

Задача А. МКАД

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Длина Московской кольцевой автомобильной дороги — 109 километров. Байкер Вася стартует с нулевого километра МКАД и едет с постоянной скоростью v километров в час. На какой отметке МКАД он остановится через t часов?

Формат входных данных

Программа получает на вход два целых числа: значения v и t , записанные в отдельных строках. Если $v > 0$, то Вася движется в положительном направлении по МКАД, если же значение $v < 0$, то в отрицательном. Гарантируется, что $v \neq 0$.

Формат выходных данных

Программа должна вывести целое число от 0 до 108 — номер отметки, на которой остановится Вася.

Примеры

стандартный ввод	стандартный вывод
60 2	11
-1 1	108

Задача В. Планета Арифмет

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Где-то в далекой галактике есть планета Арифмет. Населяют ее небольшие, человеко-подобные существа — Арифметяне. В толще планеты находится единственный ценный ресурс — Числы. Числы — это большие красивые кристаллы, состоящие из слипшихся Цифров. Каждая Числа имеет эквивалент в обычных, Земных числах. Многие века Арифметяне добывают Числы, которые необходимы им, чтобы выжить. Но однажды на их планету упала Белая Вычислительная коробка С Надкушенным Яблоком и сказала им человеческим голосом: «Простые Числы вкуснее». Однако после падения коробка повредилась и забыла, что такое Простые Числы. Вам, как самому умному Арифметянину предстоит разобраться с этой коробкой и заставить ее говорить, вкусная ли данная ей Числа. Единственная зацепка, которая у вас есть — это уцелевшие на коробке древние письмена:

```
def is_prime(n):  
    # Jwo oISJi skjewo ...  
  
number = int(input())  
print(is_prime(number))
```

Восстановите повреждённую часть!

Формат входных данных

В коробку вводится одно число N ($1 \leq N \leq 10^9$).

Формат выходных данных

Коробка должна сказать `True`, если Числа вкусная, и `False`, если Числа невкусная.

Примеры

<code>stdin</code>	<code>stdout</code>
2	True
10	False

Задача С. Разложение на множители

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 0.25 секунда
Ограничение по памяти: 64 мегабайта

Дано число. Требуется разложить его на простые множители.

Формат входных данных

Вводится число N ($2 \leq N \leq 10^9$).

Формат выходных данных

Выведите через пробел разложение на простые множители в порядке возрастания множителей.

Примеры

<code>stdin</code>	<code>stdout</code>
17	17
60	2 2 3 5

Задача D. МегаНОД

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано несколько чисел. Найти самое большое число, на которое делятся все несколько чисел.

Формат входных данных

В единственной строке даны через пробел несколько чисел ($1 \leq \text{несколько} \leq 1000$, $1 \leq \text{каждое} \leq 10^9$).

Формат выходных данных

Выведите искомое число.

Примеры

<code>stdin</code>	<code>stdout</code>
18 30 21	3

Задача Е. Решето Эратосфена

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

По введённым числам A и B вывести все простые числа в интервале от A до B включительно.

Формат входных данных

В единственной строке вводятся два числа $1 \leq A \leq B \leq 100\,000$

Формат выходных данных

Вывести в одну строку все простые числа в интервале от A до B включительно

Примеры

<code>sieve.in</code>	<code>sieve.out</code>
2 2	2
1 100	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Задача F. За мной просили не занимать

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На одну ленту, вставляемую в кассовый аппарат, помещается L строк текста. Описание каждого приобретаемого товара занимает в чеке m строк, никакой дополнительной информации в чеке не печатается. Кассир меняет ленту либо в конце дня, либо когда видит, что покупки следующего покупателя не поместятся на ленте. Ваша задача — определить, после какого покупателя кассиру придется впервые за день поменять ленту в аппарате

Формат входных данных

В первой строке заданы два числа: длина ленты L ($1 \leq L \leq 1000$) и количество строк, которые занимает один товар ($1 \leq m \leq 50$).

Во второй строке задано число n — количество покупателей, пришедших в течение дня ($1 \leq n \leq 1000$).

В третьей — через пробел указано количество товаров, приобретаемых каждым покупателем. Гарантируется, что покупки каждого покупателя могут уместиться на одной ленте.

Формат выходных данных

Одно число — номер покупателя, после которого придется поменять ленту (покупатели нумеруются с 1).

Примеры

<code>stdin</code>	<code>stdout</code>
10 1 5 2 4 3 5 1	3
7 2 4 1 2 1 1	2

Замечание

Во втором тесте: первые два покупателя совершили в сумме 3 покупки и заняли на ленте 6 строк. Осталась одна свободная строка. Покупки третьего покупателя занимают 2 строки и не влезают на ленту, следовательно перед ним ленту надо менять.

Задача G. Количество делителей

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Для улучшения качества обучения в параллели С' преподаватели решили ввести индекс счастья ученика. В начале смены индекс счастья каждого ученика равен 1. Каждый последующий день смены индекс счастья может увеличиться или уменьшиться в m раз. Статистика показывает, что количество решенных задач на практике прямо пропорционально количеству делителей индекса счастья, поэтому бывают дни, когда индекс счастья не изменяется, а преподаватели считают количество делителей у индекса счастья.

Но так как процесс умножения, деления и подсчета количества делителей достаточно тяжелый, то вам поручено автоматизировать процесс вычислений.

Формат входных данных

В первой строке задается число n ($1 \leq n \leq 10^4$) — количество дней в смене. Каждая из последующих n строк содержит строчку запроса. Строчка запросов имеет один из трех видов: «?», «* m », «/ m » ($1 \leq m \leq 10^{100}$), все простые числа, на которые делится m , не превосходят 10^4 . На запросы второго и третьего вида нужно, соответственно, умножить индекс счастья на m и разделить на m . На запрос первого вида нужно вывести количество делителей индекса счастья, если индекс целый и строчку "Current number is not integer!", если индекс не целый.

Количество операций «* m », «/ m » в сумме не превышает 10^4 .

Формат выходных данных

Для каждого запроса типа «?» выведите одно число — количество делителей, или "Current number is not integer!", если соответствующее число не целое.

Примеры

stdin	stdout
3 * 4 * 3 ?	6
4 * 6 ? / 10 ?	4 Current number is not integer!

Задача Н. Марсианские факториалы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В 3142 году очередная экспедиция на Марс обнаружила в одной из пещер таинственные знаки. Они однозначно доказывали существование на Марсе разумных существ. Однако смысл этих таинственных знаков долгое время оставался неизвестным. Недавно один из ученых, профессор Очень-Умный, заметил один интересный факт: всего в надписях, составленных из этих знаков, встречается ровно K различных символов. Более того, все надписи заканчиваются на длинную последовательность одних и тех же символов.

Вывод, который сделал из своих наблюдений профессор, потряс всех ученых Земли. Он предположил, что эти надписи являются записями факториалов различных натуральных чисел в системе счисления с основанием K . А символы в конце — это конечно же нули, ведь, как известно, факториалы больших чисел заканчиваются большим количеством нулей. Например, в нашей десятичной системе счисления факториалы заканчиваются на нули, начиная с $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$. А у числа $100!$ в конце следует 24 нуля в десятичной системе счисления и 48 нулей в системе счисления с основанием 6 — так что у предположения профессора есть разумные основания!

Теперь ученым срочно нужна программа, которая по заданным числам N и K найдет количество нулей в конце записи в системе счисления с основанием K числа $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N-1) \cdot N$, чтобы они могли проверить свою гипотезу. Вам придется написать им такую программу!

Формат входных данных

В первой строке входных данных содержатся числа N и K , разделенные пробелом, ($1 \leq N \leq 10^9$, $2 \leq K \leq 1000$).

Формат выходных данных

Выведите число X — количество нулей в конце записи числа $N!$ в системе счисления с основанием K .

Примеры

<code>stdin</code>	<code>stdout</code>
5 10	1
100 6	48
100 10	24
3 10	0

Задача I. Привидения

Имя входного файла: `calc-restore.in`
Имя выходного файла: `calc-restore.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Привидения нашли калькулятор. К их несчастью, калькулятор оказался сломан. Когда привидения включили калькулятор, на нем было число 1, а работали только две кнопки:

1. кнопка, увеличивающая значение на 5
2. кнопка, увеличивающая значение в 3 раза

Привидения за ночь напугали N детей и очень хотят, чтобы все об этом знали. Поэтому они решили получить на калькуляторе число N . Но сейчас уже светает, и привидения идут спать, поэтому они попросили вас узнать, можно ли получить такое число N и, если это возможно, последовательность нажатий, необходимую для этого. Конечно же привидениям нужна самая короткая такая последовательность.

Формат входных данных

В первой строки входного файла находится единственное число N ($2 \leq N \leq 10^{18}$) – число, которое хотят получить привидения.

Формат выходных данных

Если получить число N невозможно, то выведите -1. Иначе выведите в первой строке P – количество промежуточных значений, получаемых при последовательности нажатий, приводящей к N . В следующей строке через пробел выведите P чисел – все промежуточные значения (включая 1 и N).

Примеры

<code>calc-restore.in</code>	<code>calc-restore.out</code>
4	-1
8	3 1 3 8