Задача A. Массовая проверка простоты

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Целое число $p \geqslant 2$ является простым, если у него нет делителей кроме 1 и p. Необходимо для всех чисел во входном файле проверить простые они или нет.

Формат входных данных

В первой строке задано число n ($2 \leqslant n \leqslant 500\,000$). В следующих n строках заданы числа a_i ($2 \leqslant a_i \leqslant 2 \cdot 10^7$), которые нужно проверить на простоту

Формат выходных данных

Для каждого числа во входном файле выведите на отдельной строке «`YES`» или «`NO`» в зависимости от того, простое оно или нет.

Пример

стандартный ввод	стандартный вывод
4 60 14 3 55	NO NO YES NO

Задача B. Массовое разложение на множители

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Дано много чисел. Требуется разложить их все на простые множители.

Формат входных данных

В первой строке задано число n ($2 \leqslant n \leqslant 300000$). В следующих n строках заданы числа a_i ($2 \leqslant a_i \leqslant 10^6$), которые нужно разложить на множители.

Формат выходных данных

Для каждого числа выведите в отдельной строке разложение на простые множители в порядке возрастания множителей.

Пример

стандартный ввод	стандартный вывод
4 60 14 3 55	2 2 3 5 2 7 3 5 11

Задача C. Больше простых!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	10 секунд
Ограничение по памяти:	64 мегабайта

Найдите все простые числа не большие n. Поскольку n в этой задаче не просто большое, а прямо здоровенное, для того чтобы проверить, что вы нашли числа правильно, мы попросим вас посчитать от найденных чисел специальный хеш.

Хеш будет считаться по следующему алгоритму. В начале переменная $h = 0$. После каждого очередного встреченного простого числа p_i, будем пересчитывать h по формуле $h = h \cdot x + p_i$, при этом будем игнорировать переполнение знакового 32-битного целого типа. Значение переменной n в конце — это хеш, который вам нужно вывести.

Формат входных данных

Входной файл содержит два числа n ($2 \leqslant n \leqslant 10^9$) и x ($1 \leqslant x \leqslant 10^9$).

Формат выходных данных

Выведите полученный хеш.

Примеры

стандартный ввод	стандартный вывод
10 10	2357
11 100	203050711
1000000000 2	1576840463

Задача D. Проверка на простоту

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дано n натуральных чисел a_i. Определите для каждого числа, является ли оно простым.

Формат входных данных

Программа получает на вход число n, $1 \leqslant n \leqslant 1000$ и далее n чисел a_i, $1 \leqslant a_i \leqslant 10^{18}$.

Формат выходных данных

Если число a_i простое, программа должна вывести `YES`, для составного числа программа должна вывести `NO`.

Пример

стандартный ввод	стандартный вывод
4 1 5 10 239	NO YES NO YES

Задача E. Взлом RSA

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

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа p и q, вычислить $n = pq$ и сгенерировать два числа e и d такие, что $ed \equiv 1 \pmod{(p-1)(q-1)}$ (заметим, что $(p-1)(q-1) = \varphi(n)$). Числа n и e составляют открытый ключ и являются общеизвестными. Число d является секретным ключом, также необходимо хранить в тайне и разложение числа n на простые множители, так как это позволяет вычислить секретный ключ d.

Сообщениями в системе RSA являются числа из $\mathbb{Z}_n$. Пусть M — исходное сообщение. Для его шифрования вычисляется значение $C = M^e \bmod n$ (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение C передается по каналу связи. Для его расшифровки необходимо вычислить значение $M = C^d \bmod n$, а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение C и знаете только открытый ключ: числа n и e. “Взломайте” RSA — расшифруйте сообщение на основе только этих данных.

Формат входных данных

Программа получает на вход три натуральных числа: n, e, C, $n \leqslant 10^9$, $e \leqslant 10^9$, $C < n$. Числа n и e являются частью какой-то реальной схемы RSA, т.е. n является произведением двух простых и e взаимно просто с $\varphi(n)$. Число C является результатом шифрования некоторого сообщения M.

Формат выходных данных

Выведите одно число M ($0 \leqslant M < n$), которое было зашифровано такой криптосхемой.

Примеры

стандартный ввод	стандартный вывод
143 113 41	123
9173503 3 4051753	111111

Задача F. Китайская теорема

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Решите в целых числах систему уравнений

$$\begin{cases} x \equiv a \pmod{n} \\ x \equiv b \pmod{m}, \end{cases}$$

где n и m взаимно просты. Среди решений следует выбрать наименьшее неотрицательное число.

Формат входных данных

Первая строка входных данных содержит число N, $1 \leqslant N \leqslant 10^4$, — количество тестов, для которых нужно решить задачу.

Следующие N строк содержат по четыре целых числа a_i, b_i, n_i и m_i ($1 \leqslant n_i, m_i \leqslant 10^9$, $0 \leqslant a_i < n_i$, $0 \leqslant b_i < m_i$).

Формат выходных данных

Для каждого из тестов выведите искомое наименьшее неотрицательное число x_i.

Пример

стандартный ввод	стандартный вывод

Задача G. Дискретное логарифмирование

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Даны натуральные числа a, b, n. Требуется найти *дискретный логарифм* b по основанию a по модулю n, то есть такое число x ($0 \leqslant x < n$), что $a^x \equiv b \pmod{n}$.

Формат входных данных

В первой строке заданы через пробел три целых числа a, b и n ($0 \leqslant a, b, n \leqslant 10^{12}$), $n \neq 0$.

Формат выходных данных

В первой строке выведите -1, если дискретного логарифма не существует. Иначе следует вывести его значение.

Если ответ неоднозначен, разрешается выводить любой.

Задача H. Задача для второклассника

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам даны два числа. Необходимо найти их произведение.

Формат входных данных

Входные данные состоят из двух строк, на каждой из которых находится целое одно **целое** число, длина которого не превосходит двухсот пятидесяти тысяч символов.

Формат выходных данных

Выведите произведение данных чисел.

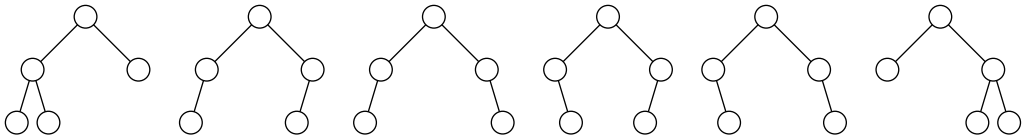
Примеры

стандартный ввод	стандартный вывод
2 2	4
1 -1	-1

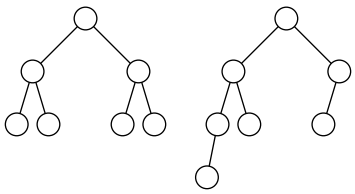
Задача I. AVL

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

AVL деревья, придуманные российскими учёными Адельсон-Вельским и Ландисом, являются примером сбалансированного бинарного дерева поиска. В терминологии AVL, подвешенное бинарное дерево называется *сбалансированным*, если для каждой вершины высоты её левого и правого поддеревьев отличаются не более, чем на один. Такое дерево, собственно, и называется *AVL-деревом*. Разумеется, существует далеко не единственное AVL-дерево при фиксированном числе вершин. К примеру, существует шесть AVL-деревьев с пятью вершинами, они изображены на рисунке ниже.



Деревья с одинаковым числом вершин могут иметь разную высоту, к примеру, на рисунке снизу нарисовано два дерева с семью вершинами, которые имеют высоты 2 и 3, соответственно.



Вам даны два числа — N и H, требуется найти число AVL-деревьев, которые состоят из N вершин и имеют высоту H. Поскольку их число довольно велико, выведите искомое количество по модулю 786 433.

Формат входных данных

Единственная строка входного файла содержит два числа — N и H ($1 \leqslant N \leqslant 65\,535, 0 \leqslant H \leqslant 15$).

Формат выходных данных

Выведите единственное число — количество AVL деревьев с N вершинами высоты H, по модулю 786 433.

Пример

стандартный ввод	стандартный вывод
7 3	16

Замечание

786 433 простое число, и $786\,433 = 3 \cdot 2^{18} + 1$.

Задача J. RSA factoring

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

RSA — криптографическая система, где важную роль играют числа вида $n = pq$, где p и q — различные простые числа. Число n называют модулем RSA и используют для дальнейших вычислений. Стойкость RSA основана на том факте, что для известного числа n не известно достаточно быстрого алгоритма разложения n на множители для достаточно длинных чисел n (от 1024 бит и больше). При этом, рекомендуется выбирать p и q большими случайными простыми числами примерно одинаковой длины. Генерация таких n — процесс, требующий аккуратности и понимания происходящего. Существует большое количество атак на ключи RSA, которые были сгенерированы ненадлежащим образом. Знание других деталей реализации RSA для этой задачи не понадобится.

Прочитав, что в RSA используют близкие простые числа, Карл реализовал свой алгоритм генерации:

1. Сгенерировать случайное простое число p_1, состоящее из b бит.
2. Начиная с $p_1 + 1$, перебрать все числа подряд по возрастанию, пока не встретим следующее простое число p_2.
3. Выдать $n = p_1 p_2$.

Поскольку выбирается случайное простое число p_1, а в среднем расстояние между соседними простыми числами невелико, этот алгоритм достаточно быстро найдет следующее простое число p_2. Друг Карла Пьер обнаружил, что числа, которые выдает алгоритм Карла, можно быстро разложить на делители. Поэтому Пьер предложил брать не два простых числа, а четыре! Независимо от p_1 мы также выберем случайное b-битное простое число q_1 и следующее за ним простое число q_2 и возьмем $n = p_1 p_2 q_1 q_2$. Однако такой способ генерации тоже оказался уязвим: число n возможно разложить на множители.

Вам дано число n, сгенерированное либо изначальным методом Карла с 2 простыми множителями, либо с обновленным методом Пьера с 4 простыми множителями. Разложите его на простые множители.

Формат входных данных

В первой строке заданы два числа b и k ($4 \leqslant b \leqslant 60$, $k = 2$ или $k = 4$). В следующей строке содержится число n в шестнадцатеричной системе счисления, от старших разрядов к младшим, без ведущих нулей.

Гарантируется, что n является произведением ровно k простых множителей, сгенерированных случайно одним из двух методов, описанных в условии. Каждый из этих множителей состоит ровно из b бит в двоичной системе счисления, все множители различны.

Формат выходных данных

Выведите k простых множителей n в шестнадцатеричной записи без ведущих нулей, по одному в каждой строке.

Система оценки

Подзадача	Баллы	Ограничения
1	10	$b \leqslant 8$, $k = 2$
2	10	$b \leqslant 8$, $k = 4$
3	7	$b \leqslant 15$, $k = 2$
4	8	$b \leqslant 15$, $k = 4$
5	15	$b \leqslant 30$, $k = 2$
6	15	$b \leqslant 30$, $k = 4$
7	15	$b \leqslant 60$, $k = 2$
8	20	$b \leqslant 60$, $k = 4$

Примеры

стандартный ввод	стандартный вывод
4 2 8f	b d
6 4 534ee3	25 29 3b 3d

Пояснения к примерам

В первом примере $n = \texttt{8f}_{16} = 143 = 11 \cdot 13$. $\texttt{b}_{16}$ равно числу 11, а $\texttt{d}_{16}$ равно числу 13. Во втором примере задано число $n = 5459683$, что раскладывается в произведение $37 \cdot 41 \cdot 59 \cdot 61$.