

А. Функция

Ограничение по времени: **1000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Нужно определить, пересекается ли с осями график функции $f(x) = ax^2 + bx + c$.

Входные данные

Три целых числа отделённые друг от друга пробелом: a, b, c ($-10^8 \leq a, b, c \leq 10^8$).

Выходные данные

Выведите X, если график пересекается с осью OX, выведите Y, если пересекается с осью OY, если пересекается с обеими осями, то выведите XY. Если не пересекается ни с одной из осей, то выведите слово NONE.

Примеры

Ввод	Вывод
0 0 0	XY
1 1 1	Y
-1 2 -1	XY

В. Сила

Ограничение по времени: **1000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Имеется множество точек на плоскости заданных своими координатами (x_i, y_i) . Между двумя точками существует сила притяжения направленная внутрь отрезка соединяющего эти точки. Величина силы обратно пропорциональна квадрату расстояния между точками. Суммарная сила действующая на точку равна сумме векторов сил действующих на точку.

Входные данные

Первая строка содержит единственное число N ($2 \leq N \leq 100$) – количество заданных точек. Далее идёт N пар чисел x_i, y_i ($-10^4 \leq x_i, y_i \leq 10^4$). Каждая пара чисел на отдельной строке. Все точки находятся в разных местах.

Выходные данные

Выведите номер точки, на которую действует максимальная по величине суммарная сила. Если имеется несколько точек, на которые действует максимальная сила, то выведите номер любой из них.

Пример

Ввод	Вывод
3 0 0 1 1 3 3	1

С. Маска

Ограничение по времени: **1000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Имеются растровые чёрно-белые изображения размером 4 на 4 пикселя. Каждый пиксель изображения может быть либо чёрным (обозначен нулём), либо белым (обозначен единицей). Требуется для всего данного набора изображений определить минимальное достаточное количество пикселей, которые однозначно определяют изображение.

Входные данные

В первой строке указано число N ($2 \leq N \leq 10^3$) – количество изображений. Далее идёт N описаний изображений, по одному на каждой строке. Описание изображения дано в виде 4-х шестнадцатеричных цифр, где каждый бит двоичной записи этих цифр и даёт информацию о цвете соответствующего пикселя.

Выходные данные

Выведите одну строку из 4-х шестнадцатеричных цифр — маску, где в двоичной записи цифр 1 означает, что соответствующий пиксель необходимо учитывать при определении изображения.

Примеры

Ввод	Вывод
2 aalb aеlb	0400
3 1234 1235 1236	0003
2 ffff ffff	0000

D. Отображение

Ограничение по времени: **1000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Для просмотра изображений заданных в задаче «Маска» требуется написать программу визуализации, то есть по описанию вывести картинку — каждый пиксель отобразить соответствующим знаком (определён во входных данных), какой бит какой цифры соответствует какому пикселю определены во входных данных.

Входные данные

Первая строка содержит описание изображение в соответствии с описанием в задаче «Маска». Далее на второй строке имеются два различных символа из множества возможных (0-9, a-z, A-Z, @, #, _, =, +), запятые, круглые скобки и дефис (минус) не входят в это множество, дефис указывает на диапазон символов. Первый символ соответствует пикселю в значении 0 (нуль), второй — 1 (единица).

Далее в 4-х строках идёт описание визуализации в смысле определения откуда брать значение пикселя (из какого бита). Пиксели перечислены слева направо и сверху вниз, а описание бит — два значения, где первое указывает номер шестнадцатеричной цифры (от 1 до 4), второе — номер бита (от 0 до 3), где 3 соответствует старшему разряду, 0 — младшему.

Гарантируется, что ни одно описание пикселя не повторяется, то есть одному пикселю соответствует один бит описания.

Выходные данные

Выведите изображение в соответствии с заданием.

Примеры

Ввод	Вывод
aalb _# 13121110 23222120 33323130 43424140	#_#_ #_#_ _# #_##
aalb #_ 40414243 30313233 20212223 10111213	_#_ _### #_#_ #_#_
aalb 01 40414243 30313233 20212223 10111213	1101 1000 0101 0101

Е. Импульсный набор номера

Ограничение по времени: **1000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Импульсный набор номера применялся для передачи его (номера) автоматической телефонной станции. При этом цифры передаются путем прерывания электрического тока в линии связи.

Число размыканий линии соответствует набираемой цифре, для нуля производится 10 размыканий. Между цифрами дополнительно выдерживается междигитровая пауза.

Входные данные

В единственной строке дан номер телефона. Он содержит от 2 до 20 цифр, может также между символами иметь символ дефиса, пробела или круглой скобки, эти символы используются для удобства чтения человеком и не влияют на импульсы. Гарантируется, что всего в строке не более 50 символов.

Выходные данные

Выведите картинку с импульсной последовательностью. Импульс отобразите следующей последовательностью `_[]_`, серия импульсов из 2-х импульсов `_[]_[]_` и так далее, межцифровую паузу обозначьте `___` (3 подчёркивания между `[]`). Для большего понимания смотрите примеры.

`[]` - открывающая и закрывающая квадратные скобки.

Пример

Ввод	Вывод
123	<code>_[]_[]_[]_[]_[]_[]_</code>
2 (1) 3-12	<code>_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_</code>
03	<code>_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_[]_</code>

Г. Произведение на отрезке

Ограничение по времени: **2000 миллисекунд**

Ограничение по памяти: **65000 кибибайт**

Дана последовательность из n целых чисел, каждое из которых находится в диапазоне от 0 до 9 включительно. Позиции в последовательности пронумерованы от 1 до n .

Требуется ответить на q запросов. Каждый запрос задаётся двумя индексами a и b . Найдите произведение чисел на позициях от a до b включительно и выведите остаток от деления этого произведения на 1 000 000 007. Последовательность во время обработки запросов не изменяется.

Входные данные

В первой строке записаны два целых числа n и q ($1 \leq n \leq 100\,000$, $1 \leq q \leq 100\,000$). Во второй строке записаны n целых чисел от 0 до 9 включительно, разделённых пробелами.

В каждой из следующих q строк записаны два целых числа a и b ($1 \leq a \leq b \leq n$), задающие очередной запрос.

Выходные данные

Для каждого запроса выведите в отдельной строке одно целое число от 0 до 1 000 000 006 включительно — остаток от деления произведения на 1 000 000 007. Ответы выводите в порядке запросов.

Пример

Ввод	Вывод
3 3 2 3 4 1 2 2 3 1 3	6 12 24
3 3 2 0 3 1 3 1 1 3 3	0 2 3