

## Задача А. Министерство правды

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Бело-чёрным по чёрно-белому написано изречение, подвергнутое цензуре со стороны Министерства правды. Автор его уже исчез вместе с историей о нём, и, пока Большой Брат наблюдает за кем-то другим, вам, как рядовому сотруднику миниправа, поручено стереть из изречения часть букв так, чтобы получилось другое, одобренное министерством, изречение.

Министерство правды называет *словом* непустую последовательность латинских букв, а *изречением* — последовательность из одного или более слов, разделённых одним или несколькими пробелами. Пробелы также могут стоять перед первым и за последним словом изречения. Для сравнения двух изречений нужно сначала удалить все начальные и конечные пробелы, затем все блоки подряд идущих пробелов заменить на один пробел. Изречения считаются одинаковыми, если получившиеся строки совпадут. Когда сотрудник стирает букву в изречении, она заменяется на пробел.

### Формат входного файла

В первой строке записано исходное изречение, а во второй — изречение, которое требуется получить. Длина каждого изречения не превосходит 100 000 символов. Слова в этих изречениях разделены ровно одним пробелом, первый и последний символ каждой строки не может быть пробелом. Исходное и требуемое изречения различны.

### Формат выходного файла

Если вы не сможете выполнить поручение, выведите в единственной строке «I HAVE FAILED!!!». В противном случае выведите исходное изречение, заменив в нём буквы, которые надо стереть, на символ подчёркивания.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>Preved to Medved Preved Me</code>	<code>Preved __ Me____</code>
<code>this is impossible im possible</code>	<code>I HAVE FAILED!!!</code>

## Задача В. Забытые технологии

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Зима. Конец декабря. Падают снег. Ветер подхватывает снежинки и со свистом проносит их меж каменных строений. Монастырь выглядит заброшенным — двери занесены снегом, окна плотно прикрыты ставнями. Лишь только окна скриптория — самые большие в монастыре — распахнуты настежь. Неяркий зимний свет падает в просторную комнату, где в полной тишине сидят несколько монахов. В комнате настолько холодно, что пальцы монахов немеют. Но они продолжают торопливо писать гусиными перьями, стараясь успеть как можно больше за короткий декабрьский день.

В скриптории совершается великое таинство — идёт копирование информации.

Один монах, бывший подмастерье, пишет свою первую книгу. Его губы шевелятся, он неслышно повторяет каждую строчку, чтобы не допустить ошибки. Другой переписал уже десятки книг. Каждая выведенная им буква неотличима от оригинала. Он занимается этим всю свою жизнь и не умеет ничего другого.

В соседнем помещении сидит монах, которого называют корректор. Он занимается не менее важным делом — расставляет на полях книги пометки и исправления. Также в скриптории работают иллюминаторы и переплётчики, придающие книге тот вид, в котором она попадает в руки богатых мирян. Труд всех этих людей является одним из столпов экономики монастыря.

Но свет тускнеет, монахи откладывают свои перья и расходятся по кельям. Заканчивается день. Заканчивается 1439 год. Заканчивается эпоха монахов-переписчиков. В следующем, 1440 году, Иоганн Гутенберг напечатает свою первую книгу.

\*\*\*

Осень. Середина ноября. Идёт дождь. Молодая девушка сидит в закусочной на пересечении 112-й улицы и Бродвея. Она смотрит в окно на свинцовые тучи, нависающие над городом. Возможно, всё дело в погоде, но с самого утра девушка чувствует странную пустоту в душе. Она чувствует себя безучастным наблюдателем, которому нет дела ни до чего вокруг.

К девушке подходит официант и наливает в её кружку кофе до половины. Она поднимает глаза, пытаясь протестовать, но официант уже не смотрит на неё. Он смотрит на женщину, которая входит в закусочную и отряхивает зонтик. На его лице появляется улыбка, и он рассыпается в любезностях. Девушка отворачивается от них и наливает молока в свой кофе.

Чтобы чем-то занять себя, она берёт свежую газету. В ней нет ничего интересного, кроме истории об актёре, который, будучи в состоянии алкогольного опьянения, упал у себя в квартире и умер. Впрочем, имя этого актёра ей ничего не говорит. Она перелистывает страницы в поисках гороскопа и карикатур.

Ей кажется, что на неё кто-то смотрит. Она поворачивает голову и видит женщину за окном, смотрящую внутрь. Через секунду она понимает, что женщина всего лишь смотрит на своё отражение, поправляя юбку и подтягивая чулки.

А дождь всё льёт и льёт. До девушки доносится звон колоколов собора Иоанна Богослова, расположенного неподалёку. Этот звон будит в ней старые воспоминания о её друге, о том, как они когда-то сидели вдвоём на ступенях этого собора в полпервого ночи.

Девушка допивает кофе и выходит из закусочной. На вывеске написано «У Тома». Девушку зовут Сюзанна Вега. Через несколько месяцев она напишет песню об этом утре. А ещё через десять лет специалист по кодированию звука из Германии использует эту песню для отладки своего нового метода сжатия. Он прослушает эту песню тысячи раз, прежде чем финальная версия алгоритма, который впоследствии назовут MP3, будет готова.

\*\*\*

В Музее истории науки, в зале «Забытые технологии» есть экспонат, который называется «Жертвы массового копирования». Под стеклом на сукне лежит рукописная книга конца XV века, а на ней — компакт-диск конца XX века. Обе эти технологии были забыты после изобретения качественно новых способов копировать и распространять информацию. Рукописные книги не смогли конкурировать с дешёвой продукцией печатных прессов, так же как компакт-диски были вытеснены повсеместным распространением цифровых сетей. И теперь их можно найти только в частных коллекциях да в этом музее.

А ещё в этом музее живёт маленькая мышка. Она живёт там потому, что ей нравится всё необычное. Она прогрызла снизу дырку в сукне, и теперь ей хочется откусить хотя бы маленький кусочек от блестящего диска и утащить его себе в нору. Но она боится, что не успеет сделать это до того, как сработает сигнализация. Помогите ей найти кратчайший путь до диска.

Книга представляет собой прямоугольник. На книге лежит диск, его центр находится либо внутри прямоугольника, либо на его границе. Мышка может бегать только по сукну, её собственными размерами можно пренебречь. В начальном положении мышка находится на ненулевом расстоянии от книги и диска. Чтобы откусить кусочек от диска, мышка должна подбежать к той его точке, что находится строго за пределами прямоугольника, то есть свободно висит над сукном.

### Формат входного файла

В первой строке указаны координаты вершин прямоугольника, перечисленные в порядке обхода против часовой стрелки. В следующей строке указаны координаты центра диска и его радиус. В третьей строке указаны координаты начального положения мышки. Все числа целые и не превышают по модулю 1000. Радиус диска больше либо равен 1.

### Формат выходного файла

Выведите длину кратчайшего пути до диска с точностью не менее  $10^{-5}$ . Гарантируется, что искомый путь существует.

### Пример

input.txt	output.txt
-4 -3 8 6 5 10 -7 1 1 7 5 7 -1	12.07107

## Задача С. База эндшпилей

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Программист Старостин уже много лет писал программу для игры в шашки на доске размера  $n \times n$ . И момент триумфа был близок — скоро он собирался выпустить финальную версию и наслаждаться славой создателя лучшей в мире шашечной программы. Программа и так была уже практически непобедима, а с новой большой базой эндшпилей ей просто не должно было быть равных. Оставалось лишь сгенерировать эту базу...

Когда Старостин дописал генератор ходов на ассемблере, он обнаружил, что место на жёстком диске его компьютера закончилось. Для базы эндшпилей же требовалось огромное количество памяти. Чтобы оценить, сколько винчестеров придётся докупить, Старостин решил посчитать количество позиций в его базе эндшпилей заранее.

В базе будут храниться лишь те позиции, в которых на доске расположено ровно  $k$  фигур. Каждая фигура характеризуется её цветом (белый или чёрный) и типом (обычная шашка или дамка). Кроме того, на доске должна присутствовать хотя бы одна фигура каждого цвета, все фигуры должны стоять на чёрных клетках, а обычные шашки не должны стоять на своих полях превращения (белые шашки не могут стоять на последней горизонтали, а чёрные — на первой).

### Формат входного файла

В единственной строке через пробел записаны числа  $n$  и  $k$  ( $4 \leq n \leq 1\,000$ ;  $2 \leq k \leq \frac{n^2}{4}$ ;  $n$  чётно).

### Формат выходного файла

Выведите количество позиций в базе эндшпилей, вычисленное по модулю  $10^9 + 7$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
4 2	172

## Задача D. Похищение века

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Злоумышленник вынес из хранилища, в котором лежали золотые и платиновые слитки,  $n$  мешков со слитками. Во всех мешках лежало по  $k$  слитков, причём в каждом мешке лежали слитки только одного из этих металлов. Чтобы вызывать меньше подозрений, грабитель покрасил все украденные слитки чёрной краской, после чего золотые слитки стали внешне неотличимы от платиновых.

Вскоре он нашёл покупателя на платиновые слитки. И в этот момент оказалось, что он не знает, какие слитки лежат в каком мешке. Грабитель не растерялся, достал весы, показывающие точный вес, и решил за одно взвешивание определить, в каких мешках лежат платиновые слитки, а в каких — золотые. Известно, что золотой слиток имеет массу  $x$  килограммов, а платиновый слиток — массу  $y$  килограммов. Подскажите грабителю, сколько слитков из какого мешка нужно положить на весы, чтобы за одно взвешивание гарантированно определить, в каких мешках лежат слитки из платины.

### Формат входного файла

В единственной строке через пробел записаны целые числа  $n, k, x, y$  ( $3 \leq n \leq 20$ ;  $1 \leq k \leq 10^6$ ;  $1 \leq x < y \leq 10$ ).

### Формат выходного файла

Если можно за одно взвешивание определить, в каких мешках лежат платиновые слитки, в первой строке выведите «YES», а во второй строке выведите через пробел  $n$  целых чисел — сколько слитков из каждого мешка нужно положить для этого на весы. Если задача имеет несколько решений, выведите любое из них. Если задача не имеет решения, в единственной строке выведите «NO».

### Примеры

input.txt	output.txt
4 8 9 10	YES 4 5 6 8
3 3 9 10	NO

## Задача Е. Китайский хоккей

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Китайский чемпионат по футболу, за которым внимательно следили Серёга с Денисом, завершился. К сожалению, команды «Katraps» и «Kolomotiv», за которые болели друзья, разделили в чемпионате последнее место. С досады Серёга предложил Денису переквалифицироваться в хоккейных болельщиков.

В китайском чемпионате по хоккею участвует  $n$  команд. Во время сезона каждая команда должна сыграть с каждой ровно один матч. Если победитель матча определяется в основное время, то он получает 3 очка, а проигравшая команда — 0 очков. Если же основное время матча заканчивается вничью, то назначается овертайм. Команда, одержавшая победу в овертайме, получает 2 очка, а проигравшая команда — 1 очко. Ничьих в хоккее не бывает.

Денис хочет определиться, за какую хоккейную команду стоит болеть. С этой целью он нашёл в Интернете таблицу, в которой для каждой из  $n$  команд указано, сколько очков она набрала по результатам прошлого хоккейного сезона. Серёга подозревает, что в этой таблице содержится ошибка, поскольку никакой круговой хоккейный чемпионат не может завершиться с такими результатами. Прав ли он?

### Формат входного файла

Первая строка содержит целое число  $n$  ( $2 \leq n \leq 200$ ). Во второй строке через пробел записаны  $n$  целых неотрицательных чисел — очки, набранные каждой из команд по результатам прошлого чемпионата. Очки перечислены в порядке невозрастания, сумма всех очков равна  $\frac{3n \cdot (n - 1)}{2}$ . Никакая команда не набрала более  $3 \cdot (n - 1)$  очков.

### Формат выходного файла

Если Серёга прав и в таблице содержится ошибка, выведите в единственной строке «INCORRECT». Иначе в первой строке выведите «CORRECT», а в следующих  $\frac{n \cdot (n - 1)}{2}$  строках выведите результаты хоккейных матчей. Результат матча должен иметь вид « $i \ ? \ j$ », где  $i$  и  $j$  — номера команд, игравших в этом матче, а ? принимает значения  $<$ ,  $<=$ ,  $>=$  и  $>$ , означающие, что первая команда проиграла в основное время, проиграла в овертайме, одержала победу в овертайме и одержала победу в основное время, соответственно. Команды пронумерованы целыми числами от 1 до  $n$  в том порядке, в котором перечислены их очки во входных данных.

### Примеры

input.txt	output.txt
4 8 7 2 1	CORRECT 2 <= 1 3 >= 4 1 > 3 4 < 2 1 > 4 2 > 3
4 8 8 1 1	INCORRECT

## Задача F. Мнемоника и палиндромы 3

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Как вы помните, когда Васечкин готовил задачу для последнего студенческого соревнования, он долго пытался придумать для неё необычное и сложное название. Но название, придуманное им, оказалось настолько сложным, что никто из участников того соревнования даже не стал читать текст задачи Васечкина.

После этого председатель программного комитета заявил, что отказывается участвовать в подготовке соревнований, пока в программном комитете работают такие неадекватные люди, как Васечкин. Так Васечкин стал новым председателем программного комитета и начал готовить следующее соревнование по программированию.

Васечкин решил, что на этот раз названия всех задач будут состоять только из букв «a», «b» и «c» и иметь длину  $n$ . Кроме того, все названия должны быть *чрезвычайно сложными*, а именно, никакая подстрока никакого названия, состоящая из двух или более символов, не должна являться палиндромом. Помогите Васечкину и найдите все чрезвычайно сложные названия для задач предстоящего соревнования.

### Формат входного файла

В единственной строке записано целое число  $n$  ( $1 \leq n \leq 20\,000$ ).

### Формат выходного файла

Выведите все различные чрезвычайно сложные названия длины  $n$ , состоящие только из букв «a», «b» и «c». Названия следует выводить в алфавитном порядке, по одному в строке. Если суммарная длина названий превосходит 100 000 букв, выведите единственную строку «TOO LONG».

### Пример

input.txt	output.txt
2	ab ac ba bc ca cb

## Задача G. Компьютерная безопасность

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В компании *ZZZ Inc.* произошло ЧП — её секретные разработки стали известны конкурентам! Конечно, в первую очередь подозрение пало на сотрудников самой компании, и системному администратору Жучкову дали задание — узнать, кто из сотрудников скачивал с сервера секретную информацию в течение последней недели. Жучков просмотрел логи и нашёл личный номер и пин-код этого сотрудника. Администратор доложил об этом начальству, и в тот же вечер сотрудник пропал без вести, а Жучков получил премию.

Однако вскоре выяснилось, что Жучков перепутал некоторые символы в личном номере и пин-коде, и в результате пострадал невинный человек! Чтобы в будущем не допускать столь досадных оплошностей, Жучков решил найти пары сотрудников с похожими номерами и проверить, насколько похожи их пин-коды.

Жучков считает два личных номера *похожими*, если один номер можно получить из другого вставкой, удалением или заменой одной цифры. При этом в записи личного номера не может быть ведущих нулей. В компании *ZZZ Inc.* сотрудники нумеруются последовательными целыми числами, начиная с единицы, а пин-коды всех сотрудников представляют собой строку из четырёх шестнадцатеричных цифр. Для каждой пары сотрудников с похожими номерами Жучков хочет вычислить количество позиций, в которых различаются их пин-коды. Помогите ему в этом.

### Формат входного файла

В первой строке записано целое число  $n$  ( $2 \leq n \leq 65\,536$ ) — количество сотрудников *ZZZ Inc.* В  $i$ -й из следующих  $n$  строк записан пин-код сотрудника с личным номером  $i$ . Для записи пин-кода используются цифры и строчные латинские буквы. Пин-коды всех сотрудников различны.

### Формат выходного файла

Выведите через пробел четыре целых числа.  $i$ -е число должно равняться количеству пар сотрудников с похожими номерами, пин-коды которых различаются в  $i$  позициях.

### Пример

<code>input.txt</code>	<code>output.txt</code>
3 dead beef f00d	0 0 2 1

## Задача Н. Фарюки

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Вы решили сделать рождественский стол оригинальным и пробрались на единственную в мире плантацию фарюков, намереваясь раздобыть этих загадочных фруктов. Но своровать фарюки не так просто. Плантация обнесена высоким забором, поэтому попасть на плантацию или выйти оттуда можно только через проходную. На плантацию можно пронести с собой сколько угодно фарюков, но выносить их с плантации категорически запрещено — фарюки обладают сильным ароматом, и автоматический определитель запаха на проходной поднимает тревогу при его обнаружении. К счастью, вы знаете, что в программном обеспечении определителя запаха есть ошибка — из-за использования однобайтовых переменных он не поднимает тревогу, если суммарная сила аромата всех выносимых фарюков целая и делится на 256.

Также вы решили захватить с собой на дело пузырёк с духами и баллончик с дезодорантом. При обработке фарюка дезодорантом сила его аромата уменьшается на 25 процентов, а при обработке духами — увеличивается на 10 процентов (при этом сила аромата фарюка может перестать быть целым числом). И того, и другого средства хватит на обработку только одного фарюка. Один и тот же фарюк можно обработать сначала одним средством, а потом другим. Эффект от дезодоранта и духов длится недолго, и после того, как вы пронесёте фарюк через проходную, сила его аромата примет исходное значение.

Итак, вы пришли на плантацию с пустыми руками и присмотрели себе  $n$  фарюков. Получится ли у вас заполучить их все?

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 12$ ) — количество интересующих вас фарюков. Во второй строке через пробел записаны силы аромата этих фарюков —  $n$  целых чисел от 1 до 1 000.

### Формат выходного файла

Если вынести с плантации все нужные вам фарюки невозможно, выведите «IMPOSSIBLE».

В противном случае, выведите в первой строке количество действий, необходимых для осуществления вашего плана. Далее выведите последовательность действий, по одному действию в строке. Если вы хотите вынести  $k > 0$  фарюков с номерами  $a_1, \dots, a_k$  с плантации, выведите строку «take  $k$   $a_1$   $a_2$  ...  $a_k$ ». Аналогично, если вы хотите пронести  $s > 0$  фарюков с номерами  $b_1, \dots, b_s$  на плантацию, выведите строку «return  $s$   $b_1$   $b_2$  ...  $b_s$ ». Если вы хотите обработать дезодорантом фарюк с номером  $p$ , выведите строку «dearomatize  $p$ ». Если вы хотите обработать духами фарюк с номером  $q$ , выведите строку «aromatize  $q$ ». Фарюки занумерованы числами от 1 до  $n$  в том порядке, в котором они описаны на входе.

Количество действий в последовательности не должно превосходить 20 000. Если существует несколько способов вынести все фарюки, выведите такой из них, при котором количество выходов с фарюками через проходную минимально. Если и таких способов несколько, выведите любой из них.

### Примеры

input.txt	output.txt
3 36 56 200	4 take 2 2 3 return 1 3 aromatize 3 take 2 1 3
1 100	IMPOSSIBLE

## Задача I. А олени лучше!

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

*Сидит ворона на суку, во рту сыр. Бежит лиса:*

*— Ворона, ты на выборы пойдёшь?*

*— Нет!*

*Сыр выпадает. Ворона сидит и думает:*

*— Если бы я сказала «да», разве что-нибудь изменилось бы?*

Чукча поехал на выборы. От его чума до избирательного участка  $l$  километров. Его олени могут бежать только так, чтобы любой отрезок пути длиной  $k$  километров пробегать ровно за  $h$  часов. Через какое минимальное и максимальное время чукча сможет выполнить свой гражданский долг?

### Формат входного файла

В единственной строке через пробел записаны целые числа  $l$ ,  $k$  и  $h$  ( $1 \leq k \leq l \leq 1000$ ;  $1 \leq h \leq 1000$ ).

### Формат выходного файла

Выведите через пробел минимальное и максимальное время, за которое чукча сможет доехать от чума до избирательного участка, с точностью не менее  $10^{-5}$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
30 10 1	3.00000000 3.00000000

## Задача J. Монстр общения

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

После возвращения из летнего лагеря мальчик Коля стал настоящим «монстром общения». Он проводит в интернете всё свободное время, общаясь с друзьями по ICQ. Но совсем недавно протокол этого сервиса был в очередной раз изменён, и клиент Коли перестал работать. Теперь, для того чтобы общаться с друзьями дальше, ему необходимо обновить свой клиент с версии 1 до версии  $n$ .

В интернете Коля нашёл  $m$  обновлений,  $i$ -е из которых обновляет клиент с версии  $x_i$  до  $y_i$  и имеет размер  $d_i$  мегабайт. Каждое обновление устанавливается только на соответствующую версию клиента, но ни на какую более раннюю или более позднюю версию.

Первая версия клиента, имеющаяся у Коли — лицензионная, а многие из найденных им обновлений являются пиратскими. После установки пиратского обновления клиент всегда будет пиратским, вне зависимости от обновлений, которые будут устанавливаться после этого. Среди остальных обновлений имеются как лицензионные, которые можно устанавливать на пиратскую версию клиента, так и лицензионные, которые этого не допускают. Пиратские обновления можно ставить как на лицензионную, так и на пиратскую версию клиента.

Коля уже очень соскучился по друзьям, поэтому он хочет как можно быстрее скачать нужные обновления, а интернет у него совсем не быстрый. Помогите ему определить минимальный суммарный объём трафика, который нужно потратить для обновления клиента с версии 1 до  $n$ . Стоит отметить, что Коле не важно, будет ли клиент версии  $n$  лицензионным.

### Формат входного файла

В первой строке через пробел записаны целые числа  $n$  и  $m$  ( $2 \leq n \leq 10^4$ ;  $1 \leq m \leq 10^4$ ).

Каждая из следующих  $m$  строк содержит описание одного обновления в формате  $x_i y_i d_i s_i$ , где  $s_i$  — тип обновления: «Pirated» — пиратское, «Cracked» — лицензионное, которое ставится на пиратскую версию, «Licensed» — лицензионное, которое не ставится на пиратскую версию, а числа  $x_i, y_i, d_i$  означают, что обновление устанавливается на клиент версии  $x_i$ , обновляет его до версии  $y_i$  и имеет размер  $d_i$  мегабайт ( $1 \leq x_i < y_i \leq n$ ;  $1 \leq d_i \leq 10^6$ ). Данные в каждой строке разделены ровно одним пробелом.

### Формат выходного файла

Если у Коли получится успешно обновить клиент с версии 1 до версии  $n$ , в первой строке выведите «Online», а во второй — минимальный суммарный объём входящего трафика, необходимый для обновления.

Если же обновить клиент не получится, выведите «Offline».

### Примеры

input.txt	output.txt
3 4 1 3 10 Licensed 1 2 2 Pirated 2 3 3 Licensed 2 3 6 Cracked	Online 8
3 1 1 2 10 Licensed	Offline

## Задача К. Фантазия

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Кончилась.

### Формат входного файла

В первой строке записано целое число  $d$  ( $1 \leq d \leq 50$ ) — количество тестовых примеров.

В каждой из следующих  $d$  строк через пробел записаны три целых числа  $p$ ,  $a$  и  $m$  ( $2 \leq p < 10^9$ ;  $0 \leq a < p$ ;  $1 \leq m \leq 20$ ;  $m < p$ ). Гарантируется, что  $p$  простое.

### Формат выходного файла

Выведите одну строку для каждого тестового примера. Если существует такое целое положительное  $n < 10^{18}$ , что  $n^n + n^m$  даёт остаток  $a$  по модулю  $p$ , выведите через пробел «YES» и любое из чисел  $n$ , удовлетворяющих этим ограничениям. В противном случае выведите «NO».

### Пример

<code>input.txt</code>	<code>output.txt</code>
2	YES 567
11 3 1	YES 2
11 8 2	