) + 2n + 1$.

```
n = int(input())
print(n * (n + 1) + 2 * n + 1)
```

Задача 3. Соревнование делимости

Для того, чтобы набрать 60 баллов, можно просто перебрать все числа от x до y , посчитать число делящихся на k среди них, посчитать число делящихся на m , и вычесть из одного числа другое. В решении ниже эти количества не считаются отдельно, а считается сразу же ответ. Если число делится на k , то ответ увеличивается на 1, если число делится на m , то ответ уменьшается на 1, одно число может и увеличить, и сразу же уменьшить ответ на 1.

```
k = int(input())
m = int(input())
x = int(input())
y = int(input())

ans = 0
for i in range(x, y + 1):
    if i % k == 0:
        ans += 1
    if i % m == 0:
        ans -= 1
print(ans)
```

Чтобы набрать 100 баллов, нужно быстро посчитать количество чисел, делящихся на данное число на данном отрезке. Количество чисел от 1 до y , делящихся на k , равно $\lfloor \frac{y}{k} \rfloor$ (целая часть от частного $\frac{y}{k}$, то есть целочисленное деление y на k). А чтобы посчитать количество кратных k на отрезке от x до y нужно из количества кратных k на отрезке от 1 до y вычесть количество кратных

k на отрезке от 1 до $x - 1$, то есть $\lfloor \frac{y}{k} \rfloor - \lfloor \frac{x-1}{k} \rfloor$. Аналогично посчитаем количество кратных m на отрезке от x до y . Пример решения.

```
k = int(input())
m = int(input())
x = int(input())
y = int(input())
print(y // k - (x - 1) // k - y // m + (x - 1) // m)
```

Задача 4. Задача из ЕГЭ

40 баллов можно набрать, если реализовать то, что требуется в условии задачи, “в лоб”: вычислить $2^a + 2^b - 2^c$, перевести результат в двоичную систему счисления и посчитать число нулей и единиц.

Пример решения на языке Python.

```
a, b, c, d = [int(input()) for i in range(4)]
n = (1 << a) + (1 << b) - (1 << c)
print(bin(n)[2:].count(str(d)))
```

В этом решении для вычисления степени 2^a используются битовый сдвиг числа 1 на a бит влево: $1 << a$, но можно использовать и возведение в степень: $2 ** a$. Для перевода числа в двоичную систему счисления здесь используется функция `bin` языка Python, а для подсчёта числа символов «0» или «1» используется метод `count`. Если в языке программирования нет встроенной функции перевода числа в двоичную систему счисления, можно сделать это циклом, деля число на 2, например, так.

```
count = [0, 0]
while n > 0:
    count[n % 2] += 1
    n //= 2
```

Чтобы набрать полный балл необходимо разобрать различные случаи соотношения чисел a , b , c . Например, если $a > b > c$, то значение $2^a + 2^b - 2^c$ будет иметь в двоичной системе счисления вид $100...0011...1100...00$, где старшая единица соответствует позиции a , а группа из единиц расположена между позициями b (не включительно) и c (включительно). Необходимо аккуратно разобрать все случаи и для каждого из них посчитать количество нулей и единиц. Пример решения на полный балл.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())

if a < b:
    a, b = b, a

if c <= b:
    if d == 0:
        print(c + a - b)
    else:
        print(b - c + 1)
elif a > c > b:
    if d == 0:
        print(c - 1)
    else:
        print(a - c + 1)
else: # c == a
```

```
if d == 0:
    print(b)
else:
    print(1)
```

Задача 5. Бизнесмен Василий

В этой задаче нужно посчитать количество способов разбить массив на три непрерывных непустых части с равной суммой. Для того, чтобы набрать 40 баллов достаточно перебрать все возможные разбиения, то есть перебрать значения границ i между первой и второй частью и границы j между второй и третьей частью. Посчитаем для каждой части сумму элементов в ней и если три эти суммы равны увеличим ответ на 1. Такое решение имеет сложность $O(n^3)$.

```
n = int(input())
a = [int(input()) for i in range(n)]
ans = 0
for i in range(1, n - 1):
    for j in range(i + 1, n):
        if sum(a[:i]) == sum(a[i:j]) == sum(a[j:]):
            ans += 1
print(ans)
```

Можно ускорить это решение, если сумму элементов в каждой части вычислять за $O(1)$, а не за $O(n)$. Это можно сделать при помощи префиксных сумм, или следующим образом: передвигая границу одной части на один элемент вправо, будем добавлять значение этого элемента к сумме той части, к которой добавился этот элемент. Если сумма всех элементов в массиве равна s , то значение s должно делиться на 3, и тогда достаточно проверить, что суммы первой и второй части равны $s/3$. Общая сложность решения будет $O(n^2)$. Такое решение набирает 60 баллов.

```
n = int(input())
a = [int(input()) for i in range(n)]
s = sum(a)
ans = 0
if s % 3 == 0:
    s = s // 3
    s1 = 0
    for i in range(n - 2):
        s1 += a[i]
        if s1 == s:
            s2 = 0
            for j in range(i + 1, n - 1):
                s2 += a[j]
                if s2 == s:
                    ans += 1
print(ans)
```

Уменьшим сложность решения до $O(n)$. Для этого будем перебирать значение j — границы между второй и третьей части. Если сумма элементов левее этой границы равна $2s/3$, то это подходящая граница. Также нам нужно знать количество подходящих границ между первой и второй частью левее данной границы — это количество таких границ, левее которых сумма элементов будет равна $s/3$. Количество таких подходящих границ будем хранить в переменной `count`, которую будем увеличивать, если значение текущей суммы (слева от рассматриваемой границы) равно $s/3$. А если значение текущей суммы равно $2s/3$, то будем к ответу добавлять значение `count`.

```
n = int(input())
a = [int(input()) for i in range(n)]
s = sum(a)
```

```
ans = 0
if s % 3 != 0:
    s //= 3
count = 0
curr_sum = 0
for elem in a[::-1]:
    curr_sum += elem
    if curr_sum == 2 * s:
        ans += count
    if curr_sum == s:
        count += 1
print(ans)
```