Задача A. Антипалиндромные числа

Имя входного файла:	`anti.in`
Имя выходного файла:	`anti.out`
Ограничение по времени:	4 секнды
Ограничение по памяти:	64 мегабайта

Палиндром — это слово, которое читается одинаково от в обе стороны. Определим аналогичным образом *антипалиндром*, как слово, у которого i-й с начала и i-й с конца символы не совпадают ни для какого i. Например, слово “computer” является антипалиндромом, а слово “information” — нет (‘m’ на 6-й позиции и от начала и от конца).

Назовем число антипалиндромным, если его десятичная запись является антипалиндромным.

По заданному числу x найдите минимальное антипалиндромное число, строго большее x.

Формат входного файла

Входной файл содержит несколько тестов. Каждый тест состоит из единственного числа x ($1 \le x \le 10^{100}$), записанного на отдельной строке. После последнего теста идет число 0, его не следует обрабатывать.

Формат выходного файла

Для каждого числа во входном файле выведите в отдельную строку минимальное большее его антипалиндромное число.

Примеры

anti.in	anti.out
5 20 99 0	10 21 1010

Задача B. BibTEX

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

BibTeX — это средство для форматирования списка литературы в системе верстки TeX. BibTeX использует текстовый формат, в котором не хранится информация о форматировании, для каждого источника хранится информация об авторе, месте и времени публикации.

В этой задаче вы должны будете реализовать часть функциональности системы BibTeX, чтобы создать список литературы, в который входят книги и статьи.

Каждый источник в списке литературы начинается с тега "`@reference type`", за которым следует '`{`', затем список полей отформатированных как "`name = "value"`", разделенных запятыми, затем '`}`'.

Ваша программа должна поддерживать следующие типы элементов:

- **article** — Статья в журнале.

  Обязательные поля: author, title, journal, year

  Необязательные поля: volume, number, pages

- **book** — Книга.

  Обязательные поля: author, title, publisher, year

  Необязательные поля: volume

Поля должны быть отформатированы следующим образом:

Поле	Описание форматирования
author	Автор источника. Каждый автор форматируется как "`Имя1 Имя2 ... Фамилия`". У каждого автора не менее 1 и не более 10 имен. Если у источника несколько авторов, то они разделяются предлогом "`and`". Имя и фамилия автора не может быть "`and`". Общая длина этого поля не превышает 200 символов. Имена и фамилии авторов содержат только буквы латинского алфавита и разделяются пробелами.
title	Название источника. Строка не длиннее 200 символов, содержащая буквы латинского алфавита, пробелы, цифры и знаки препинания.
journal, publisher	Название журнала и издание. Строки не длиннее 200 символов, содержащие буквы латинского алфавита, пробелы, цифры и знаки препинания.
year	Год издания, целое число от 1500 до 2008.
volume	Номер тома, целое число от 1 до 10^6.
number	Номер журнала, целое число от 1 до 10^6.
pages	Страницы в журнале, отформатированные как "`from--to`" или "`page`". "From", "to" и "page" — целые числа от 1 до 10^6, "from" < "to".

Авторы каждого источника сортируются по фамилии, затем по первому имени, второму имени, и т. д. После этого источники сортируются по первому автору (аналогично, происходит сравнение сначала по фамилии, затем по первому имени, второму имени, и т. д., если у одного из авторов соответствующее имя есть, а у другого — нет, то вместо отсутствующего имени подставляется пустая строка), затем по второму автору, и т. д., затем по названию. Никакие два источника не имеют полностью совпадающие множества авторов и название, кроме нескольких томов одной книги. Тома одной книги сортируются по номеру тома.

Ссылки на статьи должны быть отформатированы следующим образом:

“`Authors Title // Journal[, Volume][ (Number)] -- year[ -- pages]`”.

Здесь в квадратных скобках “`[...]`” приведена часть, которая может отсутствовать. Авторы разделяются запятыми. Каждый автор форматируется как “`Фамилия И1. И2. ...`” где И1, И2, и т. д. — инициалы автора (первые буквы соответствующего имени). Страницы форматируются как “`p. page`” если источник занимает одну страницу, или “`pp. from--to`”, если источник занимает несколько страниц.

Ссылки на книги форматируются следующим образом:

“`Authors Title[, Vol. Volume] -- Publisher, Year`”.

Ссылки нумеруются, начиная с 1 и предваряются их номером в квадратных скобках.

Используйте приведенные примеры ввода и вывода в качестве образца.

Формат входного файла

Входной файл содержит список источников в формате BibTeX, во входном файле содержится не более 100 источников. Все элементы входного файла чувствительны к регистру символов. Все элементы разделяются пробелами и/или переводами строки.

Формат выходного файла

Выведите список литературы в описанном формате.

Пример

bibtex.in

```
@book
{
    author = "Donald Ervin Knuth
    title = "The Art of Computer Programming
    volume = "1
    publisher = "Addison-Wesley Professional
    year = "1997"
}

@book
{
    author = "Donald Ervin Knuth
    title = "The Art of Computer Programming
    volume = "2
    publisher = "Addison-Wesley Professional
    year = "1997"
}

@article
{
    author = "Robert Endre Tarjan and Andrew Goldberg
    title = "A new approach to the maximum flow problem
    journal = "Journal ACM
    volume = "35
    year = "1988
    pages = "921--940"
}
```

bibtex.out

```
[1] Goldberg A., Tarjan R. E. A new approach to the maximum flow problem // Journal ACM, 35 -- 1988 -- pp. 921--940
[2] Knuth D. E. The Art of Computer Programming, Vol. 1 -- Addison-Wesley Professional, 1997
[3] Knuth D. E. The Art of Computer Programming, Vol. 2 -- Addison-Wesley Professional, 1997
```

Задача C. Бурундучки

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

Аня и Андрей работают в медицинской лаборатории. Они тестируют экспериментальные лекарства на бурундучках. Исходно у них в лаборатории было два бурундучка с номерами 1 и 2, и это было очень удобно. Можно было сказать "бурундучки с суммарным номером 2" — и ясно, что речь идет о втором бурундучке. Или "бурундучки с суммарным номером 3", и ясно что речь идет об обоих бурундучках.

Вскоре был куплен третий бурундучок для экспериментов. В описаниях множеств бурундучков с использовнием суммарного номера теперь появилась неоднозначность: фраза "бурундучки с суммарным номером 3" теперь может означать как бурундучков 1 и 2, так и третьего бурундучка. Но Андрей и Аня быстро придумали решение: теперь они говорили "один бурундучок с суммарным номером 3" или "два бурундучка с суммарным номером 3" и неоднозначность разрешалась.

Но с покупкой четвертого бурундучка проблемы усугубились. Если присвоить ему номер 4, то фраза "два бурундучка с суммарным номером 5" становится неоднозначной. Но разумеется, и для этой проблемы решение было найдено. Бурундучку был присвоен номер 5, и фраза "k бурундучков с суммарным номером s" по прежнему однозначно характеризовала множество бурундучков для любых k и s.

Помогите Ане и Андрею продолжить присваивать номера бурундучкам по мере того, как больше и больше зверюшек покупается для экспериментов. Пусть первые $n-1$ бурундучков были пронумерованы «жадным алгоритмом» — каждому был присвоен минимальный номер, при котором фраза "k бурундучков с суммарным номером s" однозначно характеризует множество бурундучков для любых k и s. Какой номер следует присвоить n-му бурундучку, в соответствии с жадным алгоритмом?

Формат входного файла

Входной файл содержит число n ($1 \le n \le 20$).

Формат выходного файла

Выведите одно число — номер, которые следует присвоить n-му бурундучку.

Примеры

chipmunks.in	chipmunks.out
1	1
4	5
5	8

Задача D. Производство деталей

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

В связи с наступившим кризисом, а также в целях повышения моральной стойкости коллектива, финансирование секретного военного завода «Фабрика мягких игрушек №37» было урезано в несколько раз. В сложившейся ситуации самое важное — не дать рабочим уйти с предприятия до тех пор, пока вопрос о возобновлении финансирования в прежнем объеме не будет улажен.

К счастью, пока информация еще не успела распространиться, руководство придумало хитроумный план по поддержанию поставок на прежнем уровне. По результатам последних исследований, после изготовления каждой секретной детали оставалось некоторое количество ценного секретного материала — мифрила. Мифрил для изготовления одной детали упакован в специальную капсулу. Оказалось, что после изготовления детали немного мифрила остается на стенках. Силами сотрудников завода было построено устройство, которое из m использованных капсул производит k полных. При этом полученные таким образом капсулы совершенно идентичны исходным.

Теперь руководство завода хочет узнать, сколько деталей можно изготовить из каждой из n партий мифрила, которые остались на складе. Вам поручается исследовать данный вопрос.

Формат входного файла

В первой строке входного файла содержатся целые числа n, m, k ($1 \le n \le 10^3$, $1 \le k < m \le 10^6$). Во второй строке входного файла содержится n целых чисел a_i — число полных капсул в i-й партии мифрила ($1 \le a_i \le 10^6$).

Формат выходного файла

Для каждого числа a_i выведите количество деталей, которое можно произвести из i-й партии.

Примеры

factory.in	factory.out
2 6 1 36 30	43 35
1 5 2 11	17

Во втором примере оптимальная последовательность действий выглядит следующим образом: из десяти полных капсул сделаем десять деталей, преобразовав десять пустых капсул в четыре полные. Далее из пяти полных капсул сделаем пять деталей, преобразовав пять пустых капсул в две полные, из которых сделаем еще две детали. Таким образом получим 17 деталей и две пустые капсулы, которые использовать не сможем.

Задача E. Футбол

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

Олег — большой любитель футбола и статистики. Недавно он нашел результаты участия его любимой команды в каком-то давнем чемпионате. К сожалению, единственной сохранившейся информацией оказалось то, сколько матчей было сыграно и сколько очков набрала команда. Напоминаем, что если матч завершается победой команды, то ей присуждается три очка, ничьей — одно очко, и если команда проигрывает матч, то она не получает ни одного очка.

Олегу стало интересно, сколько различных вариантов прохождения чемпионата было у его любимой команды. Различными считаются варианты, если результат хотя бы одного матча различен, причем счет не принимается во внимание, а учитывается только то, завершился матч победой, ничьей или проигрышем.

Формат входного файла

Первая строка входного файла содержит два целых числа: n ($0 \le n \le 36$) — количество набранных командой очков, и k ($1 \le k \le 12$) — количество матчей, сыгранных этой командой в чемпионате.

Формат выходного файла

В выходной файл выведите число различных вариантов прохождения чемпионата любимой команды Олега.

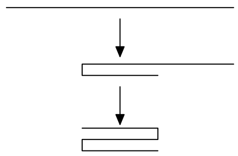
Примеры

football.in	football.out
3 2	2
4 3	6

Задача F. Бумага

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

Вова — большой любитель оригами. Особенно он любит складывать самолетики. Недавно он прочитал в книге описание нового самолета. Одно из действий, которое нужно совершить при изготовлении этого самолетика, — из клетчатого листка бумаги размером n на m получить «гармошку» размером 1 на 1 клетку. Каждое складывание состоит в том, что Вова сгибает листок бумаги, вдоль одной из линий разлиновки. Например, если $n = 3$, а $m = 1$, то возможна следующая последовательность складываний (вид сбоку):



Вова хочет сложить самолетик как можно быстрее, поэтому ему интересно за какое минимальное количество складываний он может получить «гармошку» размером в одну клетку?

Формат входного файла

В первой строке входного файла записаны размеры исходного листка в клетках — два натуральных числа n и m ($1 \le n, m \le 10^3$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

paper.in	paper.out
2 2	2
5 3	5

Задача G. Среднее количество точек

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

На плоскости расположено n точек. Никакие четыре точки не лежат на одной окружности, никакие три точки не лежат на одной прямой.

Случайным образом равновероятно выбираются три точки и через них проводится окружность. Найдите среднее число точек в этой окружности.

Формат входного файла

Первая строка входного файла содержит число n — количество точек ($3 \le n \le 1500$). Следующие n строк содержат по два целых числа — координаты точек. Координаты не превышают по модулю 10^6.

Формат выходного файла

Выведите одно вещественное число — среднее количество точек внутри окружности, проведенной через три случайных точки. Ваш ответ должен отличаться от правильного не больше чем на 10^{-9}.

Примеры

points.in	points.out
4 1 1 10 10 10 0 0 10	0.5
4 7 7 10 10 10 0 0 10	0.25

Задача H. Тетрис

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

Вася изобрел новую модификацию тетриса. Основное нововведение этой версии состоит в том, что целиком заполненные строки исчезают только в конце игры, при подсчете очков. При этом за n подряд идущих заполненных строк начисляются очки в количестве $\frac{n(n+1)}{2}$.

Как своего лучшего друга, Вася попросил Вас реализовать функцию подсчета очков.

Формат входного файла

В первой строке входного файла содержатся целые числа n и m ($1 \leq n, m \leq 100$). Далее следуют n строк из m символов с описанием игрового поля, причем на j-й позиции i-й строки содержится «*» (звездочка), если клетка занята, и «.» (точка) иначе.

Формат выходного файла

В выходной файл выведите количество начисленных очков.

Примеры

tetris.in	tetris.out
3 2 ** .. **	2
3 2 ** ** **	6

Задача I. Кладоискатель

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

Юный кладоискатель Рома прошел курс обучения по специальности «кладовое дело», и теперь проходит летнюю практику. Летняя практика проходит близ поселка «Каменные Зори» и длится ровно b дней. Каждый день Рома находит a закопанных в окрестности монет. Таким образом, в конце первого дня у него было a монет, в конце второго — $2 \cdot a$, а по окончании практики у Ромы должно накопиться $b \cdot a$ монет.

Если в конце дня ответственный преподаватель замечал, что число Роминых монет делится на b, то Роме разрешалось взять с полки пирожок, который он тут же съедал. Помогите Роме посчитать, сколько пирожков он съест за время прохождения практики.

Формат входного файла

Первая строка входного файла содержит два целых числа a и b ($1 \le a, b \le 10^9$).

Формат выходного файла

В выходной файл выведите число съеденных Ромой пирожков.

Примеры

treasure.in	treasure.out
2 1	1
2 2	2
5 4	1
10 5	5