

Задача А. Коробки

Имя входного файла: `boxes.in`
Имя выходного файла: `boxes.out`
Ограничение по времени: 2 секунды
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

<code>boxes.in</code>	<code>boxes.out</code>
3	5
3 1 3	2
2 2 2	1 1 3 3
1 2 1	3 1 1 2

Задача В. Японский компьютер

Имя входного файла: `computer.in`
Имя выходного файла: `computer.out`
Ограничение по времени: 2 секунды
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 програм-

ма 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

computer.in	computer.out
3 3 5 18	5 1<<1 1<<4 1+2 2+3 2+16
1 29	4 1<<1 1<<5 1+2 32-3
4 12 19 41 42	8 1<<1 1<<4 1+2 3<<2 3+16 19<<1 3+38 1+41

Формат выходных данных

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

paths.in	paths.out
0	1
1	4
2	12

Задача С. Пути на доске

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

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входных данных

В первой строке входного файла задано целое число n ($0 \leq n \leq 22$).