

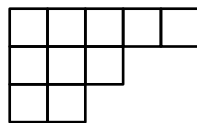
Задача А. Увидеть Юнга и умереть

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

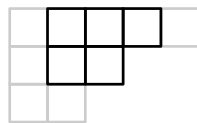
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа n на слагаемые представляет собой сумму вида $n = m_1 + m_2 + \dots + m_k$, где $m_1 \geq m_2 \geq \dots \geq m_k$.

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 ($1 \leq n \leq 100$).

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 В. 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 ($n \times m \leq 500$, $k \leq 1000$).

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 С. 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 ($1 \leq n \leq 40$, $1 \leq k \leq 1000$).

Далее следует n строк. В i -й из них содержится n целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($a_{i,j} \in \{0, 1\}$). 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 $1 \leq i, j \leq n$.

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 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$.

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

Формат выходных данных

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

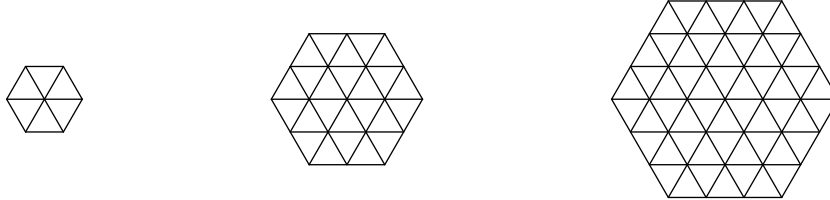
Пример

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

Задача Е. Шестиугольные домино-ромбы

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

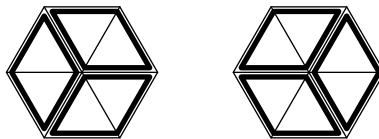
Правильный шестиугольник со стороной длины n разделен на $6n^2$ единичных треугольников.



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 ($1 \leq n \leq 7$).

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 работа 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, \dots, 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, \dots, a_nx , не имеет значения.

Формат входных данных

В первой строке находится число n — количество требуемых значений ($1 \leq n \leq 41$). Во второй строке находится n чисел a_i ($2 \leq a_i \leq 42$). 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 ($1 \leq N \leq 30$). Следующие N строк содержат по три целых числа a_i, b_i и c_i ($1 \leq a_i, b_i, c_i \leq 10^6$) — 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	5
3 1 3	2
2 2 2	1 1 3 3
1 2 1	3 1 1 2

Задача Н. 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 — подмаска 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 ($1 \leq n \leq 20$).

В следующей строке даны 2^n чисел, i -е из них — значение $P[i]$ ($-10^9 \leq P[i] \leq 10^9$) (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