

## Задача 1. Классики

Входной файл	classics.in
Выходной файл	classics.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_1.pas



Маленькие девочки Оля и Катя любят гулять в парке рядом со своим домом. В этом парке есть асфальтовая дорожка, часть которой вымощена большими одинаковыми бетонными плитами. Ширина каждой плиты равна ширине дорожки, таким образом, дорожка образована просто последовательно лежащими плитами. До и после участка плит идёт асфальт.

Конечно, Катя и Оля всегда, когда идут по этому участку, прыгают по плитам. Катя за один прыжок прыгает ровно на  $A$  плит вперёд (т.е. если она стояла на второй плите, то за один прыжок она окажется на  $(2 + A)$ -й), Оля — на  $B$  (например, со второй плиты на  $(2 + B)$ -ю); назад девочки не прыгают. Первый прыжок обе девочки делают с асфальта на одну из первых  $A$  плит, при этом Катя может прыгнуть сразу на любую из первых  $A$  плит, а Оля — на любую из первых  $B$ . Аналогично, когда Катя оказывается на одной из последних  $A$  плит, она одним прыжком перепрыгивает на продолжение асфальта; Оля так же поступает, когда оказывается на одной из  $B$  последних плит. В частности, если плит меньше  $A$ , то Катя может перепрыгнуть их все за один прыжок (сразу с асфальта на асфальт), а может затратить и два прыжка; Оля может поступить так же, если плит меньше  $B$ .

Однажды девочки заспорили: сколько всего плит на этом участке? Катя помнит, что последний раз, когда они гуляли, она сделала  $N$  прыжков, а Оля —  $M$ . Помогите им определить, сколько всего могло быть плит.

### Формат входных данных

Во входном файле находятся четыре натуральных числа:  $A$ ,  $B$ ,  $N$  и  $M$ . Все числа не превосходят  $10^9$ .

### Формат выходных данных

В выходной файл выведите два натуральных числа: минимальное и максимальное ненулевое количество плит, которые могли быть на этом участке. Если ни одно ненулевое количество плит не соответствует условию, выведите два раза число 0.

### Пример

Входной файл	Выходной файл
2 3 2 2	1 3
2 3 4 3	5 7
2 3 100 3	0 0
4 6 3 2	5 11

*Примечание:* Тесты к этой задаче будут устроены следующим образом:

- тесты суммарной стоимостью 25 баллов будут иметь  $A = 2$ ,  $B = 3$ ,  $N \leq 1000$ ,  $M \leq 1000$ ;
- тесты суммарной стоимостью 25 баллов будут иметь  $A = 2$ ,  $B = 3$ , но или  $N$ , или  $M$ , или оба значения будут больше 1000;
- тесты суммарной стоимостью 25 баллов будут иметь  $N \leq 1000$ ,  $M \leq 1000$ , но или  $A \neq 2$ , или  $B \neq 3$ , или и то, и другое;
- тесты суммарной стоимостью 25 баллов не будут попадать ни под одно из указанных выше условий.

*Примечание:* В первом примере количество плит может быть от 1 до 3. Действительно, если плита была только одна, то обе девочки могли совершить два прыжка: первым прыжком прыгнуть с асфальта на плиту, вторым прыжком — с плиты на асфальт. Если плит было две, то обе девочки могли прыгнуть сначала, например, на первую плиту, а с неё сразу на асфальт после плит. Если плит было три, то обе девочки могли прыгнуть сразу на среднюю плиту, и с неё сразу на асфальт после плит. А если бы плит было четыре, то Катя уже не смогла бы их перепрыгнуть за два прыжка.

## Задача 2. Лотерея

Входной файл	lottery.in
Выходной файл	lottery.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_2.pas

В стране Грустьландии на железных дорогах все продаваемые билеты нумеруются последовательными натуральными числами. На этих железных дорогах проводится лотерея по номеру билета.

А именно, пассажир, покупая билет, может также приобрести за дополнительные деньги «сертификат участия в лотерее», в котором указывается номер купленного пассажиром билета; если пассажир не купил сертификат, то он не участвует в лотерее. В начале каждого месяца проводится розыгрыш главного приза за прошедший месяц. Розыгрыш происходит следующим образом.

Пусть за прошедший месяц были приобретены билеты с номерами от  $L$  до  $R$  включительно. Тогда выбирается случайное целое число  $X$  на отрезке  $[L, R]$ . После этого среди всех сертификатов участия, купленных за этот месяц, выбирается сертификат, номер билета которого находится как можно ближе к  $X$ , — и обладатель этого сертификата и получает приз. Если таких сертификатов два, то главный приз делится поровну между обладателями этих сертификатов. (В частности, если к билету с номером  $X$  был куплен сертификат, то его обладатель и получит приз.)

Некий человек получил доступ к базе данных продаж сертификатов за прошедший месяц и знает, с какими билетами покупался сертификат, а с какими нет. Он понимает, что пассажиры, которые не покупали сертификат, не будут следить за результатами лотереи, и хочет воспользоваться этим, чтобы выиграть главный приз.

А именно, он может выбрать некоторые номера билетов, на которые не были куплены сертификаты, и подделать соответствующие сертификаты (в частности, внося соответствующие изменения в базу проданных сертификатов). После этого его поддельные сертификаты будут участвовать в розыгрыше на общих условиях. Естественно, подделывать сертификаты на те билеты, на которые был куплен честный сертификат, невозможно.

Определите, какое минимальное количество сертификатов он должен подделать, чтобы максимизировать шансы и величину выигрыша. Обратите внимание, что если у него есть два способа действий, различающихся лишь тем, что при некоторых исходах лотереи в первом случае ему придётся делить выигрыш с кем-то, а во втором случае весь выигрыш достанется ему, то второй способ действий предпочтительнее, даже если в нем надо подделывать больше сертификатов.

### Формат входных данных

В первой строке входного файла находятся три целых числа —  $L$ ,  $R$  и  $N$  ( $1 \leq L \leq R \leq 10^9$ ,  $0 \leq N \leq 10^5$ ) — минимальный и максимальный номера проданных билетов и количество проданных сертификатов соответственно.

Во второй строке находятся  $N$  целых чисел  $a_i$  — номера билетов, для которых были проданы сертификаты ( $L \leq a_i \leq R$ ). Номера билетов расположены строго по возрастанию.

### Формат выходных данных

В первой строке выходного файла выведите минимальное количество сертификатов, которые надо подделать, чтобы максимизировать шансы выиграть.

Во второй строке выведите номера билетов, сертификаты к которым необходимо подделать, упорядоченные по возрастанию.

### Пример

Входной файл	Выходной файл
5 20 3	4
10 16 17	9 11 15 18
1 4 2	2
1 4	2 3

*Примечание:* Тесты к этой задаче будут устроены следующим образом:


- тесты суммарной стоимостью 20 баллов будут иметь  $L, R, N \leq 1000$ ;
- тесты суммарной стоимостью 30 баллов будут иметь  $L, R, N \leq 100\,000$ , при этом как минимум одно из значений будет больше 1000;
- тесты суммарной стоимостью 50 баллов не будут попадать ни под одно из указанных выше условий.

## Задача 3. Рисование кругами

Входной файл	<code>circles.in</code>
Выходной файл	<code>circles.out</code>
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	<code>p99_3.pas</code>

Маленькая девочка Оля очень любит пони. И очень любит рисовать. Поэтому она была очень рада, когда родители подарили ей на день рождения раскраску с пони. Причём, поскольку Оля уже большая девочка и недавно научилась считать, то эта раскраска необычная — в ней на каждой странице просто отмечены несколько точек, занумерованных последовательными натуральными числами. Прежде чем начать раскрашивать, надо эти точки последовательно соединить: первую со второй, вторую с третьей и так далее, предпоследнюю с последней, а последнюю — с первой.

Оля принялась за работу. Но, поскольку бабушка подарила Оле на тот же день рождения набор для рисования, и Оле там очень понравился циркуль, то она хочет соединять точки с помощью циркуля. А циркулем, как известно, можно нарисовать только дуги окружностей. Поэтому Оля хочет нарисовать одну дугу окружности от первой точки ко второй, ещё одну — от второй к третьей, и т.д.

Оля уже начала рисовать, но обнаружила проблему: в местах стыка двух последовательных дуг окружностей получаются острые углы, а Оля очень не любит углы. Поэтому она хочет, чтобы окружности стыковались как можно плавнее. Строго говоря (хотя Оля и не знает таких слов), она хочет, чтобы у каждой пары последовательных дуг окружностей в их общем конце (в точке, изначально отмеченной в раскраске) была общая касательная; и более того, чтобы ни в какой точке не происходило изменения направления движения на противоположное (т.е. не допустимы фрагменты типа такого: .

Помогите ей: определите, какие точки Оля должна использовать в качестве центров дуг.

### Формат входных данных

В первой строке входного файла находится одно число  $n$  — количество точек на очередной странице раскраски. Далее следуют  $n$  строк,  $i$ -я из которых содержит два вещественных числа  $x_i$  и  $y_i$  — координаты очередной точки.

Гарантируется, что  $3 \leq n \leq 50\,000$ , и что координаты точек не превосходят по модулю 10 000. Гарантируется, что расстояние между любыми двумя последовательными точками не меньше 1, и что расстояние между  $n$ -ой и первой точкой также не меньше 1.

### Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно слово “yes” (без кавычек), а далее выведите  $n$  строк. На  $i$ -ую из них выведите два числа: координаты центра дуги, соединяющей  $i$ -ую и  $(i + 1)$ -ую точку ( $n$ -ая строка должна содержать координаты центра дуги, соединяющей  $n$ -ую и первую точку), а затем один символ: “+”, если дугу от  $i$ -ой точки до  $(i + 1)$ -ой (для последней строки — от  $n$ -ой до первой) следует рисовать против часовой стрелки, и “-”, если по часовой стрелке. Разделяйте числа между собой, а также отделяйте их от символа пробелом.

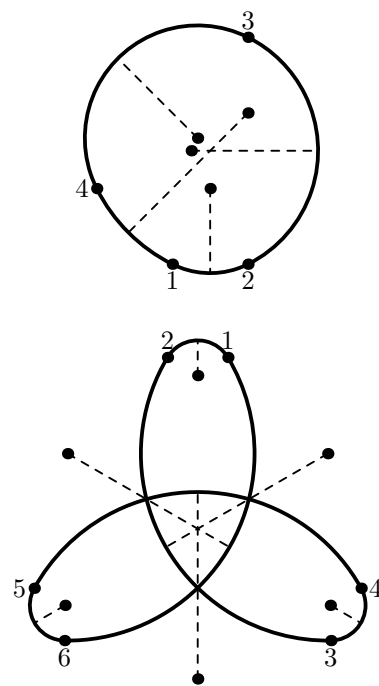
Если решения не существует, то выведите в выходной файл одно слово “no” (без кавычек).

Если существует несколько решений, выведите любое.

Гарантируется, что если решение существует, то существует решение, в котором координаты центров окружностей не превосходят по модулю 1 000 000. (Но вы не обязаны искать именно такое решение.) Также гарантируется, что если решения не существует, то при достаточно малом сдвиге данных точек оно не появляется.

При проверке вашего ответа все сравнения будут проверяться с абсолютной точностью 0.01.

Обратите внимание, что решение вполне может быть самопересекающимся.



*Пример и комментарии на обороте*

**Пример**

<i>Входной файл</i>	<i>Выходной файл</i>
4 0 0 2 0 2 6 -2 2	yes 1 2 + 0.5 3 + 0.66666 3.33334 + 2 4 +
6 0.69458 3.93921 -0.69458 3.93921 3.06415 -2.57117 3.75879 -1.3681 -3.75879 -1.3681 -3.06415 -2.57117	yes 0 3.52246 + 2.99188 1.72736 + 3.05054 -1.76123 + 0 -3.4544 + -3.05054 -1.76123 + -2.99188 1.72736 +

*Примечание:* Примеры соответствуют рисункам. На каждом рисунке точками с цифрами обозначены изначально нарисованные в раскраске точки. Точками без цифр обозначены центры дуг окружностей; кроме того, из каждого центра пунктиром проведён радиус до некоторой точки соответствующей дуги, чтобы было понятно, какой центр к какой дуге относится.

*Примечание:* Среди тестов будут, в том числе, тесты с маленьким  $n$ , а именно, с  $n = 3, 4, 5, 6$ . Для каждого из этих значений  $n$  суммарная стоимость соответствующих тестов будет 15 баллов.

## Задача 4. Кастрюли и крышки

Входной файл	pots.in
Выходной файл	pots.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_4.pas

Страшный беспорядок произошёл сегодня утром в столовой школы, где вы учитесь. Мария Ивановна, уборщица, подметая полы, уронила один из шкафов, и вся находившаяся в нем посуда разлетелась по всей столовой. К счастью, в этом шкафу хранились лишь металлические кастрюли с крышками. Однако, при падении некоторые из них погнулись, и их пришлось выбросить.

Теперь директор школы хочет подсчитать ущерб, и понять, сколько же новых кастрюль или крышек придётся закупать. Но сначала она хочет выяснить, а сколько же уцелевших кастрюль можно накрыть оставшимися крышками?

Кастрюли и крышки в сечении являются окружностями. Крышкой можно накрыть кастрюлю тогда и только тогда, когда радиус крышки больше либо равен радиуса кастрюли.

### Формат входных данных

В первой строке входного файла находятся два целых числа  $n$  и  $m$  ( $1 \leq n \leq 1000$ ,  $1 \leq m \leq 1000$ ) — количество оставшихся кастрюль и крышек соответственно.

Во второй строке находятся  $n$  целых чисел  $a_i$  — радиусы оставшихся кастрюль ( $1 \leq a_i \leq 1000$ ).

В третьей строке находятся  $m$  целых чисел  $b_i$  — радиусы оставшихся крышек ( $1 \leq b_i \leq 1000$ ).

### Формат выходных данных

Выведите единственное число — максимальное количество кастрюль, которые можно накрыть имеющимися крышками.

### Пример

Входной файл	Выходной файл
5 5 4 8 1 2 5 7 2 4 6 5	4

## Задача 5. Сумма степеней

Входной файл	powersum.in
Выходной файл	powersum.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МиБ
Максимальный балл за задачу	100
Пример названия программы	p99_5.pas

Ваш маленький брат уже учится в девятом классе, и вовсю интересуется арифметикой и алгеброй. Например, он давно знает, как вычислить сумму

$$1^2 + 2^2 + \dots + n^2.$$

Ему теперь хочется научиться вычислять аналогичную сумму для произвольного показателя степени:

$$S_{nk} = 1^k + 2^k + \dots + n^k.$$

Правда, он не любит возиться с большими числами, зато ему очень нравится операция взятия остатка от деления одного числа на другое. Напишите программу, которая по данным  $n$ ,  $m$  и  $k$  вычислит остаток от деления  $S_{nk}$  на  $m$ .

### Формат входных данных

В первой строке входного файла находятся три целых числа —  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 10^9$ ,  $1 \leq m \leq 10^9$ ,  $1 \leq k \leq 20$ ).

### Формат выходных данных

Выведите единственное число — остаток от деления  $S_{nk}$  на  $m$ .

### Пример

Входной файл	Выходной файл
3 1000 5	276
3 4 5	0

*Примечание:* Среди тестов будут тесты с  $n \leq 10^5$  общей стоимостью 30 баллов, и тесты с  $n > 10^5$ , но  $m \leq 50\,000$ , общей стоимостью ещё 30 баллов.

## Задача 6. Расписание поездов

Входной файл	trains.in
Выходной файл	trains.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МиБ
Максимальный балл за задачу	100
Пример названия программы	p99_6.pas

Ваш друг все ещё работает главным диспетчером в компании, владеющей сетью маршрутных такси. Благодаря вашей помощи он сумел весьма эффективно оптимизировать расписание маршруток, и на проходившем недавно бизнес-форуме он выступал с докладом, посвящённым как раз этой оптимизации. После доклада к нему подошли представители одной из железнодорожных компаний и попросили помочь с оптимизацией расписания движения поездов их компании.

Их компания обслуживает несколько железнодорожных маршрутов. По каждому маршруту каждый день должен следовать поезд, содержащий определённое количество вагонов: а именно, в одно и то же время каждый день поезд должен отправляться с начальной станции и в определённое время в этот или на следующий день прибывать на конечную станцию. Времена отправления и прибытия давно согласованы с администрацией железнодорожных станций, да и вообще с клиентами этой компании, поэтому они жёстко фиксированы. Количество вагонов в каждом поезде также жёстко фиксировано.

Компания перевозит в основном однотипный груз, и поэтому все вагоны, которые у неё имеются, абсолютно одинаковы: любой вагон можно поставить в любой поезд. Более того, когда некоторый поезд прибывает на конечную станцию, этот поезд можно расформировать и соответствующие вагоны использовать в других поездах, отправляющихся с этой станции в тот же момент или позже (в этот же день или на следующий день).

Соответственно, перед компанией стоит задача оптимизации вагонного парка: они хотят оставить у себя минимальное количество вагонов, которых будет достаточно, чтобы обслуживать все маршруты.

Помогите им и вашему другу определить это минимальное количество.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $N$  и  $M$ : количество станций на железной дороге и количество маршрутов, которые обслуживает компания, соответственно. Гарантируется, что оба числа не превосходят 100 000.

Далее следуют  $M$  строк, каждая из которых содержит описание очередного маршрута. А именно,  $i$ -я строка содержит сначала три натуральных числа  $s_i$ ,  $f_i$  и  $n_i$ : номера начальной и конечной станции этого маршрута ( $1 \leq s_i, f_i \leq N$ ) и количество вагонов, которые должны быть в поездах этого маршрута ( $1 \leq n_i \leq 100\,000$ ). Далее в той же строке в формате `hh:mm` заданы времена отправления и прибытия поездов этого маршрута (где `hh` — количество часов, а `mm` — количество минут). Если первый момент времени меньше второго, то поезд должен прибывать в тот же день, если же второй момент меньше первого, то это обозначает, что поезд должен прибывать на следующий день в указанное время.

Гарантируется, что для каждого поезда заданные времена отправления и прибытия не совпадают. Гарантируется, что для каждой станции количество вагонов, прибывающих на эту станцию за день, в точности равно количеству вагонов, отправляющихся с этой станции за день.

Считайте, что вагоны, прибывшие на станцию в некоторый момент, можно отправлять с поездами, которые отправляются в этот же момент времени или позже.

### Формат выходных данных

В выходной файл выведите одно число — минимальное количество вагонов, которые надо иметь железнодорожной компании, чтобы иметь возможность обслуживать все маршруты.

*Пример и примечания на обороте*

**Пример**

<i>Входной файл</i>	<i>Выходной файл</i>
2 2 1 2 5 06:00 13:00 2 1 5 16:00 17:00	5
3 4 3 1 3 23:59 00:01 1 2 15 06:00 13:00 2 1 12 16:00 04:00 2 3 3 17:00 23:59	15

*Примечание:* В первом примере достаточно иметь пять вагонов, которые каждый день будут сначала отправляться с первой станции на вторую с поездом первого маршрута, а потом возвращаться с поездом второго маршрута.

Во втором примере достаточно иметь 15 вагонов. Каждый день в 6 утра все пятнадцать вагонов в составе поезда второго маршрута будут отправляться с первой станции на вторую. Далее в 16:00 двенадцать из этих вагонов возвращаются обратно на первую станцию с поездом третьего маршрута. Три оставшиеся вагона в 17:00 отправляются на третью станцию с поездом четвертого маршрута, куда прибывают в 23:59, и тут же отправляются с поездом первого маршрута обратно на первую станцию.