

Задача А. Степень

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Всё, что требуется сделать в данной задаче – найти последнюю цифру числа 5^n .

Формат входных данных

Целое число n , не превосходящее по модулю 10^9 .

Формат выходных данных

Выведите одну цифру – ответ

Примеры

стандартный ввод	стандартный вывод
2	5
-2	4

Замечание

Пояснение ко второму примеру: $5^{-2} = 0.04$, последняя цифра = 4.

Задача В. Массивы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Вам дано N линейных массивов целых чисел, пронумерованных по порядку от 0 до $N - 1$. Над массивами последовательно выполняется набор следующих действий: переместить все элементы массива A_i в конец массива A_j (порядок элементов сохраняется). Массив A_i при этом становится пустым.

Ваша задача – вывести значения массивов после выполнения всех операций.

Формат входных данных

В первой строке входного файла содержится целое число N ($1 \leq N \leq 10^5$) — количество массивов.

В следующих N строках идут описания массивов: сначала целое число Q ($0 \leq Q \leq 2 \cdot 10^5$) — количество элементов в очередном массиве. Далее идут сами элементы, разделённые пробелом — целые числа в диапазоне от 0 до 10^9 . Суммарное количество элементов во всех массивах не превышает $2 \cdot 10^5$.

В следующей строке содержится одно число M ($0 \leq M \leq 2 \cdot 10^5$) — количество операций над массивами.

В последних M строках содержатся по 2 целых числа i и j , разделённые пробелом ($i \neq j$, $0 \leq i, j < N$) — указание переместить элементы массива A_i в конец массива A_j .

Формат выходных данных

Выведите N строк — новые значения массивов в том же формате, как и во входных данных (сначала количество чисел, потом сами значения).

Пример

стандартный ввод	стандартный вывод
3	0
1 1	0
1 2	3 3 2 1
1 3	
2	
0 1	
1 2	

Задача С. Miner

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Маленькая девочка Виолетта уже несколько месяцев ходит на кружок по программированию. Недавно она решила, что, как и любой другой программист, она обязательно должна написать игру. В результате ночи разработки к утру получилась игра, которую Виолетта назвала «Miner».

Правила игры довольно просты. Цель игры – поиск мин на двухмерном квадратном поле. Нажав на любой клетке поля правую кнопку мышки, вы размещаете в этой клетке флажок, означающий, что, по вашему мнению, в этой клетке поля находится мина. Нажав левую кнопку мышки, можно открыть клетку, в которой (по вашему мнению) мины нет. Если в этой клетке была мина, то вы проигрываете. В противном случае в клетке либо отображается число от 1 до 8, показывающее, сколько мин находится вокруг данной клетки, либо, если вокруг данной клетки мин нет, открывается не только данная клетка, но и все ещё не открытые соседние. Если среди открывшихся клеток также есть такие, вокруг которых нет ни одной мины, то открываются и все их соседние свободные клетки, и так далее. Цель игры – открыть все свободные клетки поля.



Уже утром Виолетта обнаружила две досадные проблемы. Во-первых, оказывается, очень похожую игру уже написали до нее ;-) А во-вторых, она сама с трудом в неё выигрывает. Подумав, Виолетта решила упростить правила. В новой версии игры положение всех мин известно заранее, а цель игры – открыть все остальные клетки поля, выполнив при этом как можно меньше нажатий левой кнопки мышки.

Так как Виолетта – начинающий программист, то поставленная задача оказалась сложной для нее, поэтому она обратилась к Вам. Помогите Виолетте решить поставленную задачу.

Формат входных данных

Первая строка входных данных содержит натуральное число N ($5 \leq N \leq 1000$) – размер поля. Каждая из последующих N строк содержит по N символов, обозначающих статус клетки. Символ '@' обозначает мину, символ '.' – отсутствие мины.

Формат выходных данных

Выведите одно число – минимальное число нажатий левой кнопки мышки

Пример

стандартный ввод	стандартный вывод
9 ...@...@..@...@.. ...@.. @@...@...@@..	24

Задача D. Криптография

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Тайные агенты одной секретной организации (название которой мы по понятным причинам сообщать не будем) в своей переписке используют шифрование. Одним из этапов генерации ключей для этого шифрования является нахождение положительного корня n -ой натуральной степени из натурального числа p .

Так как среди агентов не так много программистов и все агенты-программисты заняты разработкой Очень Важной Программы, то помочь агентам с шифрованием придется вам.

Формат входных данных

В единственной строке заданы два числа n и p ($1 \leq n \leq 32767$, $1 \leq p \leq 10^{10000}$).

Формат выходных данных

В единственной строке необходимо вывести искомый корень. Гарантируется, что корень является натуральным числом, не превышающим 10^9 .

Примеры

стандартный ввод	стандартный вывод
2 16	4
7 4357186184021382204544	1234

Задача Е. Обработка видео

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

С видеокамеры поступают кадры. Они помещаются в общую очередь на обработку на видеокарте. Известен момент времени появления каждого кадра в этой очереди. Каждый кадр должен пройти через обработку на одной из видеокарт, после чего результат обработки отправляется клиенту.

Всего имеется N видеокарт, каждая из которых обрабатывает кадр за T мс. Одновременно одна видеокарта не может обрабатывать больше одного кадра. Если все видеокарты заняты, то кадр в начале очереди ждёт, пока одна из карт не освободится.

Результаты обработки кадров со всех видеокарт помещаются во вторую общую очередь, которая обрабатывается отдельным процессом по следующему принципу. Пока очередь пуста, процесс ждёт появления первого результата. Как только результат появляется, процесс тут же начинает передавать его клиенту. Время передачи составляет V мс. Соответственно, через V мс после начала передачи процесс будет готов передавать следующий результат.

Для каждого кадра необходимо вычислить момент времени, когда результат по его обработке появится у клиента. Можно считать, что операции добавления и извлечения из очередей происходят мгновенно.

Формат входных данных

В первой строке записаны 4 числа через пробел:

1. F ($1 \leq F \leq 10^5$) – количество кадров для обработки.
2. N ($1 \leq N \leq 10$) – количество видеокарт.
3. T ($1 \leq T \leq 100$) – время обработки кадра на видеокарте.
4. V ($1 \leq V \leq 100$) – время на передачу результатов обработки клиенту.

В следующих F строках записано по одному числу X ($1 \leq X \leq 10^9$) – времена появления кадров. Все времена различны и даны в порядке возрастания.

Формат выходных данных

Для каждого кадра выведите в отдельной строке время получения результата обработки кадра клиентом.

Ваша программа должна работать в *интерактивном режиме*, то есть выводить ответ сразу после чтения очередного входного числа. Следующее входное число станет доступно для чтения лишь после вывода ответа на предыдущее.

После вывода каждого ответа программа должна выполнить операцию *flush* (см. примечание ниже).

Примеры

стандартный ввод	стандартный вывод
4 2 10 100 1 2 500 510	111 211 610 710
8 2 5 1 1 2 3 4 500 510 520 530	7 8 12 13 506 516 526 536

Замечание

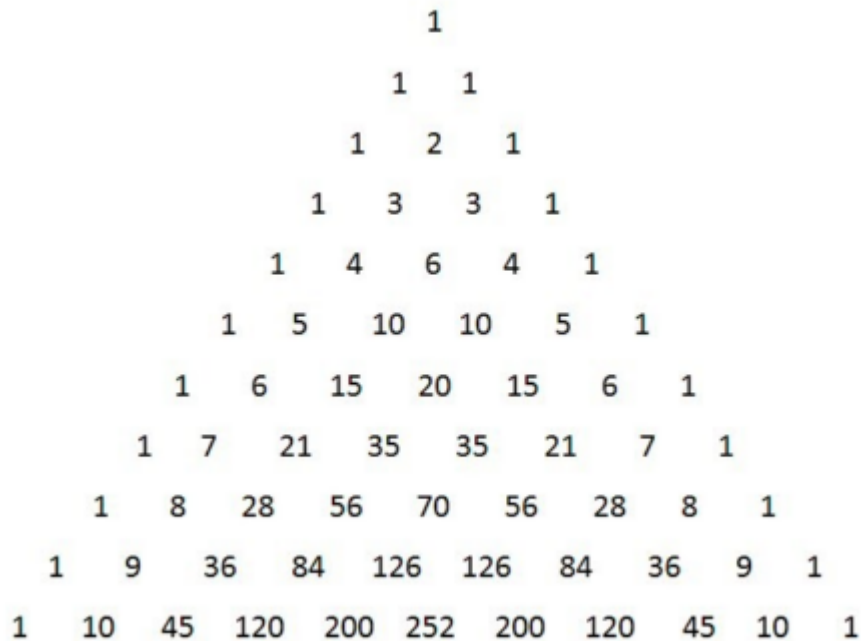
Операция *flush* (сброс данных из буфера в поток) для разных языков выполняется следующим образом:

- **с++ с библиотекой `iostream`**: `std::cout << std::flush;` *либо* `std::cout << std::endl;` (во втором варианте – также перевод строки)
- **с++ с библиотекой `stdio.h`**: `fflush(stdout);`
- **pascal**: `flush(output);`
- **java**: `PrintWriter out = new PrintWriter(System.out); ... out.flush();`
- **C#, Visual Basic**: `Console.Out.Flush()`
- **Python**: `import sys ... sys.stdout.flush()`

Задача F. Треугольник Паскаля

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Наверное, вы знакомы с треугольником Паскаля. Напомним, что треугольник Паскаля – бесконечная таблица биномиальных коэффициентов, имеющая треугольную форму. В этом треугольнике на вершине и по бокам стоят единицы. Каждое другое число равно сумме двух расположенных над ним чисел.



Математик Иван предложил программисту Тимофею взять треугольник Паскаля с количеством строк N и заменить в нём все числа на их остатки от деления на 3. Теперь Иван и Тимофей хотят посчитать количество ненулевых элементов в получившемся треугольнике. Помогите им это сделать.

Формат входных данных

В единственной строке ввода дано натуральное число $1 \leq N \leq 10^9$.

Формат выходных данных

В ответе напечатайте одно целое число – количество ненулевых элементов.

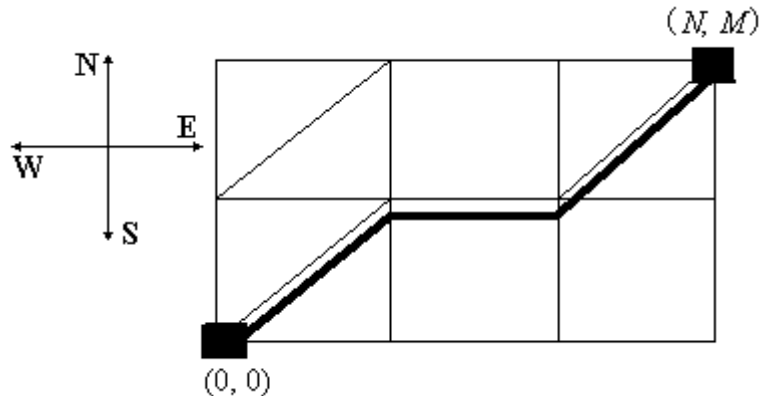
Примеры

стандартный ввод	стандартный вывод
2	3
4	8

Задача G. Кратчайший путь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Студент Василий живёт в городе, где улицы образуют правильную сетку кварталов: все кварталы являются квадратами с длиной стороны, равной 100 метрам. Дом Василия располагается в юго-западном углу – точке с координатами $(0, 0)$. Университет располагается в северо-восточном углу – точке с координатами (N, M) .



Выйдя утром из дома, Василий идёт в университет. Он движется по улицам, но при этом некоторые кварталы может пересечь также по диагонали, ведущей из юго-западного угла квартала в северо-восточный.

Напишите программу, которая вычислит длину кратчайшего маршрута от дома до университета.

Формат входных данных

В первой строке находятся два целых числа N и M ($0 < N, M \leq 10^9$) – размер сетки кварталов с запада на восток (по горизонтали) и с юга на север (по вертикали) соответственно.

Во второй строке находится целое число K ($0 \leq K \leq 10^5$) – количество кварталов, через которые можно пройти наискосок.

Далее следуют K строк с парами целых положительных чисел x и y , разделённых пробелами – координаты северо-восточных углов таких кварталов ($1 \leq x \leq N, 1 \leq y \leq M$).

Формат выходных данных

Требуется вывести длину кратчайшего пути от дома Василия до университета в метрах, округлённую до ближайшего целого.

Пример

стандартный ввод	стандартный вывод
3 2 3 1 1 3 2 1 2	383

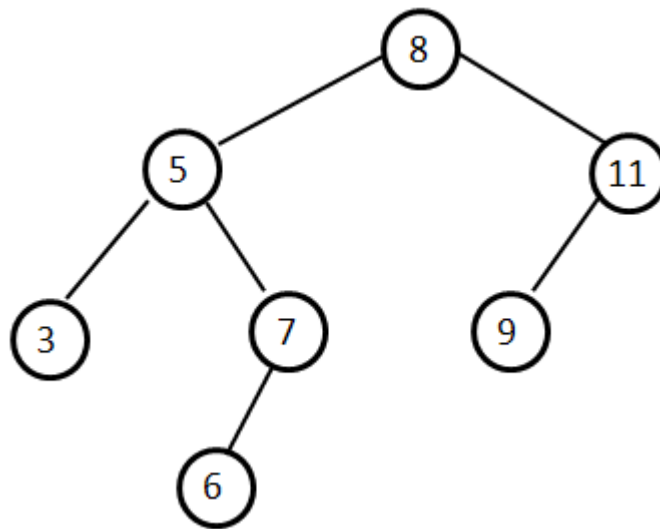
Задача Н. Двоичное дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Одним из способов представления множеств в памяти компьютера является простое (несбалансированное) двоичное дерево поиска. Каждый узел данного дерева может иметь не более двух детей (левый и правый сын).

Вставка нового элемента x в дерево начинается с корневого узла. Сравним x с ключом в корне. Если они равны, то делать ничего не надо – элемент уже содержится в дереве. Если x меньше ключа в корне, то идём в левое поддерево, если больше – в правое. Продолжаем делать так до тех пор, пока идти станет больше некуда – в этом случае создаётся новый листовой узел.

На рисунке показано дерево, полученное из пустого дерева после вставки элементов 8, 5, 7, 11, 9, 3, 6.



Высотой дерева называется длина максимального пути от корня до листа. Например, для дерева на рисунке высота равна 3, так как путь от корня до листа с ключом 6 имеет длину 3.

Вам дана последовательность элементов, которые поочерёдно вставлялись в изначально пустое дерево. Требуется найти его высоту.

Формат входных данных

В первой строке входных данных записано целое число N ($1 \leq N \leq 10^5$) – количество элементов. В следующей строке через пробел записаны N целых чисел – вставляемые элементы (все элементы не превышают по модулю 10^9).

Формат выходных данных

Выведите одно целое число – высоту дерева.

Примеры

стандартный ввод	стандартный вывод
7 8 5 7 11 9 3 6	3
3 5 5 5	0

Задача I. Гири

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Имеется N гирь с различными весами. На стрелочные весы положили несколько гирь из данного набора и определили их суммарный вес. Известно, сколько гирь лежит на весах, но неизвестно, какие именно. Можно ли однозначно определить, какие гири отсутствуют?

Формат входных данных

В первой строке входных записано целое число N ($1 \leq N \leq 50$). В следующей строке записаны через пробел N неповторяющихся целых чисел $w_1, w_2, \dots, w_N$ ($1 \leq w_i \leq 10^5$) — веса гирь. В последней строке записаны через пробел два целых числа K и S — количество гирь на весах и их суммарный вес соответственно ($0 \leq K \leq N$, $0 \leq S \leq \sum_{i=1}^N w_i$).

Гарантируется, что существует подмножество из K гирь с суммой весов, равной S .

Формат выходных данных

Выведите слово Yes, если можно однозначно определить, какие гири не лежат на весах, и слово No — в противном случае.

Примеры

стандартный ввод	стандартный вывод
4 3 2 1 4 2 6	Yes
4 3 2 1 4 2 5	No

Задача J. Поисковый индекс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Аспирантка Анна занимается разработкой нового метода индексирования для ускорения нечёткого поиска. Анна решила, что множество ключей её индекса будет определяться следующим образом.

Пусть имеется набор из N входных текстовых документов. Назовём непустую строку *полезной*, если она имеет длину не более L и встречается в качестве подстроки не менее чем в одном и не более чем в T документах. Тогда множество ключей индекса строится так:

- На первом шаге включим в результирующее множество все полезные строки.
- На втором шаге удалим из ответа лишние элементы, пользуясь следующим правилом: если в множестве присутствуют два элемента s_1 и s_2 , такие что s_1 является подстрокой s_2 , то элемент s_2 удаляется.

Рассмотрим пример. Пусть дано три входных документа – 'aba', 'caba' и 'dab', при этом $T = 2$ и $L = 3$.

После первого шага получаем множество 'aba', 'ba', 'c', 'ca', 'cab', 'd', 'da', 'dab'. Подстроки 'a' и 'b' в него не попали, так как встречаются более чем в T документах.

После удаления лишних элементов на втором шаге окончательный ответ будет равен 'ba', 'c', 'd'.

Анна заметила, что построение индекса непосредственно вышеописанным способом может выполняться слишком долго. Ваша задача – придумать и реализовать эффективный алгоритм построения результирующего множества.

Формат входных данных

В первой строке входных данных записаны три натуральных числа N , T и L ($1 \leq N, T \leq 10^6$, $1 \leq L \leq 10$).

Далее идут N входных документов – непустых строк, содержащих только строчные английские буквы. Сумма длин всех строк не превышает 10^6 .

Формат выходных данных

Выведите элементы результирующего множества в лексикографическом порядке

Примеры

стандартный ввод	стандартный вывод
3 2 3 aba caba dab	ba c d
2 1 5 aaaaaaaa aaaaaaaa	
2 1 5 aaaaaaca aaaaaaab	b c