1. Треугольники

Хорошо известно, что три отрезка могут образовывать треугольник тогда и только тогда, когда сумма длин любых двух из них строго больше длины третьего.

Пусть имеет N длин отрезков. Можете найти по крайней мере один набор из трех отрезков из этого списка, которые образуют треугольник.

Ввод

Первая строка ввода содержит единственное целое число N (3≤N≤ 1000). Следующие N строк содержат длины отрезков. Длина любого из отрезков - это целое число от 1 до 10500.

Вывод

Напишите на выходе длины отрезков, удовлетворяющих неравенству треугольника, - три числа, разделенные пробелами. Напишите три нуля, если таких трех отрезков в списке нет.

Примеры

Ввод	Вывод
7 1 2 6 4 8 100 73	8 4 6

Ввод	Вывод
7 1 2 1 2 1 2 1	2 1 2 2 1 2

Ввод	Вывод
3 1 2 3	0 0 0

2. Бега кузнечиков

Некий молодой предприниматель решил организовать бега кузнечиков.

Он предположил, что будет N разделенных беговых дорожек, каждая длиной L, по которым будут скакать кузнечики наперегонки. Кузнечики будут занумерованы числами от 1 до K и тот кузнечик, который придет первым к финишу будет считается победителем забега. Для удобства зрителей дорожка будет разделена на m равноудаленных отметок. В результате на дорожке получится сетка s из пронумерованных отметок с номерами от 0 (старт) до F (финиш).

Предприниматель обратился к математику и тот предположил, что кузнечик с номером i $(i=0,1,2,\ldots,N-1)$, двигаясь прямолинейно без остановок, в момент времени t_j $(j=0,1,2,\ldots,T)$ будет совершать прыжок на расстояние, вычисляемое по формуле:

$$l_j=2\cdot\left|0{,}3\cdot\cos\left(\frac{(t_j-5)^2}{i+3}\right)+0{,}4\cdot\sin\left(\frac{(t_j-3)^2}{2}\right)\right| .$$

Чтобы проверить адекватность выбора модели, предложенной математиком, предприниматель обратился к программисту с просьбой написать программу, которая определяла бы положение каждого кузнечика с заданным номером i в любой момент времени t_j $\left(2 \leq t_j \leq T\right)$ в сетке отметок, указывая ближайшие номера отметок, между которыми находится кузнечик в указанный момент времени. Считается, что время измеряется по таймеру равными интервалами с задаваемым шагом h_t на шкале $[0, T]$.

Ввод

Пять строк с чисел, которые задают соответствующие значения величин:

K – количество кузнечиков; L – длина дистанции; m – количество отметок на сетке s ; T – величина границы шкалы таймера; h_t интервал временного шага.

Вывод

К строк из трёх чисел, разделенных пробелом и означающих i – номер кузнечика, j – номер левой границы сегмента сетки, $j+1$ – номер правой границы сегмента сетки, в которых находится i - кузнечик в момент времени, соответствующий величине границы шкалы таймера T . Если i – кузнечик преодолевает расстояние L , то для него после номера выводится строка «Finished».

Примеры

ввод	вывод
3 15 7 2 0.1	0 4 5 1 3 4 2 2 3

ввод	вывод
2 10 7 2.5 0.1	0 Finished 1 5 6

3. Гардероб

Студенты идут после занятий к гардеробу, у каждого - свой (ровно один) номерок. Часть студентов честно встаёт в конец очереди, а часть - игнорирует очередь (встаёт в её начало). Время от времени, один из студентов (стоящий в начале очереди) получает свою верхнюю одежду и уходит.

Вся ситуация представлена в виде событий (набора строк) вида "+ x y " и "–", где первое событие означает, что в очередь пришёл студент с номерком x (и встал в конец очереди, если $y=1$ и в начало, если $y=0$), а второе - что стоящий первым студент получил свою верхнюю одежду и ушёл из очереди.

Входные данные: число n и ровно $2n$ строк указанного вида (причём ровно n из них являются строками первого вида и ровно n строками второго). При этом гарантируется, что все x различны, и что событие второго вида происходит, только когда очередь не пуста.

Выходные данные: строка, содержащая последовательность номерков, в том порядке, в котором студенты получат свою верхнюю одежду (то есть сначала - номерок получившего одежду первым, затем - вторым, и т.д.).

Пример

Ввод	Вывод
3 + 1 1 + 2 0 – + 3 0 – –	2 3 1

Примечание: Возможные значения n : от 1 до 10^5 .

4. Отметки

Группа студентов ходит на пары по разным предметам и получает там отметки.

При этом получение очередной отметки описывается строкой вида "Name Subject X", где Name - имя студента (считается, что имена разных студентов различны) - строка из английских символов 'a'-'z', 'A'-'Z', не содержащая пробелов, длиной не более 20 символов; Subject - название учебного предмета (названия разных предметов различны) - строка с аналогичными первой условиями; X - число от 2 до 5, означающее полученную отметку.

Входные данные: число n - количестве полученных отметок, далее - ровно n строк указанного вида.

Выходные данные: список всех учеников в алфавитном порядке с их средней отметкой (по всем полученным данным учеником отметкам), список всех предметов в алфавитном порядке с средней отметкой учеников по данному предмету.

Средняя оценка округляется с точностью до двух знаков после запятой.
Пример

Ввод	**Вывод**
6 Ivan Geometry 5 Ivan Math 4 Anton Geometry 3 Anton Geometry 4 Anton Math 5 Ivan Informatics 5	Anton 4.00 Ivan 4.67 Geometry 4.00 Informatics 5.00 Math 4.50

Примечание: Возможные значения n : от 1 до 10^5 .

5. Числа Шелдона.

Число 73 обладает примечательным свойством. 73 это 21-е простое число. Переставив в этой паре числа местами получим 37 и 12. Переставив в последнем числе цифры получим 21, которое равно произведению чисел 7 и 3, которые являются цифрами исходного числа.

В двоичной записи число является палиндромом, а именно, может быть прочитано одинаково как в прямом, так и в обратном порядке.

С другой стороны, двоичное представление числа 73 довольно примечательно: это 1, за которым следуют 2 нуля, за которым следует 1 единица, за которым следуют 2 нуля, а затем 1 единица.

Это интересный шаблон, который мы можем обобщить: N единиц, за которыми следуют M нулей, за которыми следуют N единиц, а затем M нули и т. д., и заканчивающиеся либо N единицами, либо M нулями.

Для 73, N равно 1, M равно 2, и существует 5 цепочек равных символов.

При $N=2$, $M=1$ и 4 цепочек равных символов, мы будем иметь строку 110110 , которая является двоичным представлением числа $54_{10}=110110_2$.

Представим понятие числа Шелдона: положительное целое число, двоичное представление которого соответствует шаблону $ABABAB\ldots ABA$ или шаблон $ABABAB\ldots AB$, где все вхождения A представляют собой строку с N вхождениями бита 1 и где все вхождения B представляют собой строку с M вхождениями бита 0 $(N>0 \text{ и } M>0)$. Кроме того, в представлении должно быть хотя бы одно вхождение строки A (но число вхождений строки B может быть равно нулю).

Вот примеры некоторых чисел Шелдона: 21, 45, 1755, 1984, 2015.

Ясно, что существует бесконечное количество чисел Шелдона, но являются ли они более плотными или менее плотными, чем простые числа?

Ваша задача - написать программу, которая, получая два натуральных числа, вычисляет количество двоичных палиндромов чисел Шелдона, которые существуют в диапазоне, определяемом заданными числами.

Ввод

Входной файл содержит несколько тестовых примеров, каждый из которых описан ниже.

Вход содержит одну строку с двумя целыми числами, разделенными пробелом, X и Y .

Ограничения:

$0\leq X\leq Y\leq 2^{63}$

Вывод

Для каждого тестового примера вывод содержит одну строку с одним числом, представляющим количество чисел-палиндромов Шелдона, которые больше или равны X и меньше или равны Y.

Если таких чисел нет, то выводится строка “NO”.

Примеры ввода:

1

Ввод	Вывод
1 10	5

2

Ввод	Вывод
70 75	1

3

Ввод	Вывод
100 160	5

Примечание к примерам:

Объяснение вывода 1. Все нечетные числа от 1 до 10 - это палиндромы числа Шелдона.

Объяснение вывода 2. 73 - единственное число-палиндром Шелдона в этом диапазоне.