

Задача А. Хомяк

Ограничение по времени: 1 секунда

Ограничение по памяти: 32 Мб

Полина обзавелась хомяком и прочитала несколько книг по зоологии о том, как нужно правильно кормить хомяков.

Оказалось, что хомяка достаточно кормить один раз в день строго в определенное время. Однако, в разные дни это время может быть разным. После чего был составлен индивидуальный график питания хомяка, и стало известно точное время кормления с точностью до минуты для каждого дня.

Память у Полины так себе. Поэтому чтобы не забывать вовремя покормить своего питомца каждый день, как только Полина покормила хомяка, она устанавливает таймер, который срабатывает на следующий день ровно в тот момент, когда пора снова кормить.

Полине сложно каждый раз вычислять, на какое время нужно установить таймер. Требуется ей помочь: написать программу, которая по времени кормления сегодня и времени кормления на следующий день посчитает, на какое время нужно установить таймер.

Входные данные

Первая строка входного файла INPUT.TXT содержит время кормления хомяка сегодня в часах и минутах в формате «HH:MM» (время может быть от 00:00 до 23:59 включительно).

Вторая строка содержит время кормления завтра в таком же формате.

Выходные данные

В выходной файл OUTPUT.TXT выведите время, которое пройдет между двумя кормлениями хомяка, в том же формате, что и во входных данных.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	22:00 07:00	09:00
2	12:32 21:41	33:09

Система оценки

Решения, работающие только для времен формата «HH:00» будут оцениваться в 40 баллов.

Задача В. Числовая лесенка

Ограничение по времени: 1 секунда

Ограничение по памяти: 32 Мб

Выпишем числа таким образом, чтобы они образовали лесенку из ряда натуральных чисел. На первой строке будет стоять одно число 1, во второй строке будут два числа 2 и 3 и так далее. В результате получим следующую последовательность из чисел:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
...
```

После этого уберем все числа на каждой строке, кроме первых K . Если в строке меньше K чисел, то оставим их все.

По заданным числам A , B и K требуется вывести строки с A -й по B -ю, которые должны получиться в результате построения нашей лесенки.

Входные данные

Входной файл INPUT.TXT содержит три строки: первая содержит число A , вторая содержит число B , в третьей записано число K ($1 \leq A \leq B \leq 10^9$, $B-A \leq 100$, $1 \leq K \leq 100$).

Выходные данные

В выходной файл OUTPUT.TXT выведите строки с A -й по B -ю, которые должны получиться в результате описанного в условии задачи алгоритма. Числа в строках следует разделять пробелами.

Пример

INPUT.TXT	OUTPUT.TXT
1	1
5	2 3
2	4 5
	7 8
	11 12

Система оценки

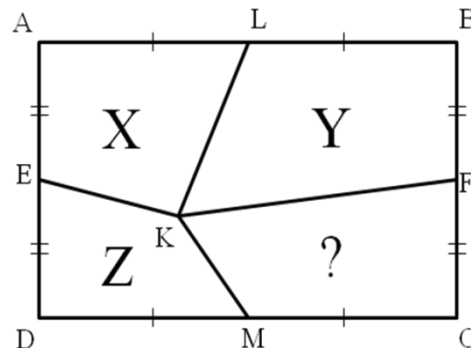
Решения, работающие только для $B \leq 100$, будут оцениваться в 50 баллов.

Задача С. Площадь четырёхугольника

Ограничение по времени: 1 секунда

Ограничение по памяти: 32 Мб

Прямоугольник $ABCD$ со сторонами, параллельными осям координат, разбили на четыре фигуры следующим образом: некоторую точку K , расположенную строго внутри данного прямоугольника, соединили отрезками KE , KL , KF и KM с серединами сторон AD , AB , BC и DC соответственно. Так были получены четыре четырёхугольника $ALKE$, $LBFK$, $EKMD$ и $KFCM$.



По известному значению площадей трёх полученных четырёхугольников требуется определить значение площади четвертого.

Входные данные

Единственная строка входного файла `INPUT.TXT` содержит три вещественных числа X , Y и Z – площади четырёхугольников $ALKE$, $LBFK$ и $EKMD$. Гарантируется, что длины сторон прямоугольника $ABCD$ положительны и не превышают 100.

Выходные данные

В выходной файл `OUTPUT.TXT` выведите вещественное число – значение искомой площади четырёхугольника $KFCM$ с точностью 10^{-6} .

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	13 13 13	13
2	2 1.6 3.2	2.800000

Система оценки

Решения, работающие верно только в случае, когда точка K совпадает с центром прямоугольника $ABCD$, будут оцениваться в 10 баллов.

Решения, работающие верно только для четырёхугольников с целочисленными координатами вершин, будут оцениваться в 50 баллов.

Задача D. N-гиперпрямоугольник

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 Мб

N-гиперпрямоугольник – это обобщение прямоугольника на N -мерном декартовом пространстве. Известно, что у N -гиперпрямоугольника всего 2^N вершин (0-мерных граней) и $N \times 2^{N-1}$ ребер (1-мерных граней). Например, для $N=2$ мы получим обычный прямоугольник с 4 вершинами и 4 ребрами (сторонами); а при $N=3$ это будет прямоугольный параллелепипед с 8 вершинами и 12 ребрами.

Задан N -гиперпрямоугольник с целочисленными координатами вершин и с ребрами, параллельными осям координат. Одна из его вершин находится в начале координат. Координата другой вершины (w_1, w_2, \dots, w_N) определяет его размеры и является самой дальней координатой от начала координат.

Требуется осуществить обход всех вершин заданного N -гиперпрямоугольника по ребрам таким образом, чтобы при этом посетить каждую вершину ровно один раз. Формально, если (x_1, x_2, \dots, x_N) и (y_1, y_2, \dots, y_N) – две последовательные вершины в маршруте обхода, то должно выполняться следующее:

$$\begin{cases} \sum_{i=1}^N |(x_i - w_i)x_i| + |(y_i - w_i)y_i| = 0 \\ \sum_{i=1}^N \left| \frac{x_i - y_i}{w_i} \right| = 1 \end{cases}$$

Входные данные

Первая строка входного файла INPUT.TXT содержит целое число N ($1 \leq N \leq 16$) – размерность N -гиперпрямоугольника. Во второй строке записаны N целых чисел w_1, w_2, \dots, w_N ($1 \leq w_i \leq 100$) – координаты дальней вершины N -гиперпрямоугольника от начала координат.

Выходные данные

В выходной файл OUTPUT.TXT выведите 2^N строк по N целых чисел в строке – последовательность обхода вершин в требуемом порядке. Если существует несколько возможных обходов, выведите любой.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	1 3	0 3
2	2 5 7	5 0 0 0 0 7 5 7

Система оценки

Решения, работающие верно только для $N \leq 4$, будут оцениваться в 25 баллов.

Задача Е. Задача из ЕГЭ

Ограничение по времени: 3 секунды

Ограничение по памяти: 256 Мб

Задан массив целых чисел $A[1..N]$, состоящий из N элементов.

Требуется определить количество таких троек чисел в массиве, чтобы расстояние между любой парой чисел из тройки было не менее K , а произведение всех трёх чисел было кратно натуральному числу M .

Расстояние между элементами массива A_i и A_j равно значению $|i-j|$.

Входные данные

Первая строка входного файла INPUT.TXT содержит три целых числа: N , K и M ($1 \leq N, K, M \leq 10^6$). При этом гарантируется, что число M представляет собой произведение двух простых чисел.

Вторая строка входных данных содержит элементы массива A_i – целые числа, не превышающие 10^7 по абсолютной величине.

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Пример

INPUT.TXT	OUTPUT.TXT
6 2 10 24 13 25 60 88 3	4

Пояснение к примеру

В исходном массиве можно выделить 4 тройки, удовлетворяющие условию задачи: (24, 25, 88), (24, 25, 3), (24, 60, 3) и (13, 60, 3).

Система оценки

Решения, работающие только для $N \leq 100$, будут оцениваться в 50 баллов.

Задача. Тетрис 3D

Ограничение по времени: 3 секунды

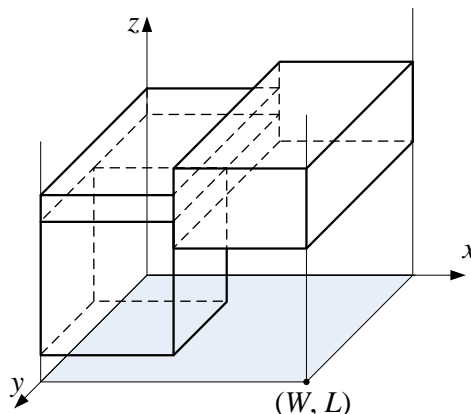
Ограничение по памяти: 256 Мб

Всем известна игра «Тетрис». Рассмотрим её трёхмерную версию, где в качестве падающих блоков будут выступать прямоугольные параллелепипеды, грани которых параллельны осям координат, а вершины имеют целочисленные координаты. Разумеется, при этом никакие два блока не могут пересекаться, но могут касаться.

Стакан, в который падают блоки, имеет вид прямоугольного параллелепипеда, основание которого имеет размер $W \times L$. При этом один из углов основания находится в начале системы координат, а другой имеет координаты (W, L) . Стены стакана параллельны осям координат.

Нам представлен некоторый момент игры, в которой положение N блоков зафиксировано. Известно, что блоки, полностью закрывающие горизонт ненулевого объема уничтожаются, причем за наименьшее количество уничтоженных при этом блоков игрок получает большее количество очков.

Требуется написать программу, которая позволяет выбрать минимальное число блоков, которые образуют перекрытие по горизонтали, либо определить, что этого сделать невозможно.



Входные данные

В первой строке входного файла INPUT.TXT указаны три целых числа: N – количество блоков ($1 \leq N \leq 10^5$) и размеры стакана W и L ($1 \leq W, L \leq 10^4$). Каждая из последующих N строк описывает один блок, определяемый координатами противоположных углов: (x_1, y_1, z_1) и (x_2, y_2, z_2) , при этом $0 \leq x_1 < x_2 \leq W$, $0 \leq y_1 < y_2 \leq L$, $0 \leq z_1 < z_2 \leq 10^9$. Все числа во входном файле целые и разделяются пробелами или переводами строк.

Гарантируется, что блоки не пересекаются друг с другом.

Выходные данные

Первая строка выходного файла OUTPUT.TXT должна содержать либо слово «YES», если искомый набор блоков существует, иначе – слово «NO». В первом случае во второй строке следует вывести минимальное число блоков, образующих перекрытие по горизонтали, а в третьей строке – номера этих блоков согласно порядку, в котором они перечислены во входных данных.

Если возможно несколько минимальных наборов блоков, выведите любой из них.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3 10 10 0 0 6 5 10 7 5 0 5 10 10 8 0 5 1 5 10 6	YES 2 1 2
2	2 10 10 0 0 6 5 10 7 5 0 5 9 10 8	NO

Система оценки

Решения, работающие верно только в случае, когда все числа во входных данных не превышают 100, будут оцениваться в 40 баллов.