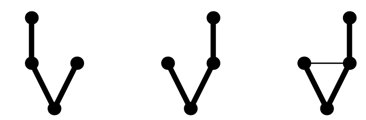
Задача A. Каталонские леса

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

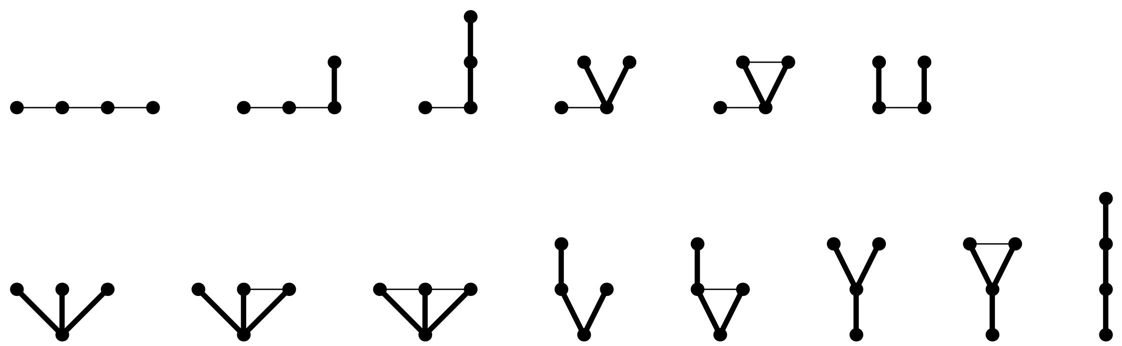
Рассмотрим корневые непомеченные деревья. Для каждой вершины дерева разделим её детей на одну или более групп, где каждый ребенок принадлежит ровно одной группе. При изображении таких деревьев вершины из одной группы располагаются подряд и соединяются горизонтальной чертой. Будем называть такое дерево *каталонским деревом*. На рисунке показаны три каталонских дерева с 4 вершинами, корень изображён внизу. Первые два дерева на самом деле представляю собой различные изображения одного и того же дерева, а вот третье дерево отличается от них, поскольку два ребёнка корня находятся в одной группе.



Каталонский лес — это неупорядоченный набор из одного или нескольких каталонских деревьев.

По заданному n найдите количество различных каталонских лесов, где суммарно все деревья леса содержат n вершин.

На рисунке ниже показаны все 14 каталонских лесов с 4 вершинами. Для удобства корни деревьев одного леса соединены горизонтальной чертой. Корни деревьев изображены внизу.



Формат входных данных

На вход подаётся несколько тестов. Каждый тест задаётся одним целым числом n, расположенным в отдельной строке ($1 \leqslant n \leqslant 60$). Последняя строка ввода содержит число 0, её не следует обрабатывать.

Формат выходных данных

Для каждого теста выведите одно число — количество каталонских лесов с n вершинами. Ответ следует выводить по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
1 2 3 4 5 0	1 2 5 14 42

Задача B. Абстрактные танцы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Последнее время абстрактные танцы набирают популярность. Петин учитель танцев, мсье Кот О'Лан недавно изобрел серию абстрактных танцев, которую он планирует представить на приближающемся фестивале танцев.

Абстрактный танец представляет собой одно *движение*. Движение состоит из набора *фигур*, показанных одна за другой и разделенных *поворотами*. Фигура в свою очередь состоит из *шага* за которым следует (возможно пустое) мультимножество композиций, разделенных *флипами*, после которых следует *прыжок*.

Петя хочет понять, насколько разнообразно изобретение его учителя. Он заметил, что хотя некоторые танцы проходят немного по-разному, на самом деле являются по сути одним танцем. Он называет две фигуры эквивалентными, если соответствующие им мультимножества движений совпадают (порядок, в котором они выполняются, не имеет значения). Петя называет два движения эквивалентными, если циклическим сдвигом одного из них можно добиться, чтобы они представляли собой эквивалентные последовательности фигур. Петя считает два танца различными, если их движения не эквивалентны.

Формально, пусть фигура F_1 состоит из движений $m_1, m_2, \ldots, m_k$ и фигура F_2 состоит из движений $n_1, n_2, \ldots, n_k$. Тогда фигуры F_1 и F_2 эквиваленты, если существует биекция $\varphi : \{1 \ldots k\} \to \{1 \ldots k\}$, такая что m_i эквивалентно $n_{\varphi(i)}$ для всех i от 1 до k. Пусть движение M_1 состоит из фигур $f_1, f_2, \ldots, f_l$, а движение M_2 состоит из фигур $g_1, g_2, \ldots, g_l$. Движения M_1 и M_2 эквивалентны, если существует такое j, что f_i эквивалентна $g_{(i+j) \bmod n+1}$ для всех i от 1 до l.

Например, обозначим поворот как «`,`», шаг как «`(`», прыжок как «`)`» и флип как «`!`». Тогда фигуры «`((()))`» и «`(()!())`» различны, а вот фигуры «`(()!(()))`» и «`((())!())`» эквивалентны (потому что они только различаются порядком движений между шагом и прыжком). Движения «`(),(()),((()))`» и «`(()),(),((()))`» различны, а вот движения «`(),(()),((()))`» и «`(()),((())),()`» эквивалентны, потому что они совпадают с точностью до циклического сдвига.

Пете стало интересно, сколько различных абстрантных танцев, содержащих ровно n шагов, существует. Помогите ему это выяснить. Ниже приведены все 5 абстрактных танцев, содеражщих 3 шага. «`((()))`», «`(()!())`», «`((),())`», «`(()),()`» (эквивалентен «`(),(())`»), «`(),(),()`».

Формат входных данных

Входные данные содержат несколько тестов. Каждый тест состоит из одной строки, на которой находится целое число n ($1 \leqslant n \leqslant 60$). После последнего теста следует строка, содержащая $n = 0$, её не надо обрабатывать.

Формат выходных данных

Для каждого теста выведите на отдельной строке количество абстрактных танцев с n шагами. Ответ должен быть выведен по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
1 2 3 4 5 0	1 2 5 14 42

Задача C. Каталанские комбинаторные объекты

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

Андрей очень любит числа Каталана. А еще Андрей очень любит шутить.

Андрей — автор многих задач и придумал немало контестов для разных сборов. Для многих контестов он делал задачи, в которых на ввод подаётся одно число, а ответ для вводов 0, 1, 2, 3, 4, и 5 равны, соответственно, 1, 1, 2, 5, 14 и 42. А вот ответы для других входных данных отличаются от соответствующих чисел Каталана.

Андрей сделал уже так много контестов, что ему тяжело придумывать задачи с таким свойством. Он решил автоматизировать процесс придумывания таких задач. Хороший пример возможных подобных задач — подсчет комбинаторных объектов определенной структуры. Андрей придумал, каким он хочет видеть k — ответ на задачу для ввода 6, и хочет придумать такой тип комбинаторных объектов, чтобы существовало 1, 1, 2, 5, 14, 42, k объектов этого типа для веса 0, 1, 2, 3, 4, 5, 6, соответственно.

Андрей использует следующие способы конструирования комбинаторных объектов.

Базовое множество B состоит из одного объекта u с весом 1. Каждый сконструированный объект x имеет некоторый вес $w(x)$. Если объект сконструирован из одного или нескольких других объектов, его вес равен сумме весов этих объектов.

Пусть X задаёт некоторое множество комбинаторных объектов. Рассмотрим следующие способы создать новые множества объектов.

Множество $L(X)$ состоит из всех возможных списков конечной длины, каждый элемент которых имеет положительный вес и принадлежит множеству X. Например, $L(B)$ состоит из списков $[]$, $[u]$, $[u, u]$, $[u, u, u]$, и так далее. Аналогично, $L(L(B))$ состоит из $[]$, $[[u]]$, $[[u], [u]]$, $[[u, u], [u]]$, $[[u], [u, u]]$, и так далее. Обратите внимание, последние два списка различны, поскольку для списка важен порядок элементов в нем. Также обратите внимание, что $[[]]$ не является корректным списком в $L(L(B))$, поскольку только объекты положительного веса разрешаются в качестве элементов списков, а $[]$ имеет вес 0.

Множество $S(X)$ содержит все возможные мультимножества конечного размера, каждый элемент которых имеет положительный вес и принадлежит X. Например, $S(B)$ состоит из мультимножеств $\{\}$, $\{u\}$, $\{u, u\}$, $\{u, u, u\}$, и так далее. Еще один пример: $S(L(B))$ содержит, например, мультимножества $\{[u]\}$, $\{[u], [u]\}$. Обратите внимание, что мультимножество может содержать несколько равных объектов. Заметьте, что в отличие от списков для мультимножеств не важен порядок элементов, поэтому мультимножество $\{[u], [u, u]\}$ совпадает с мультимножеством $\{[u, u], [u]\}$.

Множество $C(X)$ состоит из всех возможных циклов конечной длины, каждый элемент которых имеет положительный вес и принадлежит X. Два цикла считаются равными, если они получаются друг из друга циклическим сдвигом. Например, $C(L(B))$ содержит цикл $([u], [u, u], [u, u, u])$. Заметьте, что этот цикл совпадате с циклом $([u, u], [u, u, u], [u])$, но отличается от цикла $([u, u, u], [u, u], [u])$.

Вес списка, мультимножества или цикла равен сумме весов его элементов, например, вес $([u], [u, u], [u, u, u])$ равен 6.

Наконец, последний рассматриваемый способ создания нового типа комбинаторных объектов — пара. Если X и Y — множества комбинаторных объектов, то $P(X, Y)$ представляет собой множество упорядоченных пар объектов, где первый компонент взят из X, а второй — из Y. Например, $P(S(B), L(B))$ содержит в качестве элементов $\langle\{u, u\}, [u, u, u]\rangle$ и $\langle\{\}, [u]\rangle$. Обратите внимание, что в отличие от списков, мультимножеств и циклов, пары могут содержать компоненты нулевого веса.

Дано k, постройте описание множества комбинаторных объектов, которое содержит 1 элемент веса 0, 1 элемент веса 1, 2 элемента веса 2, 5 элементов веса 3, 14 элементов веса 4, 42 элемента веса 5 и k элементов веса 6.

Формат входных данных

На вход подаётся несколько тестов.

Каждый тест содержит одну строку, на которой находится целое число k ($120 \leqslant k \leqslant 140$).

Последняя строка, которую не требуется обрабатывать, содержит число $n = 0$.

Формат выходных данных

Для каждого теста выведите описание множества комбинаторных объектов, удовлетворяющего условию. Не выводите пробелы. Длина описания не должна превышать 2000.

Пример

стандартный ввод
125 0
стандартный вывод
P(L(P(P(P(P(P(B,L(B)),P(B,L(B))),L(L(B))),P(P(B,L(B)),L(L(B)))),P(L (P(B,B)),L(P(B,B))))),L(L(B)))

Задача D. Раскраски тетраэдра

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

В игре «Тусовый Блиц» недавно появился новый тип раунда. Параметром раунда является число k, а на экране изображен правильный тетраэдр. Каждый раунд игроку необходимо раскрасить каждую из вершин тетраэдра в один из k цветов. При этом, игрок проигрывает, если очередную раскраску можно получить из какой-либо уже использованной путем вращений тетраэдра в пространстве.

Какое максимальное количество раундов нового типа мы сможем выиграть в «Тусовом Блице»?

Формат входных данных

В единственной строке входного файла содержится число k — количество цветов, которые можно использовать для раскраски ($1 \leqslant k \leqslant 1000$).

Формат выходных данных

Выведите единственное число — искомое число раундов.

Пример

стандартный ввод	стандартный вывод
1	1

Задача E. XORing

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

У вас есть 2^n ручек. Они раскрашены в c цветов. Давайте разложим эти ручки в последовательность и пронумеруем их числами от 0 до $2^n - 1$.

Рассмотрим множество чисел $A\,(0 \leqslant x < 2^n,\ x \in A)$. К последовательности ручек можно применить следующую операцию: ручка из позиции i перемещается на позицию $i \oplus x$, $x \in A$ (здесь $\oplus$ означает побитовое исключающее или, x один и тот же для всех i). Давайте назовем такую операцию *ксорирование*.

Две последовательности цветов ручек считаются одинаковыми, если одна из них может быть получена из другой последовательностью операций ксорирования (несколько раз, если необходимо).

Ваша задача состоит в том, чтобы посчитать количество различных последовательностей цветов ручек.

Формат входных данных

Первая строчка содержит три целых числа n, c, m ($0 \leqslant n \leqslant 60$, $1 \leqslant c \leqslant 10^9$, $0 \leqslant m \leqslant 10^5$). Количество ручек равно 2^n, количество цветов равно c, и мощность множества A равна m. Следующие m строк содержат элементы A, по одному числу на каждой строке. Все числа различны.

Формат выходных данных

Выведите количество различных последовательностей ручек по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 1 1	10
3 2 5 4 7 3 2 6	46

Задача F. Тороидальные билеты

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

На планете Эйсиэм пассажирские билеты для нового типа общественного транспорта имеют форму тора.

Каждый тор делается из черного резинового прямоугольника, состоящего из $N \times M$ квадратов. Несколько квадратов помечены белым цветом, таким образом закодировав место отправления и прибытия на этом билете.

Когда пассажир покупает билет, автомат по продаже билетов берет резиновый листок, помечает некоторые квадраты белым и отдает его пассажиру. Затем, пассажир должен самостоятельно склеить билет.

Билет должен быть склеен следующим способом. Сначала, две его стороны большей длины склеиваются вместе, образуя цилиндр. Затем, основания цилиндра, длина каждого из которых равна меньшей стороне оригинального прямоугольника, склеиваются вместе. Их необходимо склеить таким образом, чтобы клетки, стороны которых склеиваются, изначально принадлежали одной строке прямоугольника.

Получившийся тор и является билетом.

Заметьте, что если начальный билет был квадратным, существует два топологически различных способв склеить тор из резиновой заготовки.

Материал, из которого делаются билеты, настолько идеален, и качество склеивания настолько хорошо, что никто не может найти шов, и это приводит к некоторым проблемам.

Во-первых, один и тот же тор может получиться из разных заготовок. Более того, одна и та же заготовка может быть склеена в торы, которые выглядят по-разному.

Теперь, транспортные компании Эйсиэма интересуются, сколько различных маршрутов они могут организовать, чтобы выполнялись следующие свойства:

- билеты для разных маршрутов представляются разными торами;
- если одна прямоугольная заготовка была помечена для использования в качестве тора для одного маршрута, она не может быть использована, чтобы сделать тор для другого маршрута.

Помогите им посчитать количество маршрутов, которые они могут организовать.

Формат входных данных

Первая строка содержит два целых числа N и M ($1 \leqslant N, M \leqslant 20$).

Формат выходных данных

Выведите количество маршрутов, которые могут организовать транспортные компании Эйсиэма.

Примеры

стандартный ввод	стандартный вывод
2 2	6
2 3	13