

Задача А. Легкое уравнение

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Дано уравнение $m^x = n$. Найдите x .

Формат входных данных

Во входном файле даны два целых числа n и m ($1 \leq n, m \leq 10^7$).

Формат выходных данных

В выходной файл выведите минимальное натуральное число x , удовлетворяющее уравнению. Гарантируется, что такое натуральное x существует.

Примеры

equation.in	equation.out
8 2	3

Задача В. Урок математики

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

В начале каждого урока учительница математики пишет на доске натуральное число N . Пока идет урок, школьники пытаются найти как можно больше простых делителей этого числа. В конце урока щедрая учительница выдает конфетку всем тем, кто нашел максимальное количество простых делителей. Школьник Вася обожает сладкое и просит вас обеспечить ему конфетку от учительницы сегодня. Вам дано число N , нужно вывести все его простые делители в степени, в которой они встречаются в данном числе.

Формат входных данных

Во входном файле единственное натуральное число N ($2 \leq N \leq 10^9$).

Формат выходных данных

В выходной файл выведите все простые делители числа в порядке возрастания и укажите для каждого делителя степень, в которой он встречается в числе N , если его степень больше единицы. Формат вывода должен соответствовать примеру.

Примеры

factoriz.in	factoriz.out
3240	$2^3 * 3^4 * 5$

Задача С. Плюшки

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Вася очень любит плюшки. Однажды ему приснился сон, что он оказался со своим классом на экскурсии на плюшечной фабрике. Там Вася пробрался к плюшечному конвейеру и увидел на нем N плюшек разных видов (у каждого вида свой идентификационный номер на фабрике). Вася решил (как настоящий плюшечный гурман), что будет кушать плюшки только тех видов, которые на конвейере встречаются не более K раз.

Вам даны N чисел, каждое из которых — номер вида плюшки, лежащей на конвейере. Определите, какие плюшеки скушает Вася.

Формат входных данных

В первой строке входного файла даны два натуральных числа $N \leq 1000000$ — количество плюшек на конвейере и $K \leq 100$. Во второй строке заданы N натуральных чисел — номера плюшек. Номер плюшек не превосходят 1000000.

Формат выходных данных

В выходной файл вывести номера всех плюшек, которые съест Вася, отсортированные по убыванию.

Примеры

buns.in	buns.out
9 2 16 19 2 2 3 19 19 3 3	16 2 2

Задача D. Аудитория

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Еще только сентябрь, а школьники уже стараются под любым предлогом не ходить в школу. При этом пришедшие стараются рассестся так, чтобы создать видимость заполненной аудитории.

Заполненность аудитории — это минимум из заполненности по рядам и заполненности по колонкам. Заполненность по рядам — это максимальное число учащихся в одном из рядов, а по колонкам — максимальное число учащихся в одной из колонок.

Помогите школьникам рассестся так, чтобы заполненность аудитории была максимальной.

Формат входных данных

В единственной строке ввода записано три целых числа n , r и c — количество школьников, рядов и колонок в аудитории. $1 \leq r, c \leq 100$, $1 \leq n \leq r \times c$.

Формат выходных данных

На первой строке выведите максимально возможную заполненность аудитории. Далее выведите r строк по c символов в каждой — пример рассадки. При выводе «.» обозначает пустое место, а «#» — занятое.

Если оптимальных решений несколько, выведите любое.

Пример

class.in	class.out
16 4 6	4 .###. #..### #...## ###.##

Задача Е. Переключение между окнами

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Когда пользователь работает в операционной системе Windows, у него часто запущено несколько приложений. Каждое из приложений работает в отдельном окне. Для переключения между окнами используется комбинация клавиш «Alt+Tab». Эта комбинация делает активным окно, в котором пользователь работал перед тем, как перейти в текущее активное окно.

Чтобы переключиться в другое окно, можно нажать клавишу «Alt» и затем, не отпуская ее, несколько раз нажать клавишу «Tab». Чтобы понять, какое окно станет активным после этого, воспользуемся следующей моделью. Пусть запущено n приложений. Приложения в операционной системе организованы в виде списка и упорядочены по убыванию времени последней активности. То есть приложение, окно которого является активным в настоящий момент — первое в списке, приложение, окно которого было активно перед этим — второе, и т. д.

Если нажать клавишу «Alt» и затем, не отпуская ее, нажать клавишу «Tab» k раз, то активным станет окно приложения, которое находится на $(k \bmod n) + 1$ -м месте в списке. Здесь $a \bmod b$ означает остаток от деления a на b . Иными словами, операционная система рассматривает список как циклический, переходя после последнего элемента списка к первому.

При запуске нового приложения оно добавляется в начало списка.

Задана последовательность действий пользователя, где каждое действие — либо запуск приложения, либо переключение между окнами. Выведите в выходной файл, в каком порядке пользователь работал с приложениями.

Формат входных данных

Первая строка входного файла содержит целое число n — количество действий пользователя ($1 \leq n \leq 1000$). Следующие n строк содержат описание действий пользователя.

Запуск приложения описывается строкой «Run имя приложения». Здесь «имя приложения» — строка из не более чем 100 латинских букв, цифр и пробелов. Она отделена от слова «Run» ровно одним пробелом. Все имена приложений различны. Большие и маленькие буквы считаются различными.

Переключение между приложениями описывается строкой «Alt+Tab+...+Tab», здесь подстрока «+Tab» повторена в точности столько раз, сколько раз пользователь нажал клавишу «Tab», не отпуская клавишу «Alt». Это количество не превышает 100.

Первая команда во входном файле — всегда команда «Run».

Формат выходных данных

Выведите в выходной файл n строк — последовательность имен приложений, с которыми работал пользователь в порядке, в котором их окна становились активными.

Примеры

windows.in	windows.out
6	Mozilla Firefox
Run Mozilla Firefox	Free Pascal
Run Free Pascal	Mozilla Firefox
Alt+Tab	Miranda IM
Run Miranda IM	Free Pascal
Alt+Tab+Tab	Free Pascal
Alt+Tab+Tab+Tab	

Задача F. Книжная полка

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Вася раскладывает свои книги на полку. Если на полке нет ни одной книги, то он просто ставит её, если есть, то ставит либо справа, либо слева от уже расставленных книг. Забирает книги он так же, то есть снимает только с правого или левого края.

Формат входных данных

В первой строке содержится число N ($1 \leq N \leq 10000$) — количество операций, которые выполнил Вася. Далее в N строках находится информация об операциях. Каждая операция постановки книги на полку описывается парой чисел. Первое из них (1 или 2) показывает, книга ставится с левого края или с правого соответственно, второе целое число (от 0 до 10000) обозначает номер книги. Операции снятия книги с полки описывается одним числом — 3 или 4, с левого и правого края соответственно.

Формат выходных данных

Для каждой операции снятия книги с полки вывести номер снимаемой книги.

Пример

shelf.in	shelf.out
5	3
1 1	2
2 2	
1 3	
3	
4	