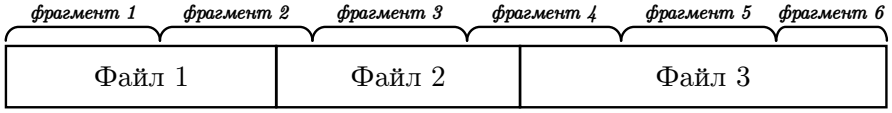
Задача A. BitTorrent

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

BitTorrent — пиринговый (*peer-to-peer*) сетевой протокол для передачи больших файлов, в котором узлы действуют и как клиенты и как серверы, в отличие от централизованной клиент-серверной архитектуры, где клиентские узлы запрашивают ресурсы у центрального сервера. Действительно, протокол BitTorrent позволяет группе хостов одновременно раздавать и получать друг у друга файлы, что снижает нагрузку и зависимость от каждого клиента-источника и обеспечивает избыточность данных.

А именно, раздача (так называемый торрент) включает в себя набор файлов, разбитых на фрагменты, как показано на рисунке. Например, пакет файлов общим размером в 10 МиБ[1] может быть разделён на десять фрагментов размера 1 МиБ или на сорок фрагментов по 256 КиБ. Как только хост (так называемый пир) получает новый фрагмент торрента, он становится источником этого фрагмента для других хостов, которым также необходим этот фрагмент. Как правило, фрагменты загружаются непоследовательно и переставляются в правильном порядке самими пирами. Каждый хост самостоятельно управляет тем, какие фрагменты должны быть загружены. Фрагменты имеют одинаковый размер в рамках одного торрента, за исключением последнего, который может иметь меньший размер. Фрагменты скачиваются и закачиваются только целиком.



Набор файлов (торрент)

Вы хотите скачать один торрент, но приближается ваш месячный лимит входящего интернет-трафика, и на скачивание всех файлов остатка может не хватить. Тем не менее, вы не желаете ждать до следующего месяца и хотите заполучить хотя бы некоторые файлы из торрента. Какое наибольшее число отдельных файлов можно скачать, уложившись в лимит?

Формат входных данных

В первой строке через пробел записано три целых числа N, P и L, где N — количество файлов в торренте ($1 \leqslant N \leqslant 3000$), P — размер фрагмента в КиБ ($1 \leqslant P \leqslant 1000$), L — месячный остаток интернет-трафика в КиБ ($1 \leqslant L \leqslant 1\,000\,000$). Во второй строке через пробел записано N чисел S_1, S_2, …, S_N, где S_i — размер i-го файла в КиБ ($1 \leqslant S_i \leqslant 100\,000$). Файлы перечислены в порядке следования в торренте.

Формат выходных данных

Выведите одно число — максимальное количество файлов, которые можно скачать.

Примеры

bittorrent.in	bittorrent.out
3 3 13 5 5 7	2
7 2 16 6 11 3 3 8 1 8	4

[1] Вместо десятичных приставок кило-, мега- и т. д., обозначающих 10^3, 10^6 и т. д. единиц, международная электротехническая комиссия в марте 1999 года предложила использовать двоичные приставки кеби-, меби- и т. д., обозначающие 2^{10}, 2^{20} и т. д. единц.

Задача B. Прохладительные напитки

Имя входного файла:	`drink.in`
Имя выходного файла:	`drink.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Фирма выпускает прохладительные напитки в пластиковых бутылках различного объёма. Себестоимость бутылки складывается из себестоимости тары и себестоимости самого напитка. Первая величина не зависит от об объёма бутылки, а вторая — пропорциональна объёму налитого напитка. Так, если себестоимость бутылки равна 10, а себестоимость одного литра напитка — 5, то суммарная себестоимость бутылки напитка объёмом в 1.5 литра будет равна 17.5.

Известно, что суммарная себестоимость бутылки объёмом a_1 литра равна b_1, а объёмом a_2 литра — b_2. Рассчитайте суммарную себестоимость бутылки объёмом a_3 литра.

Формат входных данных

В первой строке файла записаны величины a_1 и b_1, а во второй — величины a_2 и b_2. Наконец, третья строка содержит значение a_3. Все числа положительные, не превосходящие 10^4 и записаны не более чем с четырьмя цифрами в дробной части. Величины a_1, a_2 и a_3, а также b_1 и b_2 попарно различны.

Во всех тестах стоимость бутылки и удельная стоимость напитка положительны.

Формат выходных данных

Выведите единственное число — ответ на задачу. Ответ будет принят, если абсолютная или относительная погрешность от правильного не превышает 10^{-4}.

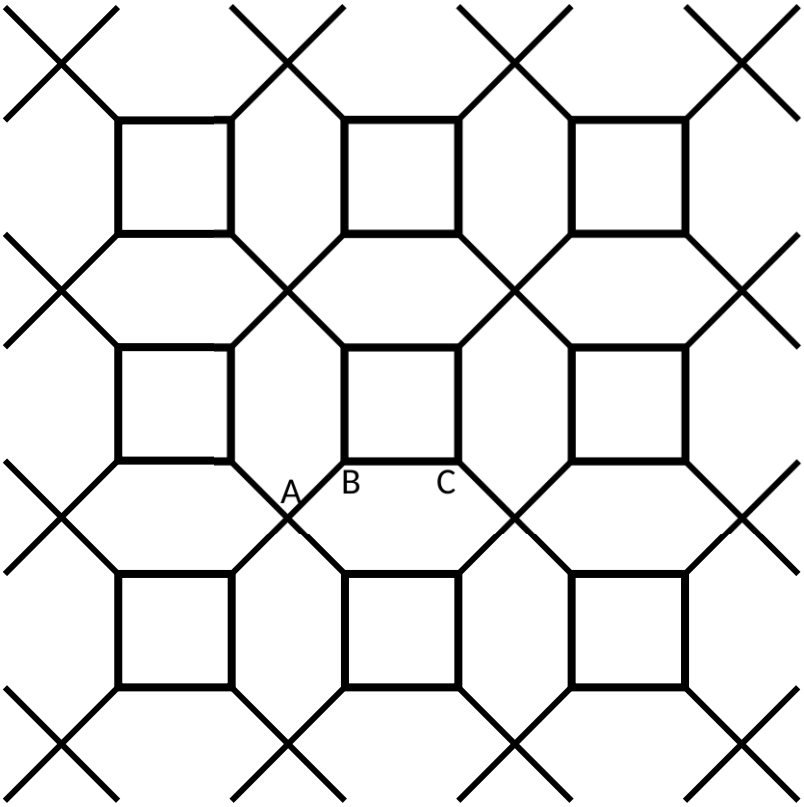
Пример

drink.in	drink.out
1 15 1.5 17.5 3.5	27.50000000

Задача C. По клумбам не ходить!

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

Садовник разбил в большом саду клумбы, имеющие вид квадратов и неправильных шестиугольников, как показано на рисунке. Затем архитектор наложил на план сада координатную сетку так, что точки A, B и C получили соответственно координаты (0, 0), (1, 1) и (3, 1).



Путешественник находится в точке (0, 0) и желает пройти в точку (x, y). Двигаться можно только по дорожкам, разделяющим клумбы! Рассчитайте минимальное расстояние, которое должен пройти путешественник.

Формат входных данных

Единственная строка содержит значения x и y — целые числа, не превосходящие по модулю 10^9.

Формат выходных данных

Если точка (x, y) не находится на дорожке, выведите -1. Иначе выведите единственное число — ответ на задачу. Ответ будет принят, если абсолютная или относительная погрешность от правильного не превышает 10^{-9}.

Примеры

garden.in	garden.out
3 1	3.414213562373095
10 10	-1

Задача D. НОД и XOR

Имя входного файла:	`gcd-xor.in`
Имя выходного файла:	`gcd-xor.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Для заданного числа N посчитайте количество пар целых чисел A и B, для которых

$$1 \leqslant A \leqslant B \leqslant N$$

и

$$\text{НОД}\{\, A, B \,\} = A \oplus B,$$

где $\text{НОД}\{\, A, B \,\}$ — наибольший общий делитель чисел A и B, $A \oplus B$ — значение побитового исключающего «или» (или, что то же самое, результат поразрядного сложения по модулю два).

Формат входных данных

Вход содержит T независимых тестов. В первой строке записано целое число T ($1 \leqslant T \leqslant 1000$). В каждой из последующих T строк записано целое число N ($1 \leqslant N \leqslant 30\,000\,000$).

Формат выходных данных

Выведите T строк, в каждой строке выведите номер теста и ответ. Придерживайтесь формата вывода, который указан в примере.

Пример

gcd-xor.in	gcd-xor.out
2 7 10000000	Case 1: 4 Case 2: 17440305

Задача E. Возрастание с разрывами

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

Необходимо из заданной числовой последовательности A, состоящей из N элементов, вычеркнуть минимальное число элементов, чтобы в оставшейся подпоследовательности каждый последующий элемент был строго больше предыдущего, кроме не более чем M пар соседних элементов (разрывов).

Формат входных данных

Первая строка содержит целые числа N и M ($0 \leqslant M < N \leqslant 30\,000$, $M \leqslant 100$). Следующая строка содержит N элементов последовательности A, которые разделены пробелом (все числа целые, по модулю не превосходящие $1\,000\,000\,000$).

Формат выходных данных

Выведите одно целое число — длину наибольшей подпоследовательности.

Примеры

m-gaps.in	m-gaps.out
9 2 1 2 12 7 3 8 14 13 9	7
9 3 1 2 12 7 3 8 14 13 9	8

Задача F. Цветные картинки

Имя входного файла:	`rectangles.in`
Имя выходного файла:	`rectangles.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Рассмотрим белый прямоугольник R, разбитый на $N \times M$ единичных квадратов, с N строками и M столбцами. Будем считать, что строки пронумерованы сверху вниз от 1 до N, а столбцы — слева направо от 1 до M. Множество клеток матрицы

$$\{(i, j) \mid r_1 \leqslant i \leqslant r_2,\ c_1 \leqslant j \leqslant c_2\}$$

будем называть подпрямоугольником (здесь r_1, r_2, c_1, c_2 — произвольные целые числа, $1 \leqslant r_1 \leqslant r_2 \leqslant N, 1 \leqslant c_1 \leqslant c_2 \leqslant M$).

Проделаем последовательно K раз следующую операцию: выберем произвольный белый подпрямоугольник в R и перекрасим все клетки этого подпрямоугольника в новый цвет (к примеру, первый раз покрасим в жёлтый, затем в красный, зелёный, синий и чёрный). Итоговый прямоугольник будет иметь какое-то количество цветных клеток и, возможно, белые клетки.

Определите, сколько различных итоговых прямоугольников может быть после K перекрашиваний.

Формат входных данных

В единственной строке записаны три целых числа N, M и K ($1 \leqslant N, M \leqslant 100$, $1 \leqslant K \leqslant 5$). Числа в строке разделены единичными пробелами.

Формат выходных данных

В единственной строке выведите количество различных результатов покраски прямоугольника.

Примеры

rectangles.in	rectangles.out
1 1 1	1
1 1 4	0
2 2 4	24
2 3 5	1560
100 100 2	361263595027500

Задача G. Бег

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

Чтобы поддерживать себя в хорошей физической форме, студент Вася решил добираться по утрам из общежития в университет бегом. Вася выбирает маршрут пробежки: хочется, чтобы в начале пути нужно было поднапрячься (чтобы сначала дорога шла в гору), а затем во время второй части пути можно было расслабиться (и насладиться лёгким ветерком на спуске).

В маршруте, конечно, могут попадаться горизонтальные участки. Тем не менее, первая часть пути должна включать некоторый подъём и не может содержать спусков (чтобы бегун не расслаблялся), а вторая часть пути, наоборот, обязана включать спуск под уклон и не может предусматривать движения вверх (под конец у нетренированного бегуна сил уже не хватает...). Движение всё время по ровной местности Васю бы не устроило: это скучно.

Используя информацию о рельефе и конфигурации дорожной сети города, в котором живёт Вася, помогите ему выбрать самый короткий маршрут пробежки. По всем дорогам города можно двигаться в любом из двух направлений.

Формат входных данных

В первой строке записаны целые числа N и M — число перекрёстков ($2 \leqslant N \leqslant 100\,000$) и число дорог ($0 \leqslant M \leqslant 100\,000$). Перекрёстки пронумерованы целыми числами от 1 до N. Общежитие находится возле перекрёстка 1, здание университета — возле перекрёстка N. Во второй строке записано N чисел h_i — высоты расположения перекрёстков. Высота задаётся в метрах над уровнем моря целым числом в промежутке от $-10\,994$ (глубина Марианского жёлоба) до $8\,848$ (высота горы Джомолунгма) включительно. В последующих M строках описываются дороги. Каждая дорога задаётся двумя числами u_i и v_i — номерами перекрёстков, которые она соединяет ($1 \leqslant u_i \neq v_i \leqslant N$), — и числом d_i — длиной в метрах ($1 \leqslant d_i \leqslant 10\,000$). Гарантируется, что между любой парой перекрёстков может быть не более одной дороги. Высота местности при движении вдоль любой дороги изменяется монотонно (возрастает, убывает или остаётся постоянной в зависимости от высот концевых перекрёстков).

Формат выходных данных

Выведите одно число — длину кратчайшего маршрута из общежития в университет, который удовлетворяет требованиям Васи. Если такого маршрута нет, выведите -1.

Примеры

running.in	running.out
4 5 100 100 101 99 1 2 3 2 3 5 3 4 7 1 4 14 2 4 10	15
3 2 100 99 100 1 2 1 2 3 1	-1

Задача H. Сортировка

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

Не так давно Рома стал свидетелем того, как шеф-повар в ресторане лихо переворачивал сразу по три рядом лежащие отбивные на гриле при помощи специальной лопатки. Это натолкнуло Рому на мысль о новом методе сортировки массивов, где за один шаг выполняется разворот некоторой тройки подряд идущих элементов.

Задан массив размера N. Доступна такая операция: заменить три последовательных элемента массива (x, y, z) на эти же элементы, но в обратном порядке (z, y, x). Помогите Роме определить, можно ли упорядочить массив по возрастанию и какое минимальное число операций для этого необходимо.

Формат входных данных

В первой строке записано целое число N ($1 \leqslant N \leqslant 1\,000\,000$). Во второй строке через пробел записаны элементы массива — попарно различные целые числа от 1 до N.

Формат выходных данных

Выведите одно число — минимальное количество операций, или -1, если отсортировать массив указанным способом нельзя.

Примеры

sorting.in	sorting.out
6 3 4 1 6 5 2	3
5 3 1 5 4 2	-1

Замечание

В первом примере одна из возможных последовательностей операций такова:

3	4	1	6	5	2

3	4	1	2	5	6

1	4	3	2	5	6

1	2	3	4	5	6

Задача I. Сверхпростые числа

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

«Простые числа — это слишком просто!» — подумал учитель мальчика Кости и предложил ему найти несколько сверхпростых чисел.

Назовём простое число сверхпростым, если оно остаётся простым после отбрасывания цифр справа. Например, число 7331 является сверхпростым, потому что

- 7331 — простое,
- 733 — простое,
- 73 — простое,
- 7 — простое.

Напомним, что число является простым, если имеет ровно два различных натуральных делителя.

Формат входных данных

В первой строке записано два целых числа A и B ($1 \leqslant A \leqslant B \leqslant 10^9$).

Формат выходных данных

В первой строке выведите число K — количество сверхпростых чисел на отрезке от A до B включительно. В последующих K строках выведите эти числа в порядке возрастания.

Примеры

superprimes.in	superprimes.out
23000000 32000000	2 23399339 29399999
1 2	1 2

Задача J. Путешествие

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

Коля очень любит путешествовать, но еще больше он любит собирать мили. Он живет в городе A, финальная часть соревнования, в которую он вышел, будет проходить в городе B. Организаторы сообщили Коле, что он может выбрать любой маршрут. Коля очень суеверный и при путешествии на соревнование всегда делает нечётное количество пересадок.

Приятная в общении девушка сообщила Коле, что он может сделать не более K пересадок. К его счастью, он может делать пересадку в любой точке планеты (даже если там нет аэропорта). Коля не хочет, чтобы его заподозрили в пустой трате денег организаторов, поэтому пересадки он хочет делать в различных точках.

В нашем случае планета — сфера с радиусом 3959 миль и центром в начале координат. Две точки считаются различными, если расстояние между ними не менее одной сотой мили по сфере. При перемещении из точки X в точку Y всегда выбирается кратчайшее расстояние по сфере между этими точками, в этом случае Коле на его счет будет добавлено количество миль, равное этому кратчайшему расстоянию.

Заметим, что если Коля определил некоторую последовательность S из n точек для пересадки, то его путешествие будет следующим: $A \to S_1$, $S_1 \to S_2$, $\ldots$, $S_{n-1} \to S_n$, $S_n \to B$.

Формат входных данных

В первой строке записаны три целых числа X_A, Y_A, Z_A — координаты города A. Во второй строке записаны три целых числа X_B, Y_B, Z_B — координаты города B. В третьей — единственное целое число K ($1 \leqslant K \leqslant 100$). Точки A и B лежат на сфере и не совпадают. Все координаты заданы в милях.

Формат выходных данных

В первой строке выведите выведите два числа: максимальное количество миль (с относительной точностью не менее 10^{-9}), которое Коля может заработать, и количество пересадок, которое он при этом должен совершить. Далее для каждой пересадки укажите, в какой точке планеты она должна быть совершена (три вещественных числа), точка должна быть на планете с относительной точностью не менее 10^{-9}. Расстояние между любыми двумя точками путешествия (A, B, последовательность S), не обязательно последовательными, должно быть не менее 0.01 мили по сфере.

Пример

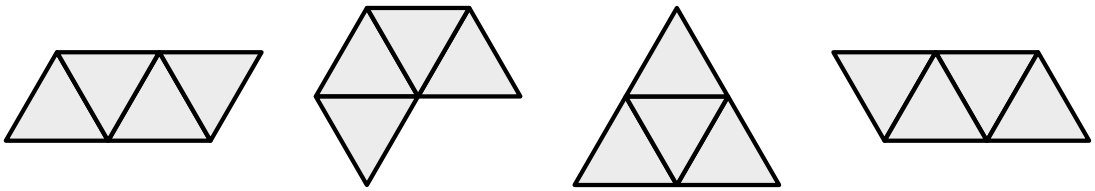
travel.in	travel.out
3959 0 0 -3959 0 0 1	12437.56531556199 1 0 2799.43574671754 -2799.43574671754

Задача K. Треугольная пицца

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

Пиццерия «Pizza C:\Temp'o» освоила выпуск нового типа пиццы с кусочками треугольной формы. Пицца состоит из нескольких одинаковых равносторонних треугольников. Более того, фигура является связной (два треугольника непосредственно соседствуют, если они имеют общую сторону), но может содержать «дырки».

Сколько всего существует различных форм для пиццы из N кусков? Две фигуры считаются одинаковыми, если их можно совместить наложением (фигуры можно двигать и вращать в плоскости, но нельзя переворачивать).



Формат входных данных

На входе записано одно целое число N ($1 \leqslant N \leqslant 16$).

Формат выходных данных

Выведите одно число — количество возможных форм пиццы.

Примеры

triangular.in	triangular.out
2	1
4	4

Задача L. UUE-кодирование

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

UUE (англ. Uuencode) — метод представления двоичных данных в текстовой форме, пригодной для передачи через средства, предназначенные только для передачи текстов, например через e-mail (*транспортное кодирование*). В настоящее время UUE в интернет-среде был заменён на более совершенный метод MIME, однако сохранил свою популярность в сети Фидонет.

В упрощённом виде процесс UUE-кодирования можно записать следующим образом.

При кодировании из файла берутся данные по 3 байта (в случае, если осталось меньше трёх байт, недостающие заменяются нулями). 24 бита, образующие эти 3 байта, делятся на четыре группы по 6 бит. Каждая шестибитная группа интерпретируется как число (от 0 до 63), к которому добавляется 32. Получившееся число в диапазоне от 32 до 95 трактуется как код символа в ASCII-таблице, так что получаются символы от пробела (символа с кодом 32) до знака подчёркивания (символа с кодом 95).

Каждая группа из 60 символов (соответствует 45 байтам исходного файла) используется для создания отдельной строки. В начале строки записывается символ с кодом, равным количеству закодированных символов в строке, увеличенному на 32 (во всех строках, кроме последней, это символ `M`). Каждая строка завершается символом перевода строки.

После окончания данных кодируемого файла помещается строка, содержащая единственный символ ‘ (код которого равен 96). После этой строки также записывается перевод строки.

В таблице показан пример кодирования строки `Cat`.

Исходные символы	`C`								`a`								`t`							
ASCII-коды (десятичные)	67								97								116							
ASCII-коды (двоичные)	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Новые десятичные значения	16				54				5				52			
+32	48				86				37				84			
Символы UUE	`0`				`V`				`%`				`T`			

На последнем этапе к строке приписывается символ `#`, код которого равен 35 ($3 + 32$).

Таким образом, текстовый файл с результатом UUE-кодирования будет записан как

```
#0V%T
‘
```

Выполните UUE-кодирование исходной последовательности байт.

Формат входных данных

Первая строка входного файла содержит десятичное представление длины кодируемой последовательности (целое число от 1 до 50 000). Вторая строка содержит значения каждого байта этой последовательности, записанные в шестнадцатеричной системе счисления (цифры выбираются из строки `0123456789ABCDEF`). Эти значения разделяются одиночными пробелами.

Формат выходных данных

Текстовый файл с результатом UUE-кодирования.

Пример

uue.in	uue.out
3 43 61 74	#0V%T ‘