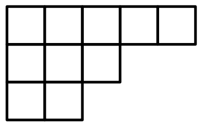
Задача A. Увидеть Юнга и умереть

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

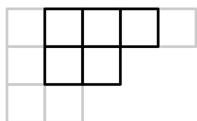
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа n на слагаемые представляет собой сумму вида $n = m_1 + m_2 + \ldots + m_k$, где $m_1 \geqslant m_2 \geqslant \ldots \geqslant m_k$.

Диаграмма состоит из n квадратиков, организованных в виде k рядов, где k количество слагаемых в разбиении. Ряд, соответствующий числу m_i, содержит m_i квадратиков. Все ряды выровнены по левому краю и упорядочены от более длинного к более короткому.

Например, диаграмма Юнга, приведенная на рисунке, соответствует разбиению $10 = 5 + 3 + 2$.



Иногда можно вписать одну диаграмму Юнга в другую. Диаграмму X можно вписать в диаграмму Y, если можно удалить некоторые квадратики из диаграммы Y так, чтобы получилась диаграмма X. Отметим, что разрешается только удалять некоторые квадратики, вращать или отражать диаграмму не разрешается. Например, диаграмма для разбиения $5 = 3 + 2$ может быть вписана в диаграмму для разбиения $10 = 5 + 3 + 2$, как показано на рисунке.



С другой стороны, диаграмму для разбиения $8 = 4+4$ нельзя вписать в диаграмму для разбиения $10 = 5 + 3 + 2$.

Для заданного n найдите такое разбиение n на слагаемые, что в соответствующую ему диаграмму Юнга можно вписать максимальное количество различных диаграмм.

Например, в диаграмму для разбиения $10 = 5 + 3 + 2$ можно вписать 36 различных диаграмм. Однако это не максимальное значение. В диаграмму для разбиения $10 = 4 + 2 + 2 + 1 + 1$ можно вписать 41 диаграмму Юнга.

Формат входных данных

Входной файл содержит целое число n ($1 \leqslant n \leqslant 100$).

Формат выходных данных

На первой строке выходного файла выведите максимальное число диаграмм Юнга, которые можно вписать в некоторую диаграмму, соответствующую разбиению на слагаемые числа n.

На второй строке выведите одно или более целых чисел — количество квадратиков в каждом из рядов оптимальной диаграммы.

Пример

стандартный ввод	стандартный вывод
10	41 4 3 2 1

Задача B. Переворот битов

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

Дана таблица $n \times m$ из 0 и 1. За одно действие вы можете поменять все биты на противоположные (то есть все 0 на 1 и все 1 на 0) в любом столбце или в любой строке этой таблицы. Вы можете совершить k действий. Какое максимальное количество 1 вы можете получить?

Формат входных данных

Первая строка содержит три целых числа n, m и k ($n \times m \leqslant 500$, $k \leqslant 1000$).

Следующие n строк содержат описание таблицы: в i-й из этих строк содержится битовая строка длины m.

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 2 2 11 10 11	5
5 4 4 0011 1100 0001 0101 0010	16

Задача C. Мать драконов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.3 секунд
Ограничение по памяти:	256 мегабайт

В Королевстве Ланнистеров n замков и несколько стен, соединяющих два замка, никакие два замка не соединены более, чем одной стеной, ни одна стена не соединяет замок с собой.

Сир Джейме Ланнистер узнал, что Дейенерис Таргариен собирается атаковать его королевство. Он хочет защитить свои владения. У него есть k литров странной жидкости. Он хочет распределить эту жидкость между замками так, чтобы каждый замок содержал некоторое количество жидкости (возможно, нулевое или нецелое количество литров). После этого стабильность стены, соединяющей замки a и b, содержащие x и y литров жидкости, соответственно, равна $x \cdot y$.

Ваша задача — найти максимальную возможную сумму стабильностей стен, которую Сир Джейме Ланнистер сможет достичь

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leqslant n \leqslant 40$, $1 \leqslant k \leqslant 1000$).

Далее следует n строк. В i-й из них содержится n целых чисел $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($a_{i,j} \in \{0, 1\}$). Если замки i и j соединены стеной, $a_{i,j} = 1$. В противном случае оно равно 0.

Гарантируется, что $a_{i,j} = a_{j,i}$ и $a_{i,i} = 0$ для всех $1 \leqslant i, j \leqslant n$.

Формат выходных данных

Выведите одно число — максимальную возможную сумму стабильностей стен, которую Сир Джейме Ланнистер сможет достичь.

Ваш ответ будет считаться правильным, если его абсолютная или относительная точность не превосходит 10^{-6}.

А именно, если ваш ответ равен a, а ответ жюри равен b, то ваш ответ будет зачтен, если $\frac{|a-b|}{max(1,b)} \leqslant 10^{-6}$.

Примеры

стандартный ввод	стандартный вывод
3 1 0 1 0 1 0 0 0 0 0	0.250000000000
4 4 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0	4.000000000000

Замечание

В первом примере, если замки $1, 2, 3$ содержат $0.5, 0.5, 0$ литров жидкости, соответственно, ответ равен 0.25.

Во втором примере, если замки $1, 2, 3, 4$ содержат $1.0, 1.0, 1.0, 1.0$ литров жидкости, ответ равен 4.0.

Задача D. Число совершенных паросочетаний

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Совершенным называется паросочетание, покрывающее все вершины графа.

Дан произвольный неориентированный граф.
Найдите количество совершенных паросочетаний по модулю $10^9 + 7$.

Формат входных данных

Число вершин n ($1 \leqslant n \leqslant 30$). Числе рёбер m ($0 \leqslant m \leqslant \frac{n(n-1)}{2}$)

Формат выходных данных

Выведите одно число – количество совершенных паросочетаний по модулю $10^9 + 7$.

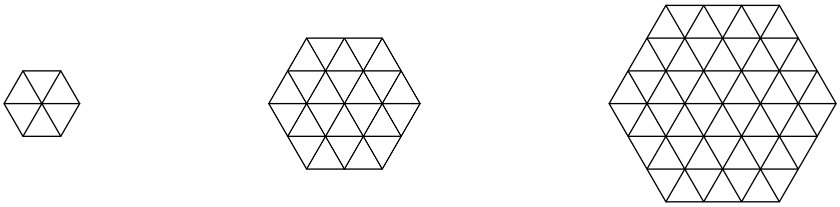
Пример

стандартный ввод	стандартный вывод
4 4 1 3 1 4 2 3 2 4	2

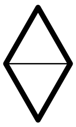
Задача E. Шестиугольные домино-ромбы

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

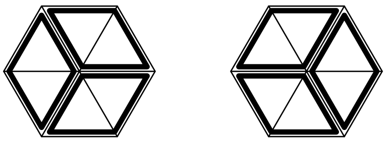
Правильный шестиугольник со стороной длины n разделен на $6n^2$ единичных треугольников.



Его требуется полностью покрыть без наложений и пересечений домино-ромбами — фигурами, составленными из двух единичных треугольников с общей стороной.



Требуется посчитать число способов покрыть шестиугольник таким образом. Например, два способа покрыть шестиугольник со стороной 1 приведены на рисунке.



Формат входных данных

Входной файл содержит число n ($1 \leqslant n \leqslant 7$).

Формат выходных данных

Выведите число способов покрыть шестиугольник домино-ромбами.

Примеры

стандартный ввод	стандартный вывод
1	2
2	20

Задача F. Японский компьютер

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Как известно, для обороны границ японские инженеры разрабатывают огромных боевых человекоподобных роботов. Каждый такой робот управляется японским компьютером. Понятно, что для повышения эффективности робота программа в компьютере должна быть как можно более оптимальной, чтобы компьютер мог выполнять как можно больше программ за как можно меньшее время.

На данный момент японским программистам задали следующую задачу (её смысл секретен, поэтому здесь его описывать нельзя): изначально в памяти компьютера находится единственное число x. Требуется получить в его памяти следующие числа: $a_1x, a_2x, \ldots, a_nx$. При этом компьютер может выполнять следующие операции:

1. Сложение двух чисел
2. Вычитание двух чисел
3. Побитовый сдвиг влево (сдвиг на k бит эквивалентен умножению на 2^k)

Все полученные промежуточные значения сохраняются в памяти, так что ими можно пользоваться при вычислении других значений.

При вычислениях никогда не должно получаться значение большее, чем $42x$. Гарантируется, что при выполнении этого ограничения, в компьютере не происходит переполнений. Также, компьютер не может работать с отрицательными числами, так что вычитать большее число из меньшего также запрещено.

Порядок, в котором в памяти будут появляться числа $a_1x, a_2x, \ldots, a_nx$, не имеет значения.

Формат входных данных

В первой строке находится число n — количество требуемых значений ($1 \leqslant n \leqslant 41$). Во второй строке находится n чисел a_i ($2 \leqslant a_i \leqslant 42$). Все a_i различны. Само число x вам не дано, так что ваша последовательность операций должна быть верной для любого x.

Формат выходных данных

В первой строке выведите единственное число — минимальное количество требуемых операций. Далее выведите требуемые операции в следующем формате:

1. Сдвиг влево ax на k бит: “`a<<k`”
2. Сложение ax и bx: “`a+b`”
3. Вычитание ax из bx: “`b-a`”

Запись операций не должна содержать пробелов.

Примеры

стандартный ввод	стандартный вывод
3 3 5 18	5 1+1 2+1 3+2 1<<4 16+2
1 29	4 1+1 2+1 1<<5 32-3
4 12 19 41 42	8 1+1 2+1 3<<2 12+12 24-2 22-3 19+22 41+1

Задача G. Коробки

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

Дано N коробок в пространстве. Про каждую коробку известны её три измерения. Коробки можно вращать — то есть выбирать, какое измерение будет длиной, какое — шириной, а какое — высотой. Из этих коробок необходимо построить башню максимальной высоты. Коробки разрешается ставить друг на друга, только если длина коробки сверху не превосходит длины коробки снизу, а ширина коробки сверху — ширины коробки снизу.

Формат входных данных

В первой строке задано целое число N ($1 \leqslant N \leqslant 30$). Следующие N строк содержат по три целых числа a_i, b_i и c_i ($1 \leqslant a_i, b_i, c_i \leqslant 10^6$) — три измерения i-ой коробки.

Формат выходных данных

В первой строке выведите максимальную высоту башни H. Во второй строке выведите целое число k — количество коробок в искомой башне. В последующих k строках выведите по четыре числа для каждой коробки — её номер m_i, длину l_i, ширину d_i и высоту h_i (коробки нужно выводить в порядке снизу вверх). Коробки нумеруются с единицы в порядке, заданном во входном файле.

Пример

стандартный ввод	стандартный вывод
3 3 1 3 2 2 2 1 2 1	5 2 1 1 3 3 3 1 1 2

Задача H. Украденный массив

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Мальчик Петя любит массивы. Недавно ему подарили огромный массив чисел F размера 2^n. Петя человек странный, и при виде массива из чисел сразу начинает считать какие-то суммы на нём. Специально для этого он купил в магазине новый пустой массив P размера 2^n и начал его заполнять по следующему правилу: $P[i] = \sum_{j\&i=j} F[j]$. Другими словами для каждого j, такого что j — подмасска i (т.е. побитовое «И» чисел i и j равно j), Петя прибавил $F[j]$ к изначально нулевому значению $P[i]$.

Но потом случилось ужасное — массив F украли! Теперь Петя хочет разыскать массив F, но он не помнит, какие значения там были изначально. Единственное что у него есть — массив P. Помогите Пете восстановить массив F.

Формат входных данных

В первой строке входных данных дано одно число n ($1 \leqslant n \leqslant 20$).

В следующей строке даны 2^n чисел, i-е из них — значение $P[i]$ ($-10^9 \leqslant P[i] \leqslant 10^9$) (нумерация ведётся с нуля).

Формат выходных данных

В одной строке выведите 2^n чисел, i-е из которых — значение $F[i]$ (нумерация ведётся с нуля).

Примеры

стандартный ввод	стандартный вывод
2 1 2 3 4	1 1 2 0
3 1 3 4 10 6 14 16 36	1 2 3 4 5 6 7 8

Задача I. Два капитана

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

Как известно, у Черной Жемчужины два капитана: капитан Джек Воробей и Барбосса. Корабль содержит ровно n пушек, расположенных в ряд. Во время боя оба капитана раз в минуту одновременно дают команды своим матросам. Команды бывают следующих видов:

- `send` l r — послать своих матросов стрелять из пушек с номерами от l до r включительно
- `back` l r — отозвать всех своих матросов с пушек с номерами от l до r включительно. Если на каких-то пушках из этого отрезка нет матросов, подчиняющихся этому капитану, то с такими пушками ничего не происходит
- `rum` — принести еще одну бутылку рома

Каждая команда выполняется мгновенно, после чего сражение идет еще минуту до следующей команды. Если в какой-то момент у одной и той же пушки окажутся матросы, подчиняющиеся разным капитанам, они подерутся и убьют друг друга. Эта ситуация не устраивает никого из капитанов, и поэтому они обратились к вам с просьбой помочь им в решении этой проблемы.

Перед началом очередного сражения капитан Джек Воробей и Барбосса составили планы своих действий. Известно, что план капитана Джека Воробья состоит из m_1 команд, а план Барбоссы — из m_2 команд. В начале i-ой минуты боя каждый капитан дает своим матросам i-ую команду из своего плана, если в нем есть хотя бы i команд. Вам поручили исправить планы так, чтобы все матросы остались живы. Единственная доступная вам модификация плана сражения — вставка нескольких команд `rum` в любые места. Понятно, что капитаны не очень любят менять свои планы, поэтому суммарное количество команд, добавленных Вами в оба плана, должно быть минимально.

Формат входных данных

В первой строке дано число n — количество пушек на корабле ($1 \leqslant n \leqslant 10^9$).

Во второй строке задано число m_1 — количество команд в плане Джека Воробья ($1 \leqslant m_1 \leqslant 3\,000$). В следующих m_1 строках перечислены команды из плана Джека Воробья. Команды заданы так, как они описаны выше. Для всех команд, использующих l и r, верно, что $1 \leqslant l \leqslant r \leqslant n$. Гарантируется, что последняя команда в плане — `back 1` n.

Во следующей строке задано число m_2 — количество команд в плане Барбоссы ($1 \leqslant m_2 \leqslant 3\,000$). В следующих m_2 строках перечислены команды из плана Барбоссы. Команды заданы так, как они описаны выше. Для всех команд, использующих l и r, верно, что $1 \leqslant l \leqslant r \leqslant n$. Гарантируется, что последняя команда в плане — `back 1` n.

Формат выходных данных

В единственной строке выведете минимальное количество дополнительных команд.

Система оценки

Первая группа тестов стоит 20 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие ($n, m_1, m_2 \leqslant 20$).

Вторая группа тестов стоит 35 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие, ($n, m_1, m_2 \leqslant 300$).

Третья группа тестов стоит 25 баллов. Баллы за эту группу начисляются только при прохождении всех тестов группы. Для всех тестов этой группы выполнено условие, ($n, m_1, m_2 \leqslant 1\,000$).

Четвертая группа тестов стоит 20 баллов. Каждый тест этой группы стоит определенное количество баллов.

Пример

стандартный ввод	стандартный вывод
3 4 send 1 1 send 2 2 back 1 1 back 1 3 5 send 2 3 send 1 1 back 2 2 rum back 1 3	3

Задача J. Необычная сортировка

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

Вам дана последовательность различных целых чисел $a_1, a_2, \ldots, a_n$. Все числа в последовательности лежат в отрезке $[0, 2^k - 1]$, где k дано.

Давайте определим функцию $f(x)$ для числа x, которое лежит в отрезке $[0, 2^k - 1]$ как количество инверсий в последовательности $a_1 \oplus x, a_2 \oplus x, \ldots, a_n \oplus x$.

Теперь давайте отсортируем все целые числа из отрезка $[0, 2^k - 1]$ по возрастанию значений функции f, а затем по возрастанию самих чисел.

Вам дана позиция p. Найдите p-е число в этом порядке сортировки всех целых чисел из отрезка $[0, 2^k - 1]$.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leqslant t \leqslant 20$) — количество тестовых случаев. Далее находится описание t тестовых случаев в следующем формате:

Первая строка содержит три целых числа n, k, p, разделенных пробелом ($1 \leqslant n \leqslant 10^6, 1 \leqslant k \leqslant 30, 1 \leqslant p \leqslant 2^k$) — количество чисел в последовательности, параметр k и заданная позиция.

В следующей строке находятся n различных целых чисел $a_1, a_2, \ldots, a_n$, разделенных пробелами ($0 \leqslant a_i < 2^k$).

Сумма всех n во всех тестовых случаях не превосходит 10^6.

Формат выходных данных

Для каждого тестового случая выведите единственное целое число — p-е целое число в порядке сортировки всех целых чисел из отрезка $[0, 2^k - 1]$, описанном в условии задачи.

Пример

стандартный ввод	стандартный вывод
2 4 3 5 2 0 3 7 2 2 1 2 0	4 2

Задача K. Ним

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Сергей Х. и Сергей Б. собираются поиграть в Ним. Сначала они выставляют k стопок камней, содержащие $a_1, a_2, \ldots, a_k$ камней соответственно. Затем они по очереди ходят, причем Сергей Х. начинает первым.

В свой ход игрок выбирает любую непустую кучку и забирает оттуда любое ненулевое количество камней, если на момент хода все кучки пустые, то у игрока нет хода, и он проигрывает.

Из особой любви к простым числам, они решили сделать каждое a_i от 0 до A, где a_i — простое число. По данным k и A найдите число начальных состояний, дающих победу Сергею Б., если оба игрока играют оптимально. Выведите ответ по модулю $10^9 + 7$.

Формат входных данных

Единственная строка входных данных содержит два числа: k и A ($1 \leqslant k \leqslant 10^9$, $2 \leqslant A \leqslant 5 \cdot 10^4$).

Формат выходных данных

Выведите по модулю $10^9 + 7$ число начальных состояний, обеспечивающих выигрыш Сергею Б.

Примеры

стандартный ввод	стандартный вывод
3 7	6
4 13	120

Задача L. Свёртка

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

Рассмотрим все подмножества множества $U = \{0, 1, 2, \ldots, n-1\}$. Каждому подмножеству $A = \{a_1, a_2, \ldots, a_k\}$ соответствует уникальное целое число, равное $p(A) = \sum_{i=1}^{k} 2^{a_i}$. Функцию F от n-элементного множества будем задавать массивом целых чисел f длины 2^n так, что значение функции $F(A)$ равно $f[p(A)]$.

Вам даны две функции F и G, нужно найти функцию H такую, что

$$H(A) = \sum_{B \cup C = A} F(B)G(C).$$

Формат входных данных

В первой строке заданы два целых числа n и t ($1 \leqslant n \leqslant 16$, $1 \leqslant t \leqslant 100$). Здесь n — размер множества U, а t — количество тестовых случаев. Во второй строке заданы целые числа a и b, каждое от 1 до 10^9. Эти числа используются в следующем генераторе псевдослучайных чисел:

```
unsigned int cur = 0; // беззнаковое 32-битное число
unsigned int nextRand16() {
    cur = cur * a + b; // вычисляется по модулю 2^32
    return cur / 2^16; // целое число от 0 до 2^16 - 1
}
```

Тестовые случаи генерируются последовательно. В каждом из них сперва генерируются по порядку элементы массива f (значения функции F), а затем генерируются по порядку элементы массива g (значения функции G). Каждое следующее целое число генерируется вызовом функции `nextRand16()`.

Формат выходных данных

В ответ на каждый тестовый случай выведите в отдельной строке одно целое число: $\left(\sum_{A} H(A) \cdot (p(A)+1)\right) \bmod 2^{32}$.

Примеры

стандартный ввод	стандартный вывод
3 2 30 239017	2723387430 3167905008
16 2 239 17	551267264 1632349120

Замечание

Массивы в первом тесте из примера:
f_1: 3, 113, 3395, 36331, 41370, 61471, 9130, 11774
g_1: 25547, 45526, 55066, 13590, 14501, 41817, 9356, 18543
h_1: 76641, 8167827, 273846333, 5284992017, 1656829263, 11450721456, 3699971823, 14260048942
f_2: 32024, 43238, 51978, 52034, 53714, 38578, 43250, 52338
g_2: 62834, 50034, 59250, 8050, 44914, 36722, 53106, 20338
h_2: 2012196016, 6482475400, 8243104152, 15561662464, 7225902008, 16869349792, 22350138288, 44342816072