

Задача А. Интересная характеристика

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

Глеб — креативный студент! Он только что изучил битовые операции и с их помощью придумал разные характеристики для чисел. *Интересной характеристикой* целого неотрицательного числа A он называет минимальное целое неотрицательное число B , для которого верно равенство $A \oplus B = A \& B$, где \oplus - операция побитового исключающего «ИЛИ», $\&$ - операция побитового «И». О том, как работают данные битовые операции, Вы можете узнать из статьи на Википедии: <https://clck.ru/EUMdQ>.

Глеб написал элегантную программу, которая очень быстро вычисляет интересную характеристику даже для очень больших чисел. Если Вы сможете написать свою версию программы, то сразу же заслужите уважение студента.

Формат входных данных

В первой строке задано единственное целое неотрицательное число A ($0 \leq A \leq 2 \cdot 10^9$).

Формат выходных данных

Выведите в одной строке единственное целое неотрицательное число — интересную характеристику числа A . Если же такой не существует, выведите -1 .

Система оценки

Задача содержит три подзадачи.

Подзадача 1 (50 баллов)

Для всех тестов данной подзадачи $0 \leq A \leq 100$.

Подзадача 2 (30 баллов)

Для всех тестов данной подзадачи $0 \leq A \leq 10^9$.

Подзадача 3 (20 баллов)

Для всех тестов данной подзадачи $0 \leq A \leq 2 \cdot 10^9$.

Баллы за подзадачу начисляются только в случае успешного прохождения всех тестов данной подзадачи и всех тестов более ранних подзадач.

Примеры

стандартный ввод	стандартный вывод
0	0
3	-1

Задача В. Ошибка министерства

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

У Пупы и Лупы есть доминошки двух типов:



И как-то раз Пупа и Лупа получили новую задачу — покрыть доминошками прямоугольником, то есть найти такой прямоугольник, в котором можно расположить все доминошки без наложения друг на друга.

Так как таких прямоугольников существует бесконечное количество, то ребятам надо выбрать какой-то определённый из всех, но в министерстве всё перепутали и раздали разные задачи: Пупа должен минимизировать площадь прямоугольника, а Лупа — периметр.

Для Пупы задача очевидна — расположить доминошки в ряд и получить прямоугольник, у которого площадь равна суммарной площади доминошек.

Но вот Лупа не может решить свою задачу так же быстро, поэтому просит помощи у вас.

Формат входных данных

Входные данные состоят из двух чисел n и m — количества доминошек у Лупы первого и второго типов соответственно ($0 \leq n, m \leq 10^9$).

Формат выходных данных

Выведите одно число — минимальный периметр прямоугольника, который покроет все доминошки Лупы.

Система оценки

Задача содержит три подзадачи.

Подзадача 1 (30 баллов)

Для всех тестов данной подзадачи $0 \leq n, m \leq 100$.

Подзадача 2 (30 баллов)

Для всех тестов данной подзадачи $0 \leq n, m \leq 10^5$.

Подзадача 3 (40 баллов)

Для всех тестов данной подзадачи $0 \leq n, m \leq 10^9$.

Баллы за подзадачу начисляются только в случае успешного прохождения всех тестов данной подзадачи и всех тестов более ранних подзадач.

Примеры

стандартный ввод	стандартный вывод
1 1	8
0 1	6
1 0	4

Задача С. Погрызенный массив

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

Хомяк Гриша, как и все другие хомяки, обожает грызть различные предметы. Особое удовольствие ему доставляет отгрызать кусочки от массивов.

У Гриши есть массив длины n . Каждый день он совершает одно из двух действий:

- Отгрызть a элементов с начала массива и b элементов с конца;
- Отгрызть b элементов с начала массива и a с конца.

Если в какой-то день в массиве остается не более $a + b$ элементов, то Гриша сгрязает их все, после чего у него остается пустой массив, точить зубы об который более не интересно.

Ваша задача — посчитать, сколько различных **непустых** подотрезков массива могло остаться у Гриши после того, как он сколько-то дней (возможно, ноль) грыз массив.

Подотрезком массива $[l..r]$ считается последовательность индексов вида

$$l, l+1, l+2, \dots, r-2, r-1, r.$$

Формат входных данных

В единственной строке заданы три числа n, a, b ($1 \leq n \leq 10^9, 1 \leq a, b \leq n$) — размер массива и параметры Гришиных зубов.

Формат выходных данных

Выведите единственное число — количество **непустых** подмассивов, которое могло остаться после некоторого числа (возможно, нулевого) укусов.

Система оценки

Задача содержит четыре подзадачи. Баллы за каждую из подзадач начисляются только при условии прохождения **всех** тестов данной подзадачи и **всех** тестов **всех** предыдущих подзадач.

Подзадача 1 (20 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10$.

Подзадача 2 (20 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10^3$.

Подзадача 3 (20 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10^5$.

Подзадача 4 (40 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10^9$.

Примеры

стандартный ввод	стандартный вывод
5 1 2	3
7 2 4	3
4 3 3	1

Замечание

В первом примере могли остаться следующие подмассивы:

- [1..5] (до первого дня)

- [3..4] (после первого дня, если он сгрызет 2 элемента из начала и 1 из конца)
- [2..3] (после первого дня, если он сгрызет 1 элемент из начала и 2 из конца)

Несложно показать, что после двух укусов массив становится пустым.

Задача D. Скоростная печать

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

Женя недавно отлично выступил на региональном туре всероссийской олимпиады школьников по информатике и уверенно квалифицировался на заключительный тур. Разумеется, на всерос не стоит ехать неподготовленным, а потому Женя активно вспоминает хитрые алгоритмы и упражняется в скорости печати.

Известно, что скорость печати Жени настолько высока, что за одну миллисекунду он способен нажать ровно одну произвольную клавишу на клавиатуре (то есть либо добавить какой-то символ в конец к напечатанному на экране тексту, либо нажать клавишу Backspace и стереть последний напечатанный символ).

Женя частенько сбивается с ритма, а потому просит вас его проконтролировать. Более формально, вам дано q запросов, каждый из которых звучит следующим образом: правда ли, что Женя мог превратить заданную строку s в заданную строку t ровно за k миллисекунд?

Формат входных данных

В первой строке задано число q ($1 \leq q \leq 10^5$), означающее число запросов.

В каждой из следующих строк через пробел заданы число k_i и две строки s_i и t_i ($1 \leq k_i \leq 10^9$, $1 \leq |s_i|, |t_i| \leq 10^6$), состоящие из маленьких латинских букв. Здесь под $|s|$ подразумевается длина строки s .

Гарантируется, что $\sum_{i=1}^q (|s_i| + |t_i|) \leq 10^6$.

Формат выходных данных

В ответ на каждый запрос выведите «Yes», если можно превратить строку s_i в строку t_i за ровно k действий. В противном случае выведите «No».

Система оценки

Задача содержит две подзадачи. Баллы за каждую из подзадач начисляются только при условии прохождения **всех** тестов данной подзадачи и **всех** тестов **всех** предыдущих подзадач.

Подзадача 1 (40 баллов)

Для всех тестов данной подзадачи $1 \leq q \leq 100$, $\sum_{i=1}^q (|s_i| + |t_i|) \leq 10^3$.

Подзадача 2 (60 баллов)

Для всех тестов данной подзадачи $1 \leq q \leq 10^5$, $\sum_{i=1}^q (|s_i| + |t_i|) \leq 10^6$.

Пример

стандартный ввод	стандартный вывод
5	Yes
1 ab aba	No
4 ab abbab	Yes
3 ab acd	Yes
5 ab a	Yes
10 aaaaa bbbbb	

Замечание

В первом запросе возможная последовательность операций выглядит следующим образом:
 $ab \rightarrow aba$.

В четвертом запросе возможная последовательность операций выглядит следующим образом:
 $ab \rightarrow abd \rightarrow abdg \rightarrow abd \rightarrow ab \rightarrow a$.

Задача Е. Длинная дорога

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

Василий вырвался отдохнуть в страну под названием Руритания. В Руритании есть n городов, и лишь в одном из них (первом) находится аэропорт, в который скоро прибывает Василий. Как известно Василию, между некоторыми парами городов ходят автобусы (автобус не посещает другие города, кроме начального и конечного), стоимость проезда в которых равна одному руританскому доллару. Если автобус ходит от одного города к другому, то он же ходит от другого города к первому; кроме того, ни один автобус не ходит между городом и им же самим — это глупо.

В руках у Василия имеется книга о достопримечательностях Руритании, в котором очень красочно расписано про город под номером n , и Василий очень хочет попасть туда. Кроме того, у Василия есть справочник путешественника, в котором для любой пары руританских городов указана сумма, потратив которую, можно добраться из одного города до другого, не посещая какой-либо город дважды и не пользуясь дважды одним и тем же маршрутом (однако начало и конец пути могут совпадать). Однако справочник уже устарел, поэтому указанные суммы могут быть неоптимальны.

Поскольку Василий планирует в дальнейшем не терять ни минуты, то он хочет на основании этой информации знать, как он сможет добраться до пункта назначения, однако Василий не знает точно, между какими городами ходят автобусы. Помогите ему найти маршрут.

Формат входных данных

В первой строке входных данных задано число n ($2 \leq n \leq 1000$) - количество городов в Руритании. Далее следуют n строк по n чисел. i -ое число в j -ой строке обозначает, что от i -ого города до j -ого можно добраться за указанное количество руританских долларов. В случае, если $i \neq j$, то указанный путь не посещает ни один из городов дважды и не использует дважды один и тот же маршрут. Если же $i = j$, то указанный цикл не посещает ни один из городов дважды и не использует дважды один и тот же маршрут, за исключением того, что начальный город равен конечному.

Стоимость поездки от i -ого города до j -ого не обязана быть оптимальной. Стоимость поездки от i -ого города до j -ого может не равняться стоимости поездки от j -ого города до i -того. Стоимость поездки между городом и им самим может не равняться нулю.

Гарантируется, что существует такой набор маршрутов, для которого найдутся все указанные пути.

Формат выходных данных

Если Василий не сможет узнать заранее маршрут, с помощью которого он сможет гарантированно добраться из первого города до последнего, выведите -1 . Иначе выведите две строки:

- В первой строке необходимо вывести ровно одно число c - стоимость найденной вами поездки.
- В следующей строке необходимо вывести $c + 1$ число a_1, a_2, \dots, a_{c+1} такие, что для любого $1 \leq i \leq c$ между городами с номерами a_i и a_{i+1} обязательно ходит автобус. Должно выполняться: $a_1 = 1$, $a_{c+1} = n$.

Если ответов несколько, выведите любой. Минимизировать стоимость при этом необязательно.

Система оценки

Задача содержит три подзадачи. Баллы за каждую из подзадач начисляются только при условии прохождения **всех** тестов данной подзадачи и **всех** тестов **всех** предыдущих подзадач.

Подзадача 1 (20 баллов)

Для всех тестов данной подзадачи $n \leq 3$.

Подзадача 2 (40 баллов)

Для всех тестов данной подзадачи $n \leq 20$.

Подзадача 3 (40 баллов)

Для всех тестов данной подзадачи $n \leq 1000$.

Примеры

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

Замечание

В первом тесте правильным ответом будет выбор пути 1 – 2 – 3. Обратите внимание, что ответ 1 – 3 некорректен, так как между первым и третьим городом необязательно ходит автобус.

Во втором тесте существует возможность прокатиться по кругу из первого города по трём городам. В таком случае у нас существуют все возможные маршруты между парами городов, иначе это было бы невозможно. В частности, у нас есть маршрут 1 – 3.

Задача F. Конфеты за ответы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6.5 секунд
Ограничение по памяти:	168 мегабайт

У Андрея есть массив, состоящий из n натуральных чисел, пронумерованных от 1 до n .

За один ход Андрей может взять любое число a_i из массива и разделить его на произвольное простое число p (при условии, что $a_i \bmod p = 0$). Обозначим за $f(i, x)$ минимальное количество ходов, которое требуется, чтобы **число делителей** a_i стало меньше или равным x .

Катя, подруга Андрея, периодически спрашивает, чему равняется $\sum_{i=l}^r f(i, x)$ для некоторых (l, r, x) , и за правильные ответы кормит его вкусными конфетами.

Помогите Андрею ответить на все Катины вопросы.

Формат входных данных

В первой строке задано число n — размер массива ($1 \leq n \leq 2 \cdot 10^5$).

В следующей строке через пробел следуют n чисел a_i — элементы массива ($1 \leq a_i \leq 10^6$).

Далее идет число q ($1 \leq q \leq 10^6$), задающее количество Катиных вопросов.

Каждая из следующих q строк содержит по три числа l_i, r_i, x_i ($1 \leq l_i \leq r_i \leq n, 1 \leq x_i \leq 10^9$), означающие границы и верхнюю границу числа делителей для i -го вопроса.

Формат выходных данных

Выведите q строк — ответы на все вопросы. Обратите внимание, что все операции производятся мысленно и не изменяют содержимое массива.

Система оценки

Задача содержит пять подзадач.

Баллы за первые три подзадачи начисляются как обычно — при прохождении всех тестов подзадачи и всех тестов всех предыдущих подзадач. Баллы за каждый тест четвертой и пятой группы начисляются независимо. Обратите внимание, что баллы за последнюю подгруппу будут известны только после окончания тура.

Подзадача 1 (15 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10, 1 \leq a_i \leq 10^2, 1 \leq q \leq 10$.

Подзадача 2 (15 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10^2, 1 \leq a_i \leq 10^3, 1 \leq q \leq 10^3$.

Подзадача 3 (20 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 10^4, 1 \leq a_i \leq 10^4, 1 \leq q \leq 10^4$.

Подзадача 4 (40 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 2 \cdot 10^5, 1 \leq a_i \leq 10^6, 1 \leq q \leq 10^6$.

Подзадача 5 (10 баллов)

Для всех тестов данной подзадачи $1 \leq n \leq 2 \cdot 10^5, 1 \leq a_i \leq 10^6, 1 \leq q \leq 10^6$.

Пример

стандартный ввод	стандартный вывод
5	0
1 2 3 4 6	6
8	0
1 5 4	2
1 5 1	1
2 3 2	2
2 3 1	4
4 5 3	0
4 5 2	
4 5 1	
4 5 4	

Замечание

Рассмотрим первый пример.

1. Ответ на **первый** вопрос равен 0, так как количество делителей каждого элемента изначально не превосходит 4.
 - Число 1 не нужно делить, потому что у него изначально один делитель.
 - Числа 2 и 3 состоят из одного простого числа, поэтому их нужно единожды поделить на 2 и 3 соответственно.
 - Числа 4 и 6 состоят из произведения двух простых чисел, поэтому каждое из них нужно делить дважды.
2. Ответ на **второй** вопрос равен 6.
3. Ответ на **пятый** вопрос равен 1, потому что количество делителей числа 6 равняется 4 (к примеру, можно поделить 6 на 2, в результате чего количество делителей уменьшится до 2).
4. Ответ на **шестой** вопрос равен 2, потому что в отличие от предыдущего вопроса количество делителей числа 4 больше, чем нужно. Это означает, что делить нужно и 4, и 6.

Задача G. Пушка и гора

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

Искандер решил пострелять из пушки рядом с горой, имеющей форму треугольника. Пушка располагается в точке $(0, 0)$.

Туманно, а потому часть горы скрыта от глаз. К счастью, оба склона горы частично видны. Сможет ли Искандер попасть в мишень с координатами $(x_0, 0)$, не находящуюся на склоне или внутри горы, если пушка задает начальную скорость снаряда, равную v , и при этом она недостаточно мощна, чтобы пробить гору насеквоздь? Ускорение свободного падения считайте равным 10.

Формат входных данных

В первой строке заданы два целых числа v и x_0 ($0 \leq v \leq 10000$, $-50000 \leq x_0 \leq 50000$) — начальная скорость снаряда и координата мишени.

Во второй строке через пробел записаны две пары целых чисел (x_1, y_1) и (x_2, y_2) , обозначающие координаты концов отрезка, соответствующего видимой части левого склона. В третьей строке записаны две пары целых чисел (x_3, y_3) и (x_4, y_4) соответствующие правому склону.

Гарантируется, что отрезки корректно задают гору, находящуюся правее пушки.

Формат выходных данных

Если из пушки можно поразить цель, выведите «Yes» (без кавычек).

В противном случае выведите «No» (без кавычек).

Система оценки

Задача содержит три подзадачи.

Баллы за каждую из подзадач начисляются при условии прохождения всех тестов данной подзадачи и всех тестов предыдущих подзадач.

Подзадача 1 (15 баллов)

В данной подзадаче горы нет, то есть для всех тестов данной подзадачи: $x_1 = x_2 = x_3 = x_4 = 0$ и $y_1 = y_2 = y_3 = y_4 = 0$.

Подзадача 2 (35 баллов)

В данной подзадаче гору видно полностью, то есть для всех тестов данной подзадачи: $0 \leq x_1 < x_2 = x_3 < x_4 \leq 10000$ и $y_1 = 0, 0 < y_2 = y_3 \leq 10000, y_4 = 0$.

Подзадача 3 (50 баллов)

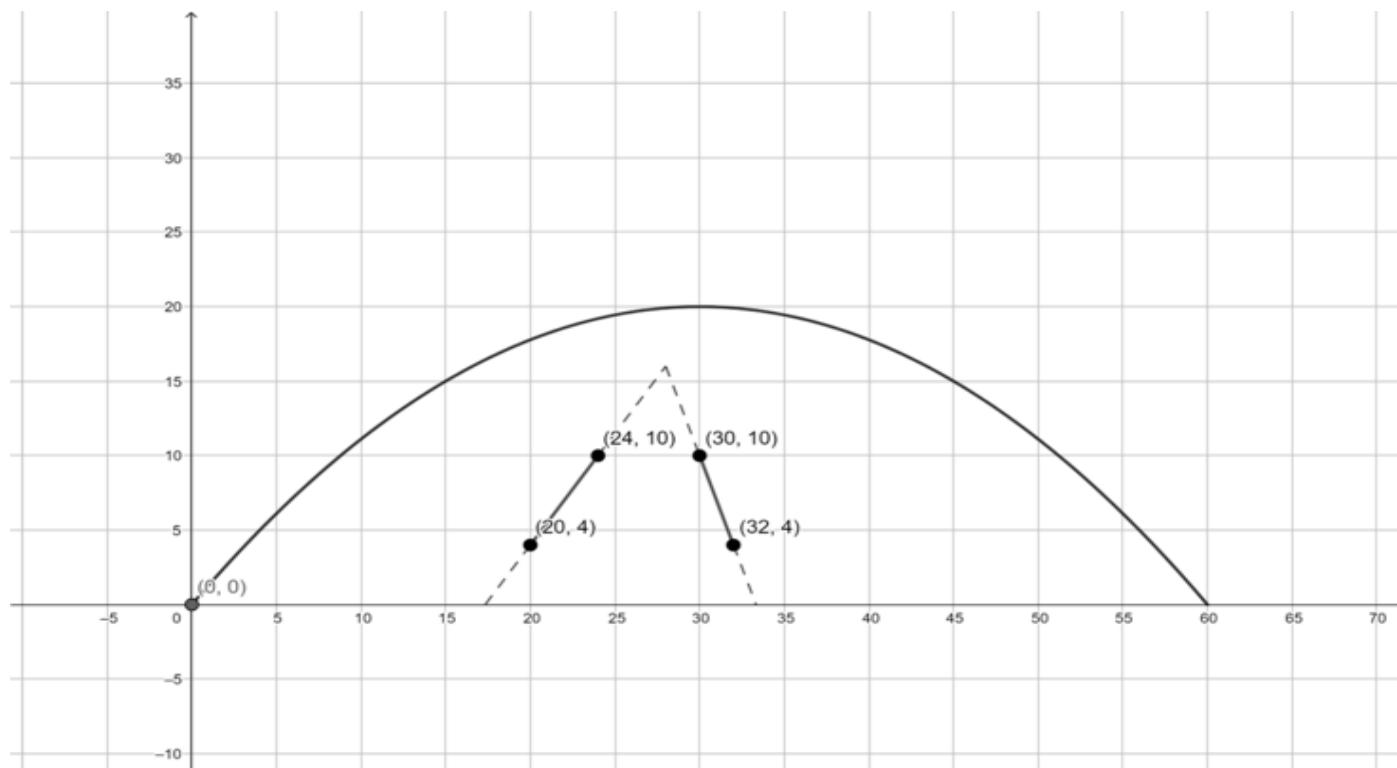
Для всех тестов данной подзадачи: $0 \leq x_1 < x_2 < x_3 < x_4 \leq 10000$ и $0 \leq y_1 < y_2 \leq 10000$, $0 \leq y_4 < y_3 \leq 10000$.

Пример

стандартный ввод	стандартный вывод
25 60 20 4 24 10 30 10 32 4	YES

Замечание

Траектория полета снаряда в первом teste:



Задача Н. Знаки

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

Гриша очень любит арифметические операции. Его личный топ-3 — сложение, умножение и взятие по модулю 10.

Также у Гриши есть массив, с которым он очень любит проводить непонятные действия. Его намерение в этот раз заключается в том, чтобы расставить между соседними элементами массива знаки + и ·, вычислить значение получившегося выражения и взять его по модулю 10.

Гриша не может определиться, как именно расставить знаки, ведь вариантов очень много. Поэтому для каждого остатка по модулю 10 он просит вас посчитать количество способов расставить знаки так, чтобы при взятии по модулю в итоге получился именно этот остаток.

Формат входных данных

В первой строке задано число n — количество чисел в массиве ($2 \leq n \leq 10^5$).

Во второй строке через пробел следуют n чисел a_i ($0 \leq a_i \leq 10^9$) — элементы массива.

Формат выходных данных

Выведите 10 чисел. i -е из них (в 0-индексации) должно означать количество способов расставить знаки между элементами массива так, чтобы остаток по модулю 10 равнялся в точности i . Так как ответ может быть достаточно большим, **выводите его по модулю $10^9 + 7$.**

Система оценки

Задача содержит три подзадачи.

Баллы за каждую из подзадач начисляются при условии прохождения всех тестов данной подзадачи и всех тестов предыдущих подзадач.

Подзадача 1 (20 баллов)

Для всех тестов данной подзадачи $n \leq 15$.

Подзадача 2 (50 баллов)

Для всех тестов данной подзадачи $n \leq 10^3$.

Подзадача 3 (30 баллов)

Для всех тестов данной подзадачи $n \leq 10^5$.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	0 0 0 0 0 1 2 1 0 0
7 118 289 144 12 0 11 593	9 8 4 9 7 4 6 4 6 7

Замечание

В первом примере возможны 4 способа расстановки знаков:

- $(1 + 2 + 3) \% 10 = 6$
- $(1 \cdot 2 + 3) \% 10 = 5$
- $(1 + 2 \cdot 3) \% 10 = 7$
- $(1 \cdot 2 \cdot 3) \% 10 = 6$

Задача I. Подсказка к замку

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

У Жени и Вали есть сундук с кодовым замком. На замке сверху есть ряд из восьми окошек, в каждом окошке видна цифра. Можно поворачивать диск под каждым окошком и таким образом менять цифру в нём на любую десятичную цифру. Замок открывается, когда сверху на дисках получается определённый код из восьми цифр.

Женя ещё маленькая, ей сложно запомнить все восемь цифр кода. Через некоторое время она забывает цифры на каких-то четырёх позициях, причём никто не знает заранее, на каких. Ребята решили, что Женя будет писать себе подсказку. Чтобы подсказки могло хватить, но никто другой не мог с её помощью открыть замок, решено было делать подсказку из шести десятичных цифр.

Итак, с замком происходит следующее. Сначала Валя выбирает код и говорит его Жене. В этот момент Женя, зная все восемь цифр кода, составляет себе подсказку из шести цифр. Через некоторое время Женя забывает цифры на каких-то четырёх позициях. Она должна, посмотрев в свою подсказку, всё-таки открыть кодовый замок.

Придумайте, как именно можно составить подсказку, чтобы потом по ней и известным цифрам на любых четырёх позициях восстановить весь код.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. В каждом тесте может быть несколько кодов.

При первом запуске решение делает для кодов подсказки. В первой строке записано слово «**memo**». Вторая строка содержит целое число n — количество кодов ($1 \leq n \leq 1000$). Каждая из следующих n строк содержит ровно восемь десятичных цифр — очередной код. Решение должно вывести n строк, в каждой ровно по шесть десятичных цифр — подсказки к кодам в порядке их следования во входных данных.

При втором запуске решение открывает замки, для каждого зная часть кода и подсказку. В первой строке записано слово «**open**». Вторая строка содержит целое число n — количество замков ($1 \leq n \leq 1000$). Каждая из следующих n строк содержит часть кода и подсказку к нему, разделённые пробелом. Часть кода состоит ровно из восьми символов: на каких-то четырёх позициях стоят правильные цифры кода, а на оставшихся четырёх — вопросики («?»). Подсказка копируется из ответа решения на этот код и содержит ровно шесть десятичных цифр. Решение должно вывести n строк, в каждой ровно по восемь десятичных цифр — коды в порядке следования вопросов во входных данных.

В каждом тесте число n при втором запуске такое же, как и при первом. Кроме того, при втором запуске используются такие же коды, как и при первом, но вопросы о них могут следовать в произвольном порядке. Этот порядок, а также положение известных позиций в кодах, зафиксированы заранее в каждом тесте.

При всех запусках каждая строка входных данных, включая последнюю, завершается переводом строки.

Примеры

В каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. В примере мы рассмотрим решение, которое просто повторяет первые шесть цифр в качестве подсказки. А при открытии замков, если неизвестна какая-то из последних двух цифр, она считается равной нулю. К сожалению, такое решение не всегда правильно восстанавливает код.

Далее показаны два запуска этого решения на первом тесте.

стандартный ввод	стандартный вывод
memo 4 00000000 12345678 98765678 09099009	000000 123456 987656 090990
open 4 ????5678 987656 0000???? 000000 ?2?4??78 123456 ????9009 090990	98765678 00000000 12345678 09099009

Система оценки

Каждый тест в этой задаче оценивается отдельно.

Задача J. Числовая пирамида

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Числовой пирамидой размера n называется треугольная таблица, которая обладает следующими свойствами:

- В таблице n рядов.
- В i -м сверху ряду ровно i чисел.
- Каждый следующий ряд расположен так, чтобы числа предыдущего ряда находились над промежутками между соседними числами следующего.
- Каждое число, кроме чисел нижнего ряда, равно сумме двух соседних чисел в следующем ряду, над промежутком между которыми оно находится.

Мирон нарисовал на доске числовую пирамиду размера n из целых чисел. После этого Никифор подошёл к доске и молча стёр некоторые числа. Мирон возмутился, но Никифор в ответ сказал, что ничего непоправимого не произошло: не существует двух различных пирамид размера n , в которых оставшиеся числа были бы такими же, даже если разрешить использовать не целые числа, а вещественные.

Помогите Мирону восстановить числовую пирамиду.

Формат входных данных

В первой строке задано целое число n — размер пирамиды ($1 \leq n \leq 9$).

Далее следует n строк. В i -й из этих строк задан i -й ряд числового пирамиды: он задаётся как i целых чисел, разделённых пробелами. Стока может начинаться с одного или нескольких пробелов. Каждое число лежит в пределах от 1 до 9, если оно известно. Неизвестные числа обозначаются нулём.

Гарантируется, что существует единственная числовая пирамида из вещественных чисел, в которой известные числа совпадают с заданными. Также гарантируется, что в этой единственной числовой пирамиде все числа целые.

Формат выходных данных

Выведите n строк. В i -й из этих строк выведите i -й ряд числового пирамиды: он задаётся как i целых чисел, разделённых пробелами. В позициях, где числа были известны, должны стоять те же числа в том же порядке. Разрешается выводить пробелы в начале и в конце строки, а также несколько пробелов подряд.

Система оценки

Задача содержит три подзадачи. Баллы за каждую из подзадач начисляются только при условии прохождения всех тестов этой подзадачи и всех тестов всех предыдущих подзадач.

Подзадача 1 (10 баллов)

Во всех тестах этой подзадачи $1 \leq n \leq 2$.

Подзадача 2 (20 баллов)

Во всех тестах этой подзадачи $1 \leq n \leq 3$.

Подзадача 3 (70 баллов)

Во всех тестах этой подзадачи $1 \leq n \leq 9$.

Примеры

стандартный ввод	стандартный вывод
2 3 0 1	3 2 1
2 0 5 5	10 5 5

Задача К. Простая задача

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

На самом деле, задача простая не потому, что ее легко решить, а потому, что в ней присутствуют простые числа.

Впрочем, условие у нее тоже довольно-таки простое. От вас требуется всего-то уметь находить k -е простое число.

Формат входных данных

В единственной строке задано число k ($1 \leq k \leq 10^8$).

Формат выходных данных

Выведите k -е простое число. Гарантируется, что тесты таковы, что ответ не превосходит $2 \cdot 10^9$.

Примеры

стандартный ввод	стандартный вывод
1	2
2	3
3	5
10	29
7777	79357

Замечание

Последовательность простых чисел начинается так: 2, 3, 5, 7, 11,

Система оценки

Тесты к этой задаче состоят из семи подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		k		
0	0	—	—	Тесты из условия.
1	7	$k \leq 10^4$	0	—
2	9	$k \leq 10^5$	0, 1	—
3	11	$k \leq 5 \cdot 10^5$	0, 1, 2	—
4	12	$k \leq 10^6$	0, 1, 2, 3	—
5	15	$k \leq 2 \cdot 10^6$	0, 1, 2, 3, 4	—
6	15	$k \leq 10^7$	0, 1, 2, 3, 4, 5	—
7	31	$k \leq 10^8$	0, 1, 2, 3, 4, 5, 6	—