Задача A. Компьютеры для сборов

Имя входного файла:	`computers.in`
Имя выходного файла:	`computers.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Клубе программистов готовятся к сборам. Организаторы смогли найти N абсолютно идентичных по своей конфигурации компьютеров. Однако, вот незадача — многие из этих компьютеров не работают. Поскольку компьютеры абсолютно одинаковые, их компоненты можно переставлять из одного компьютера в другой. Таким образом, например, можно из двух нерабочих компьютеров, в одном из которых неисправен жесткий диск, а в другом — блок питания, собрать один рабочий компьютер.

Организаторы сборов собрали информацию о том, какие компоненты в каких компьютерах исправны, а какие — нет. Помогите им подсчитать, сколько рабочих компьютеров они могут получить.

Формат входного файла

Первая строка входного файла содержит два целых числа N и K — количество компьютеров, найденных организаторами и количество компонент в каждом из компьютеров ($1 \le N \le 1000, 1 \le K \le 10$). Далее следует N строк по K чисел в каждой. Каждая строка описывает один компьютер и содержит K чисел 0 или 1, которые описывают состояние соответствующих компонент компьютера: единица соответствует рабочему компоненту, ноль — неисправному.

Формат выходного файла

Выведите единственное число — максимальное количество рабочих компьютеров, которое смогут собрать организаторы сборов.

Примеры

computers.in	computers.out
3 4 1 1 1 1 1 0 1 0 0 1 0 1	2
3 4 1 0 1 1 1 1 0 0 0 1 1 1	2
3 4 1 0 0 1 0 1 0 0 0 0 1 0	1

Задача B. CSS

Имя входного файла:	`css.in`
Имя выходного файла:	`css.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Компания «Dautroom» производит ребрендинг. В ходе ребрендинга в компании был определён новый фирменный стиль, логотип и подобраны соответствующие цвета. Одна из задач в ходе ребрендинга — изменить все цвета на корпоративном сайте компании. К счастью, сайт сделан с применением каскадных таблиц стилей (CSS), поэтому изменение цветов на сайте сводится к правке CSS-файлов.

Руководство компании не хочет слишком сильно изменять цветовую схему сайта, поэтому они решили, что каждый цвет на сайте будет заменён на наиболее похожий на него цвет из нового корпоративного стиля.

Каждый цвет, как в корпоративном стиле, так и в CSS-файле можно записать тройкой чисел (r, g, b), обозначающих интенсивность красного, зелёного и синего, соответственно $0 \le r, g, b \le 255$. Для того, чтобы запись была более компактна, тройку чисел пишут без пробелов в шестнадцатеричном формате, то есть по два символа из набора `0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F` на каждое число. Также в CSS-файле для того, чтобы отличать обозначение цвета от обычного числа, перед ним пишут символ `#`. Таким образом, цвет, в котором 255 красного, 128 зелёного и 0 синего будет в CSS-файле выглядеть как `#FF8000`.

Вам будет выдан список цветов, допустимых в новом корпоративном стиле компании «Dautroom», а также содержимое CSS-файла с сайта компании. Вам требуется заменить каждый цвет из CSS-файла на ближайший к нему цвет из корпоративного стиля. Расстояние между цветами $c_1 = (r_1, g_1, b_1)$ и $c_2 = (r_2, g_2, b_2)$ определяется по следующей формуле:

$$D(c_1, c_2) = |r_1 - r_2| + |g_1 - g_2| + |b_1 - b_2|.$$

Если есть несколько цветов корпоративного стиля, находящихся на идентичном расстоянии от цвета из CSS-файла, требуется выбрать цвет с наименьшей компонентой r, если и таких несколько — требуется из них выбрать цвет с наименьшей компонентой g. Если и после этого цвет однозначно определить не удастся, нужно из оставшихся претендентов взять цвет с минимальной компонентой b.

Для упрощения можно считать, что в CSS-файле любая последовательность из шести шестнадцатеричных цифр, идущая после символа `#` считается цветом. Также, не нужно проверять CSS-файл на корректность, требуется только заменить цвета. В компании придерживаются строгого стиля кодирования, поэтому цвета в CSS-файле будут всегда обозначаться заглавными латинскими буквами.

Формат входного файла

В первой строке входного файла находится целое число N — количество цветов в новом фирменном стиле компании ($1 \le N \le 100$). Далее в N строках идёт по одному цвету в формате CSS-файла — цвета из фирменного стиля. Оставшаяся часть входного файла представляет собой содержимое CSS-файла, цвета в котором нужно заменить.

Гарантируется, что во входном файле встречаются только символы с кодами 10, 13 (перевод строки), 9 (табуляция), 32 (пробел) и кодами от 33 до 127 включительно. Гарантируется, что суммарная длина входного файла не превосходит 70 килобайт.

Формат выходного файла

Выведите содержимое CSS-файла, в котором все цвета заменены на ближайшие к ним по указанному выше алгоритму.

Примеры

css.in	css.out

2
#FF0000
#00FF00
.header {
font-weight: bold;
color: #800000;
background-color: #000000;
}
	.header {
	font-weight: bold;
	color: #FF0000;
	background-color: #00FF00;
	}

3
#FF0000
#FFFF00
#FFFFFF
.header {
color : #FF8000 #FF0001 #00FF00;
}
.footer {
color : #123456 #890ABC #111111;
}
.strange1 {
colors : #1234567 #12345
#ABAGGG #abaaaa #ABAAAA
}
	.header {
	color : #FFFF00 #FF0000 #FFFF00;
	}
	.footer {
	color : #FF0000 #FF0000 #FF0000;
	}
	.strange1 {
	colors : #FF00007 #12345
	#ABAGGG #abaaaa #FFFFFF
	}

Примечание

Во втором примере `#1234567` состоит из описания цвета `#123456` и символа `7`, стоящего отдельно. Строка `#12345` не является описанием цвета, так как описание цвета должно содержать 7 символов, включая `#`. Строка `#ABAGGG` не является описанием цвета, так как содержит символы `G`, не являющиеся шестнадцатеричными цифрами. Строка `#abaaaa` не является описанием цвета по условию задачи, несмотря на то, что с точки зрения CSS является корректным цветом.

Задача C. Мусор

Имя входного файла:	`garbage.in`
Имя выходного файла:	`garbage.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дед Олег поссорился со своим соседом Фёдором. Решив просто так не сдаваться, Олег задумал диверсию — набросать своему соседу под забор мусора.

Олег не хочет мусорить абы где, поэтому он выделил N точек вдоль забора Фёдора (забор длинный, поэтому его можно считать просто прямой линией), в которых он может намусорить. У Олега хватает мусора, чтобы набросать его в K различных точках, причём он хочет, чтобы разных куч мусора было ровно K.

Олегу не чуждо чувство прекрасного, поэтому он не хочет, чтобы расстояние между двумя любыми кучами мусора было меньше R. Олег не хочет ограничиваться одним разом, поэтому он каждую ночь решил приходить к забору своего соседа и мусорить. Однако чтобы его не заподозрили, Олег решил каждый день раскидывать мусор по-разному (два способа различны, если существует хотя бы одна точка, в которой в первом способе есть мусор, а во втором нет).

Олега интересует, сколько дней он сможет таким образом донимать своего соседа. Так как ответ может быть очень большим (Олег весьма изобретателен), он просит вас посчитать это число по модулю $10^9 + 7$.

Формат входного файла

Первая строка входного файла содержит три целых числа N, K и R ($1 \le N \le 10^5$, $1 \le K \le 50$ $1 \le R \le 10^9$) — количество точек для мусора, количество куч, которые хочет сделать Олег и минимальное расстояние между замусоренными точками, соответственно. Вторая строка входного файла содержит N целых чисел $0 \le d_i \le 10^9$ — расстояние до i-й точки от начала забора, все расстояния различны.

Формат выходного файла

В выходной файл выведите единственное число — ответ по модулю $10^9 + 7$.

Примеры

garbage.in	garbage.out
3 2 1 1 2 3	3
3 2 2 1 2 3	1

Задача D. Перестановка

Имя входного файла:	`permutation.in`
Имя выходного файла:	`permutation.out`
Ограничение по времени:	2 секунда
Ограничение по памяти:	256 мегабайт

Егор — самый счастливый мальчик на свете, ведь у него есть своя собственная перестановка!

Перестановка из N чисел — это набор целых чисел от 1 до n, в котором каждое число встречается ровно один раз. Например, $\{1, 4, 3, 2\}$ — перестановка из четырех чисел, а $\{1, 1, 3, 4\}$ и $\{1, 5, 2, 3\}$ — нет.

Необычный мальчик Антоша очень завидует Егору. Еще бы, у Антоши нет своей перестановки, а только набор из N целых чисел a_i. Когда Егор в очередной раз хвастался перестановкой, Антоша твердо решил заполучить свою собственную перестановку из N чисел из имеющегося набора.

Антоша может совершить мысленное усилие и заменить число в своём наборе на любое другое число. Однако для совершения такого усилия ему требуется съесть шоколадку, поэтому он хочет совершить минимальное количество усилий и получить в результате из своего набора перестановку.

Заметьте, что порядок чисел важен и Антоша не может менять числа местами, а может только одно число заменить на другое.

Формат входного файла

В первой строке входного файла находится целое число N — количество чисел в наборе ($1 \le N \le 299299$). Во второй строке входного файла находятся N разделенных пробелами целых чисел a_i из набора Антоши ($1 \le a_i \le 10^9$).

Формат выходного файла

В единственной строке выходного файла выведите N целых чисел — перестановку, полученную описанным способом за минимальное количество обращений. Если ответов несколько, выведите любой.

Примеры

permutation.in	permutation.out
3 1 4 3	1 2 3
3 1 2 3	1 2 3
3 1 1 3	1 2 3

Задача E. Многоугольник

Имя входного файла:	`polygon.in`
Имя выходного файла:	`polygon.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В этой задаче не будет длинных условий про мудрых королей, собеседований на работу или про странного мальчика по имени Антон. В этой задаче вообще не будет условия. Жюри олимпиады просит вас написать программу, которая по заданному числу n выведет выпуклый n–угольник со следующими свойствами:

1. все вершины многоугольника имеют целочисленные координаты лежащие в промежутке $[-10^9, 10^9]$;
2. длины всех сторон многоугольника являются целыми числами.

Напоминаем, что *выпуклым многоугольником* называется многоугольник, обладающий тем свойством, что все его точки лежат по одну сторону от любой прямой, проходящей через две его соседние вершины, и при этом никакие три его точки не принадлежат одной прямой.

Формат входного файла

Единственная строка входного файла содержит целое число n — количество вершин многоугольника ($3 \le n \le 100$).

Формат выходного файла

В выходной файл выведите n строк: i–я строка должна содержать два целых числа x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$) — координаты i–й точки многоугольника. Координаты следует выводить в порядке обхода многоугольника по или против часовой стрелки. Если ответов несколько, то разрешается вывести любой.

Примеры

polygon.in	polygon.out
3	0 0 0 3 4 0
4	0 0 1 0 1 1 0 1

Задача F. Революция

Имя входного файла:	`rebel.in`
Имя выходного файла:	`rebel.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В одной далекой стране правит очень мудрый и дальновидный король. Как это всегда бывает, многие люди не согласны с правлением мудрого и дальновидного короля и хотят его свергнуть. Обычно, чтобы свергнуть короля, люди объединяются в коалиции. Рассматриваемая страна не стала исключением. Всего в стране есть n различных коалиций, i-я коалиция имеет влияние в c_i различных городах (при этом в любом городе может иметь влияние только одна коалиция). Внутри i-й коалиции города нумеруются от единицы до c_i.

На данный момент инфраструктура страны оставляет желать лучшего — никакие два города в стране не соединены друг с другом дорогой. Страны-соседи уже давно начали строить дороги между городами, поэтому, чтобы не отставать от современных тенденций и не прослыть консерватором, король решил строить по одной дороге в год. Обычно дорога соединяет два каких-то города, причем не важно, к какой коалиции эти города относятся. Так как строители пока малоопытны, то иногда может получаться дорога с началом и концом в одном и том же городе. А иногда из-за несогласованных планов строители могут соединять дорогой два города, между которыми уже имеется дорога.

Так как король мудр, то он приказал первому министру страны составить план строительства дорог на ближайшие m лет. Так как король дальновиден, то он подозревает своего первого министра в измене и поэтому попросил вас найти первый год, после которого между двумя различными городами одной и той же коалиции будет существовать путь из уже построенных дорог (т.е. можно будет по дорогам добраться из какого-то города в некоторый другой город этой же коалиции). Как только такой путь появится, сторонники этой коалиции устроят революцию и свергнут короля, поэтому он не может такого допустить.

Естественно, от просьбы короля отказаться невозможно, иначе вас посчитают изменником родины и казнят.

Формат входного файла

Первая строка входного файла содержит целое число n — количество различных коалиций в стране ($1 \le n \le 10^5$).

В следующей строке входного файла перечислены n целых чисел c_i — количество городов подвластных i-ой коалиции ($1 \le c_i \le 10^9$).

В следующей строке входного файла содержится целое число m — количество дорог в плане первого министра ($1 \le m \le 10^5$).

В следующих m строках перечислены дороги: j-ая строка содержит четыре целых числа f_j, u_j, s_j, v_j — в j-ый год город под номером u_j, принадлежащий коалиции f_j, соединяется дорогой с городом под номером v_j, который принадлежит коалиции s_j ($1 \le f_j, s_j \le n$, $1 \le u_j \le c_{f_j}$, $1 \le v_j \le c_{s_j}$).

Формат выходного файла

Если министр оказался верен своему королю, и после выполнения его плана никакие два города, принадлежащие одной коалиции, не окажутся соединены путём, то в единственной строке выходного файла выведите «`Plan is OK.`» (без кавычек). В противном же случае выведите «`Rebel after x year(s).`» (без кавычек) где x — первый год, после которого между какими-то двумя городами, принадлежащими одной и той же коалиции, будет существовать путь.

Примеры

rebel.in	rebel.out
3 2 2 2 3 1 1 2 2 2 2 1 2 1 2 3 1	Rebel after 2 year(s).
3 2 2 2 3 1 1 2 2 1 2 2 1 2 2 3 1	Plan is OK.

Задача G. Сахар

Имя входного файла:	`sugar.in`
Имя выходного файла:	`sugar.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Программист — это состояние души!

некто

Профессор Владимир Владимирович любит пить сладкий чай. Сахароза полезна для стимулирования работы клеток мозга. Время от времени, запасы сахара заканчиваются и приходится наведываться в магазин. Чтобы оптимизировать затраты времени на походы в магазин, Владимир Владимирович обычно покупает сразу много килограмм сахара.

Сахар продаётся в мешках по одному, пять и двадцать пять килограмм. Будучи в преклонном возрасте, профессор боится забыть основное правило маркетинга (себестоимость единицы товара тем ниже, чем выше объем товара) находясь непосредственно в магазине, потому решил заранее написать программу, которая рассчитает наиболее выгодный способ покупки заданного количества сахара.

Вам, как студенту этого профессора, было выдано домашнее задание — проверить правильность расчётов, написав свою версию программы.

Формат входного файла

Во входном файле находится одно единственное целое число X ($1 \le X \le 1000$) — требуемое количество килограмм сахара.

Формат выходного файла

В выходной файл выведите три целых числа — сколько нужно купить мешков сахара по 25, 5 и 1 килограмм, чтобы стоимость покупки была минимальна, а общий вес был равен X килограмм. Помните, что наименьшая себестоимость одного килограмма сахара в мешках по двадцать пять килограмм, а наибольшая — в мешках по одному килограмму!

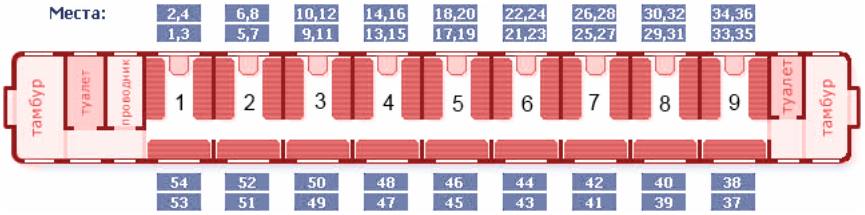
Примеры

sugar.in	sugar.out
6	0 1 1
23	0 4 3
31	1 1 1

Задача N. РЖД

Имя входного файла:	`train.in`
Имя выходного файла:	`train.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Перед вами представлен типичный плацкартный вагон РЖД. Места обозначены в синих прямоугольниках белым цветом. Чёрным цветом выделены номера секций. Нечётные номера — нижние места, чётные номера — верхние. Каждая секция состоит из купе (в котором четыре полки) со столом и двух боковых полок, разделённых проходом.



Чаще всего, участники разнообразных олимпиад по программированию добираются до места проведения олимпиады в таких вагонах. Один из сопровождающих, часто возивших школьников на олимпиады, и по совместительству человек, ответственный за покупку билетов, обнаружил некоторые закономерности в нумерации мест. Теперь он хочет так купить билеты, чтобы разместить всех компактно, по секциям и чтобы каждый программист ехал в одном направлении на нижней полке, а в другом — на верхней.

Для решения этой головоломки, сопровождающему необходимо узнавать по номеру места:

- в какой секции находится место;
- купе это или боковая полка;
- верхняя это или нижняя полка.

Помогите сопровождающему в написании такой программы и тогда вам достанется лучшее место во время поездки!

Формат входного файла

Во входном файле находится одно единственное целое число X ($1 \le X \le 54$) — номер места.

Формат выходного файла

В выходной файл выведите предложение:
«`%positon% %comfort% seat, section #%section%`», где:

- `%positon%` — «`upper`» или «`lower`», если полка верхняя или нижняя;
- `%comfort%` — «`compartment`» или «`lateral`», если полка в купе или боковая;
- `%section%` — номер секции от 1 до 9.

Примеры

train.in	train.out
1	lower compartment seat, section #1
46	upper lateral seat, section #5

Задача I. Последнее испытание

Имя входного файла:	`trial.in`
Имя выходного файла:	`trial.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы проходите собеседование в элитное засекреченное общество «`Alternative Computer Modeling`». Вам осталось пройти последнее испытание — по заданным n и k выдать строку длины n, имеющую максимально возможный период. Строка должна состоять из первых k строчных букв латинского алфавита. Как, вы не знаете что такое период строки? На ваше счастье мы напомним определение.

Периодом строки s длины n называется минимально возможное число p такое, что $\forall i = 1 \dots n-p$ выполняется следующее: $s_i = s_{i+p}$

Например, период строки `abaabaa` равен трём, а период строки `abcd` равен четырём.

Формат входного файла

Единственная строка входного файла содержит два целых числа n и k ($1 \le n \le 100, 1 \le k \le 26$).

Формат выходного файла

В единственную строку выходного файла выведите строку с максимально возможным периодом. Если таких строк несколько, то разрешается вывести любую. Строка должно состоять из первых k строчных букв латинского алфавита.

Примеры

trial.in	trial.out
3 3	abc
3 1	aaa