

## Содержание

<b>Задачи на 3</b>	<b>2</b>
Задача 13A. Малыш и Карлсон [0.1 sec, 256 mb]	2
Задача 13B. План эвакуации [0.05 sec, 256 mb]	3
Задача 13C. Платные дороги [0.5 sec, 256 mb]	5
Задача 13D. К минимумов на отрезке [3.0 sec, 256 mb]	6
<b>Задачи на 4</b>	<b>7</b>
Задача 13E. Сервера [0.7 sec, 256 mb]	7
Задача 13F. Расстояние между многоугольниками [0.4 sec, 256 mb]	8
Задача 13G. Рефрен [0.5 sec, 256 mb]	9
Задача 13H. $k$ паросочетаний [0.4 sec, 256 mb]	10
<b>Задачи на 5</b>	<b>11</b>
Задача 13I. Великая стена [0.5 sec, 256 mb]	11
Задача 13J. Дуэль [0.2 sec, 256 mb]	12
Задача 13K. Illumination [0.7 sec, 256 mb]	13
Задача 13L. Вирусы [0.05 sec, 256 mb]	14
Задача 13M. Общая подпоследовательность [3 sec, 256 mb]	15
Задача 13N. XOR [5 sec, 256 mb]	16
<b>Гробы</b>	<b>17</b>
Задача 13O. Композиция многочленов [2 sec, 256 mb]	17
Задача 13P. Выстрелы по стенам [0.4 sec, 256 mb]	18

---

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

## Задачи на 3

### Задача 13А. Малыш и Карлсон [0.1 сек, 256 mb]

На свой День рождения Малыш позвал своего лучшего друга Карлсона. Мама испекла его любимый пирог прямоугольной формы  $a \times b \times c$  сантиметров. Карлсон знает, что у Малыша еще есть килограмм колбасы. Чтобы заполучить ее, он предложил поиграть следующим образом: они по очереди разрезают пирог на две ненулевые по объему прямоугольные части с целыми измерениями и съедают меньшую часть (в случае, когда части равные, можно съесть любую). Проигрывает тот, кто не может сделать хода (то есть когда размеры будут  $1 \times 1 \times 1$ ). Естественно, победителю достается колбаса.

Малыш настаивает на том, чтобы он ходил вторым.

Помогите Карлсону выяснить, сможет ли он выиграть, и если сможет — какой должен быть его первый ход для этого.

Считается, что Малыш всегда ходит оптимально.

#### Формат входных данных

Во входном файле содержится 3 целых числа  $a, b, c$  ( $1 \leq a, b, c \leq 5\,000$ ) — размеры пирога.

#### Формат выходных данных

В случае, если Карлсон не сможет выиграть у Малыша, выведите NO. В противном случае в первой строке выведите YES, во второй — размеры пирога после первого хода Карлсона в том же порядке, что и во входном файле.

#### Примеры

stdin	stdout
1 1 1	NO
2 1 1	YES 1 1 1

### Задача 13В. План эвакуации [0.05 сек, 256 mb]

В городе есть муниципальные здания и бомбоубежища, которые были специально построены для эвакуации служащих в случае ядерной войны. Каждое бомбоубежище имеет ограниченную вместительность по количеству людей, которые могут в нем находиться. В идеале все работники из одного муниципального здания должны были бы бежать к ближайшему бомбоубежищу. Однако, в таком случае, некоторые бомбоубежища могли бы переполниться, в то время как остальные остались бы наполовину пустыми.

Чтобы разрешить эту проблему Городской Совет разработал специальный план эвакуации. Вместо того, чтобы каждому служащему индивидуально приписать, в какое бомбоубежище он должен бежать, для каждого муниципального здания определили, сколько служащих из него в какое бомбоубежище должны бежать. Задача индивидуального распределения была переложена на внутреннее управление муниципальных зданий.

План эвакуации учитывает количество служащих в каждом здании — каждый служащий должен быть учтен в плане и в каждое бомбоубежище может быть направлено количество служащих, не превосходящее вместимости бомбоубежища.

Городской Совет заявляет, что их план эвакуации оптимален в том смысле, что суммарное время эвакуации всех служащих города минимально.

Мэр города, находящийся в постоянной конфронтации с Городским Советом, не слишком то верит этому заявлению. Поэтому он нанял Вас в качестве независимого эксперта для проверки плана эвакуации. Ваша задача состоит в том, чтобы либо убедиться в оптимальности плана Городского Совета, либо доказать обратное, представив в качестве доказательства другой план эвакуации с меньшим суммарным временем для эвакуации всех служащих.

Карта города может быть представлена в виде квадратной сетки. Расположение муниципальных зданий и бомбоубежищ задается парой целых чисел, а время эвакуации из муниципального здания с координатами  $(X_i, Y_i)$  в бомбоубежище с координатами  $(P_j, Q_j)$  составляет  $D_{ij} = |X_i - P_j| + |Y_i - Q_j| + 1$  минут.

#### Формат входных данных

Входной файл содержит описание карты города и плана эвакуации, предложенного Городским Советом. Первая строка входного файла содержит два целых числа  $N$  ( $1 \leq N \leq 100$ ) и  $M$  ( $1 \leq M \leq 100$ ), разделенных пробелом.  $N$  — число муниципальных зданий в городе (все они занумерованы числами от 1 до  $N$ ),  $M$  — число бомбоубежищ (все они занумерованы числами от 1 до  $M$ ).

Последующие  $N$  строк содержат описания муниципальных зданий. Каждая строка содержит целые числа  $X_i$ ,  $Y_i$  и  $B_i$ , разделенные пробелами, где  $X_i$ ,  $Y_i$  ( $-1000 \leq X_i, Y_i \leq 1000$ ) — координаты здания, а  $B_i$  ( $1 \leq B_i \leq 1000$ ) — число служащих в здании.

Описание бомбоубежищ содержится в последующих  $M$  строках. Каждая строка содержит целые числа  $P_j$ ,  $Q_j$  и  $C_j$ , разделенные пробелами, где  $P_j$ ,  $Q_j$  ( $-1000 \leq P_j, Q_j \leq 1000$ ) — координаты бомбоубежища, а  $C_j$  ( $1 \leq C_j \leq 1000$ ) — вместимость бомбоубежища.

В последующих  $N$  строках содержится описание плана эвакуации. Каждая строка представляет собой описание плана эвакуации для отдельного здания. План эвакуации из  $i$ -го здания состоит из  $M$  целых чисел  $E_{ij}$ , разделенных пробелами.  $E_{ij}$  ( $0 \leq E_{ij} \leq 10\,000$ ) — количество служащих, которые должны эвакуироваться из  $i$ -го здания в  $j$ -е бомбоубежище.

Гарантируется, что план, заданный во входном файле, корректен.

#### Формат выходных данных

Если план эвакуации Городского Совета оптимален, то выведите одно слово **OPTIMAL**. В противном случае выведите на первой строке слово **SUBOPTIMAL**, а в последующих  $N$  строках

выведите Ваш план эвакуации (более оптимальный) в том же формате, что и во входном файле. Ваш план не обязан быть оптимальным, но должен быть лучше плана Городского Совета.

**Пример**

stdin	stdout
3 4 -3 3 5 -2 2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 1 1 0 0 0 6 0 0 3 0 2	SUBOPTIMAL 3 0 1 1 0 0 6 0 0 4 0 1
3 4 -3 3 5 -2 2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 0 1 1 0 0 6 0 0 4 0 1	OPTIMAL

### Задача 13С. Платные дороги [0.5 sec, 256 mb]

Мэр одного большого города решил ввести плату за проезд по шоссе, проходящим в районе города, чтобы снизить объем транзитного транспорта. В районе города проходит  $n$  шоссе.

Но руководство области, в которой расположен город, воспротивилось планам мэра. Действительно — дальнбойщики представляют собой неплохой источник доходов для большого количества кафе и гостиниц в небольших городках.

В результате решили, что плата будет введена только на шоссе, которые *проходят через город*.

В городе используется развитая система метрополитена, всего в городе есть  $m$  станций метро. Решено было, что шоссе проходит через город, если либо одна из станций метро расположена непосредственно на шоссе, либо есть хотя бы одна станция с каждой стороны от шоссе.

Помогите теперь мэру определить, какие шоссе проходят через город.

#### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  — количество шоссе и количество станций метро, соответственно ( $1 \leq n, m \leq 100\,000$ ).

Следующие  $n$  строк описывают шоссе. Каждое шоссе описывается тремя целыми числами  $a$ ,  $b$  и  $c$  и представляет собой прямую на плоскости, задаваемую уравнением  $ax + by + c = 0$  ( $|a|, |b|, |c| \leq 10^9$ ).

Следующие  $m$  строк входного файла описывают станции метро. Каждая станция описывается двумя целыми числами  $x$  и  $y$  и представляет собой точку на плоскости с координатами  $(x, y)$  ( $|x|, |y| \leq 10^9$ ).

#### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — количество шоссе, которые проходят через город. Вторая строка должна содержать номера этих шоссе в возрастающем порядке. Шоссе нумеруются от 1 до  $n$  в порядке, в котором они описаны во входном файле.

#### Примеры

stdin	stdout
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4

### Задача 13D. К минимумов на отрезке [3.0 сек, 256 mb]

Дан массив  $a$  из  $n$  целых чисел и  $q$  запросов вида «вывести  $k$  первых чисел в отсортированной версии отрезка  $[l \dots r]$  нашего массива».

Пример:  $n = 7$ ,  $a = [6, 1, 5, 2, 4, 3, 1]$ ,  $l = 2$ ,  $r = 4$ ,  $k = 2$ . Отрезок  $[l \dots r] = [1, 5, 2]$ . Его отсортированная версия  $= [1, 2, 5]$ . Первые 2 числа  $= [1, 2]$ .

#### Формат входных данных

На первой строке число  $n$  ( $1 \leq n \leq 100\,000$ ).

На второй строке массив  $a$  ( $n$  целых чисел от 1 до  $10^9$ ).

На третьей строке количество запросов  $q$  ( $1 \leq q \leq 100\,000$ ).

Следующие  $q$  строк содержат тройки чисел  $l_i \ r_i \ k_i$

$1 \leq l_i \leq r_i \leq n$ ,  $1 \leq k_i \leq \min(r_i - l_i + 1, 10)$

#### Формат выходных данных

Для каждого из  $q$  запросов выведите ответ ( $k_i$  чисел) на отдельной строке. Числа внутри одного запроса нужно выводить в порядке возрастания. Для лучшего понимания условия и формата данных смотрите пример.

#### Система оценки

В этой задаче есть две группы тестов:

$n, q \leq 100\,000$ ,  $l_i \leq l_{i+1}$ ,  $r_i \leq r_{i+1}$ .

$n, q \leq 30\,000$ ,  $l_i$  и  $r_i$  произвольны.

#### Пример

stdin	stdout
7	1 1 2 3 4 5 6
6 1 5 2 4 3 1	1 2
4	2
1 7 7	1 3
2 4 2	
3 5 1	
5 7 2	

## Задачи на 4

### Задача 13Е. Сервера [0.7 сек, 256 mb]

*Спонсор этой задачи — всесибирская олимпиада 2009. Новосибирск – город мечты!*

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно  $N$  компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на  $K$  компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие  $K$  компьютеров, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными

#### Формат входных данных

В первой строке входного файла записано два целых числа  $N$  и  $K$  — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ( $1 \leq K \leq N \leq 2000$ ).

Все компьютеры в сети пронумерованы числами от 1 до  $N$  по порядку подключения.

Во второй строке записано одно целое число  $T_1$  — тариф обслуживания первого компьютера.

В следующих  $N - 1$  строках записано через пробел по два целых неотрицательных числа  $L_i, T_i$  — информация об остальных компьютерах в сети по порядку номеров.  $L_i$  — длина провода, соединяющего  $i$  — компьютер с соседним с меньшим номером,  $T_i$  — тариф обслуживания данного компьютера ( $2 \leq i \leq N$ ). Все  $L_i$  и  $T_i$  от 0 до  $10^6$ .

#### Формат выходных данных

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел  $K$  номеров компьютеров, на которые необходимо установить серверы. При существовании нескольких вариантов размещения разрешается вывести любой.

#### Пример

stdin	stdout
3 1 10 2 2 3 3	19 1
3 2 10 2 2 3 3	4 1 3

### Задача 13F. Расстояние между многоугольниками [0.4 sec, 256 mb]

Нужно найти расстояние между двумя выпуклыми непересекающимися многоугольниками.

#### Формат входных данных

Во входном файле содержатся описания двух многоугольников.

Многоугольник задается числом вершин —  $N$  ( $1 \leq N \leq 50\,000$ ). И координатами  $N$  вершин. Вершины даны в порядке обхода по часовой стрелке. Координаты целые и не превосходят  $10^9$  по модулю.

В обоих многоугольниках никакие три точки не лежат на одной прямой.

#### Формат выходных данных

Выведите одно вещественное число — расстояние между многоугольниками. Выводите ответ с максимально возможной точностью. Ваше решение будет считаться верным, если относительная или абсолютная погрешность ответа не превосходит  $10^{-10}$ .

#### Пример

stdin	stdout
4 0 0 0 1 1 1 1 0 3 2 0 2 2 4 0	1.00000000000000000000



### Задача 13G. Рефрен [0.5 sec, 256 mb]

Рассмотрим последовательность  $n$  целых чисел от 1 до  $m$ . Подпоследовательность подряд идущих чисел называется *рефреном*, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

#### Формат входных данных

Два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 150\,000$ ,  $1 \leq m \leq 10$ ).

Вторая строка содержит  $n$  целых чисел от 1 до  $m$ .

#### Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

#### Пример

stdin	stdout
9 3 1 2 1 2 1 3 1 2 1	9 3 1 2 1

#### Замечание

Эту задачу обязательно сдавать суффмассивом.

Даже если больше вам по душе деревья и автоматы.

### Задача 13Н. $k$ паросочетаний [0.4 сек, 256 mb]

Дан полный взвешенный двудольный граф с равным количеством вершин в долях. Требуется выбрать  $k$  максимальных попарно не пересекающихся паросочетаний так, чтобы их суммарный вес был минимален.

#### Формат входных данных

Первая строка входного файла содержит  $n$  и  $k$  — количество вершин в каждой из долей и количество паросочетаний ( $2 \leq n \leq 50$ ,  $1 \leq k \leq n$ ). Каждая из последующих  $n$  строк содержит по  $n$  чисел:  $C_{ij}$  — вес ребра, ведущего из  $i$ -й вершины левой доли в  $j$ -ю правой.

Все значения во входном файле неотрицательны и не превосходят  $10^6$ .

#### Формат выходных данных

В первую строку выходного файла выведите одно число — искомый суммарный вес паросочетаний. Следующие  $k$  строк должны содержать по  $n$  чисел — номера вершины, правой доли, соответствующие вершинам левой.

#### Примеры

stdin	stdout
3 2	6
1 2 1	1 2 3
1 1 2	3 1 2
2 1 1	

## Задачи на 5

### Задача 13I. Великая стена [0.5 sec, 256 mb]

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города  $A$  и  $B$ , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города  $A$  в город  $B$ .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника  $m \times n$ . В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

#### Формат входных данных

В первой строке входного файла содержатся числа  $m$  и  $n$  ( $1 \leq m, n \leq 50$ ). Во второй строке заданы числа  $k$  и  $l$ , где  $0 \leq k, l, k + l \leq mn - 2$ ,  $k$  — количество клеток, на которых расположены горы, а  $l$  — количество клеток, на которых можно строить стену. Естественно, что на горах строить стену нельзя. Следующие  $k$  строк содержат координаты клеток с горами  $x_i$  и  $y_i$ , а за ними следуют  $l$  строк, содержащие координаты клеток, на которых можно построить стену —  $x_j$  и  $y_j$ . Последние две строки содержат координаты городов  $A$  ( $x_A$  и  $y_A$ ) и  $B$  ( $x_B$  и  $y_B$ ) соответственно. Среди клеток, описанных в этих  $k + l + 2$  строках, нет двух совпадающих. Гарантируется, что  $1 \leq x_i, x_j, x_A, x_B \leq m$  и  $1 \leq y_i, y_j, y_A, y_B \leq n$ .

#### Формат выходных данных

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены  $F$ , которые необходимо построить. В последующих  $F$  строках необходимо вывести один из возможных вариантов застройки.

Если невозможно произвести требуемую застройку, то необходимо вывести в выходной файл единственное число  $-1$ .

#### Пример

stdin	stdout
5 5	3
3 8	3 1
3 2	1 3
2 4	3 3
3 4	
3 1	
1 3	
2 3	
3 3	
4 3	
5 3	
1 4	
1 5	
2 1	
5 5	

### Задача 13J. Дуэль [0.2 sec, 256 mb]

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт  $n$  деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

#### Формат входных данных

Во входном файле содержится одна строка, состоящая из символов '0' и '1' — схема аллеи. Деревья обозначаются символом '1', кусты — символом '0'. Длина строки не превосходит 100 000 символов.

#### Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

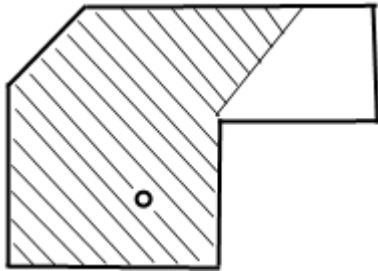
#### Примеры

stdin	stdout
101010101	4
101001	0

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **101010101**, **101010101**, **101010101** и **101010101**.

### Задача 13К. Illumination [0.7 sec, 256 mb]

In the room, shaped as the simple polygon with  $N$  vertices (i.e. closed polyline without self-intersections), an light source is put in the point  $(X_c, Y_c)$ . Find out an area of illuminated part of the room.



#### Формат входных данных

First line of the input file contains one integer  $T$  — number of the test cases ( $1 \leq T < 200$ ). First line of each test case contains two real numbers  $X_c$  and  $Y_c$  — coordinates of the light source. Next line contains one integer  $N$  — number of vertices of the polyline ( $3 \leq N \leq 5 \cdot 10^4$ ). Each of the next  $N$  lines contain coordinate of one vertices of the polyline — two real numbers  $X_i$  and  $Y_i$ . All coordinates are given with no more than 4 digits after the decimal point and does not exceed 1000 by absolute value. It is guaranteed that light source is strictly inside the room.

#### Формат выходных данных

For each test case print one integer — area of illuminated part of the room with precision  $10^{-2}$  or better.

#### Примеры

stdin	stdout
1 1 2 5 0 0 1 0 1 1 3 3 0 3	5.00

### Задача 13L. Вирусы [0.05 sec, 256 mb]

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее сегмент (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

#### Пример

Для множества кодов {011, 11, 0000} примером бесконечной безопасной последовательности является 010101... Для множества {01, 11, 00000} бесконечной безопасной последовательности не существует.

#### Формат входных данных

Первая строка входного файла `virus.in` содержит одно целое число  $N$ , равное количеству всех вирусных кодов. Каждая из следующих  $n$  строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30 000.

#### Формат выходных данных

Первая и единственная строка выходного файла должна содержать слово:

- TAK — если бесконечная, безопасная последовательность из нулей и единиц существует;
- NIE — в противном случае.

#### Пример

stdin	stdout
3 01 11 00000	NIE
3 011 11 0000	TAK

**Задача 13М. Общая подпоследовательность [3 сек, 256 mb]**

*Спонсор сегодняшней задачи — АРГО 2014 и Максим Ахмедов*

Вы играетесь с последовательностью  $n$  неотрицательных целых чисел. Цель игры – разбить последовательность на  $k + 1$  непустую часть ( $k + 1$  отрезок). Чтобы получить  $k + 1$  часть вы  $k$  раз делаете следующие шаги:

1. Выберите часть, содержащую больше одного элемента (изначально у вас ровно одна часть – вся последовательность).
2. Разбейте выбранную часть на две не пустых части.

При разбиении части на две вы получаете число очков, равное произведению сумм элементов в полученных новых частях. Задача – максимизировать итоговое число очков.

**Формат входных данных**

Первая строка содержит  $n$  и  $k$  ( $k + 1 \leq n$ ,  $2 \leq n \leq 100\,000$ ,  $1 \leq k \leq \min(n - 1, 200)$ ). Вторая строка содержит  $n$  неотрицательные целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^4$ ) – сама последовательность.

**Формат выходных данных**

На первой строке выведите  $m$  – максимальное число очков. На второй строке выведите  $k$  чисел от 1 до  $n - 1$  – позиции элементов, после которых нужно проводить разделения, чтобы в итоге набрать  $m$  очков. Если есть несколько таких последовательностей разделений, выведите любую.

**Примеры**

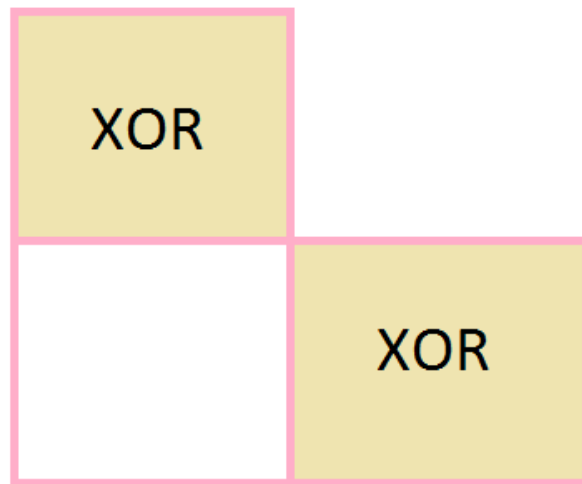
stdin	stdout
7 3	108
4 1 3 4 0 2 3	1 3 5

### Задача 13N. XOR [5 сек, 256 mb]

Даны  $n$  прямоугольников левый нижний угол которых находится в  $(0, 0)$ , а правый верхний в  $(i, A_i)$ . Гарантируется, что  $A_i$  не возрастают

Площадью симметрической разности двух прямоугольников назовем площадь, которая покрывается одним или другим прямоугольником, но не обоими сразу.

Нужно находить два прямоугольника  $i, j$ , такие что  $l \leq i < j \leq r$ , а площадь симметрической разности максимальна



#### Формат входных данных

В первой строке записано целое число  $n$  ( $2 \leq n \leq 100\,000$ ) — количество прямоугольников.

Во второй строке записано  $n$  чисел, для каждого  $i$  ( $1 \leq i \leq n$ )  $i$ -е обозначает  $A_i$  ( $1 \leq A_i \leq 10^9$ ,  $A_i \geq A_{i+1}$ ).

В следующей строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов.

В последующих  $q$  строках вводятся  $l_i, r_i$  ( $1 \leq l_i < r_i \leq n$ )

#### Формат выходных данных

На каждый запрос вывести максимальную площадь симметрической разности двух прямоугольников на отрезке.

#### Пример

stdin	stdout
7	36 19 8 27
10 9 9 8 8 6 6	
4	
2 7	
1 3	
4 5	
3 6	



## Гробы

### Задача 130. Композиция многочленов [2 sec, 256 mb]

Даны многочлены  $f(x)$ ,  $g(x)$ ,  $h(x)$  над полем  $\mathbb{Z}/2\mathbb{Z}$ .

Найдите многочлен  $f(g(x)) \bmod h(x)$ .

#### Формат входных данных

Три строки ввода содержат многочлены  $f$ ,  $g$  и  $h$ , по одному на строке. Каждый многочлен  $p$  описывается как  $n \ p_0 \ p_1 \ p_2 \ \dots \ p_n$  ( $1 \leq n \leq 4000$ ,  $p_i \in \{0, 1\}$  для всех  $i$ , а  $p_n = 1$ ). Сам многочлен  $p(x)$  в таком случае равен  $p_0 + p_1x + p_2x^2 + \dots + p_nx^n$ .

#### Формат выходных данных

Выведите ответ в том же формате.

Возможен ответ вида «0 0», обозначающий тождественный ноль.

#### Примеры

stdin	stdout
5 0 1 0 1 0 1 2 1 1 1 4 0 1 1 0 1	1 1 1
2 1 1 1 3 0 0 1 1 4 1 0 1 0 1	3 1 0 0 1

#### Замечание

Напомним несколько определений.

Поле  $\mathbb{Z}/2\mathbb{Z}$  — это множество из двух чисел 0 и 1, в котором результаты сложения, вычитания, умножения и деления — это остатки по модулю 2 от аналогичных результатов для обычных чисел.

Многочлен  $f(x)$  над этим полем — это объект вида  $f_n \cdot x^n + f_{n-1} \cdot x^{n-1} + \dots + f_1x + f_0$ , где коэффициенты  $f_n, \dots, f_0$  — числа из  $\mathbb{Z}/2\mathbb{Z}$ , и переменная  $x$  тоже может принимать значения из  $\mathbb{Z}/2\mathbb{Z}$ . Число  $n$  — максимальное такое, что  $f_n \neq 0$  — называется степенью многочлена  $p(x)$ .

Многочлены  $a(x) = \sum_k a_k x^k$  и  $b(x) = \sum_k b_k x^k$  равны, если для любого  $k$  числа  $a_k$  и  $b_k$  равны.

Сложение и вычитание многочленов определяются покомпонентно:  
 $a(x) \pm b(x) = \sum_k (a_k \pm b_k) \cdot x^k$ .

Произведение многочленов  $a(x)$  и  $b(x)$  определяется как  $c(x) = \sum_k c_k x^k$ , где  
 $c_s = \sum_{t=0}^s (a_t \cdot b_{s-t})$ .

Многочлены можно делить друг на друга. Если многочлен  $b(x)$  не является тождественным нулём, говорят, что  $a(x)/b(x) = q(x)$  и  $a(x) \bmod b(x) = r(x)$ , если  $q(x) \cdot b(x) + r(x) = a(x)$ , а степень  $r(x)$  строго меньше степени  $b(x)$ . Несложно показать, что многочлены  $q(x)$  и  $r(x)$  определены однозначно.

Композиция  $a(b(x))$  — это многочлен  $\sum_k a_k (b(x))^k$ , где степень многочлена определяется через произведение:  $(b(x))^0 = 1$ ,  $(b(x))^1 = b(x)$ ,  $(b(x))^p = b(x) \cdot (b(x))^{p-1}$  для  $p > 1$ . Коэффициенты композиции можно получить, если раскрыть скобки и сложить коэффициенты при одинаковых степенях переменной  $x$ .

### Задача 13Р. Выстрелы по стенам [0.4 sec, 256 mb]

Производится испытание нового пистолета, который может производить выстрелы с различными скоростями пуль. В некоторые моменты времени происходят выстрелы из начала координат с определенными горизонтальными скоростями, и в некоторые другие моменты строят стены на горизонтальной площадке — невырожденные отрезки, лежащие на прямых, не проходящих через начало координат. При этом стены могут пересекаться. Для обработки результатов эксперимента необходимо определить, сколько времени летела каждая пуля. Пуля летит с постоянной скоростью. Пуля останавливается сразу при касании стены.

#### Формат входных данных

В начале каждой строки написано одно из трех слов: **shot**, **wall** или **end**. Число строк не превышает 50 000. После слова **shot** следуют две координаты скорости пули. Скорость пули не может равняться нулю. После слова **wall** следуют четыре числа — координаты начала и конца стены. Слово **end** является признаком окончания набора входных данных. Все координаты являются целыми числами, по модулю не превосходящими 10 000. Все события записаны в хронологическом порядке, и интервалы времени между событиями больше, чем время, за которое строится стена, и чем время, за которое пуля пролетает расстояние до ближайшей стены или за границу испытательного полигона.

#### Формат выходных данных

Для каждого выстрела вывести на отдельной строке одно число — время, которое пролетит пуля, с точностью до  $10^{-6}$ . Если пуля не попадет ни в какую стену, то вместо числа вывести слово **Infinite**.

#### Пример

stdin	stdout
shot 1 0	Infinite
wall 1 0 0 1	0.50000000000000000000
shot 1 1	Infinite
shot -1 3	0.50000000000000000000
wall 1 0 -1 2	0.33333333333333333333
shot -1 3	2.00000000000000000000
wall 1 1 -1 1	0.05000000000000000000
shot -1 3	0.00200000000000000000
wall 2 3 2 -3	2.00000000000000000000
wall 3 -2 -3 -2	0.00100000000000000000
shot 1 -1	Infinite
shot 40 -39	0.00099950024987506247
shot 9999 -10000	1.00000000000000000000
shot -1 -1	0.50000000000000000000
shot -3000 -2000	1.00000000000000000000
shot -3001 -2000	0.90909090909090909091
shot -3000 -2001	0.43478260869565217391
shot 1 0	0.83333333333333333333
shot 1 1	2.00000000000000000000
wall -1 2 10 -10	3333.3333333333333333
shot -1 1	
shot 0 1	
shot 1 1	
shot 1 0	
shot 1 -1	
wall 0 -10000 -10000 0	
shot -2 -1	
end	