

Задача А. (a, b) -башня

Имя входного файла: `abtower.in`
Имя выходного файла: `abtower.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький Артём задался вопросом: как, записав всего несколько цифр, получить очень большое число? Конечно, один из способов — написать башню из степеней. Как известно, башня из степеней вычисляется сверху вниз: в выражении вида x^{y^z} сначала вычисляется y^z , а затем число x возводится в полученную степень. Например, короткая запись 2^{3^4} равна огромному числу $2^{3 \cdot 3 \cdot 3 \cdot 3} = 2^{81} = 2\,417\,851\,639\,229\,258\,349\,412\,352$.

Артём выписал целые числа от a до b по диагонали, двигаясь вверх и вправо. Получилась башня из степеней:

$$a^{(a+1)^{\dots^{(b-1)^b}}}$$

Помогите ему выяснить, чему равна последняя десятичная цифра этого — возможно, огромного — числа.

Формат входных данных

В первой строке записано два целых числа a и b — параметры башни ($1 \leq a \leq b \leq 10$).

Формат выходных данных

Выведите одно число — последнюю десятичную цифру числа

$$a^{(a+1)^{\dots^{(b-1)^b}}$$

Примеры

| <code>abtower.in</code> | <code>abtower.out</code> |
|-------------------------|--------------------------|
| 2 4 | 2 |
| 3 3 | 3 |

Пояснения к примерам

В первом примере получается число $2^{3^4} = 2^{81} = 2\,417\,851\,639\,229\,258\,349\,412\,352$. Последняя десятичная цифра этого числа равна 2.

Во втором примере получается просто число **3**, последняя и единственная десятичная цифра которого равна 3.

Задача В. Деление

Имя входного файла: `division.in`
Имя выходного файла: `division.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Алиса ходит в кружок по математике. Недавно на занятии кружка преподаватель Глеб Михайлович рассказывал про \mathbb{Z}_m — кольцо вычетов по модулю m . \mathbb{Z}_m состоит из m целых чисел: $0, 1, \dots, m-1$. Складывать, вычитать и умножать такие числа совсем просто: нужно сделать операцию как с обычными числами, а потом взять остаток от деления ответа на m . Например, при $m = 10$ сложение работает так: $2+3 = 5$, $3+5 = 8$, $8+5 = 3$ (поскольку ответ с обычными числами равен 13, а 3 — это остаток от деления 13 на 10). Вычитание работает так: $3-2 = 1$, $1-2 = 9$ (ответ с обычными числами равен -1 , а 9 — это остаток от деления -1 на 10). Умножение работает так: $2 \cdot 3 = 6$, $6 \cdot 8 = 8$ (с обычными числами получилось бы 48, а 8 — это остаток от деления 48 на 10).

Иначе обстоит дело с делением в \mathbb{Z}_m . Деление определяют как операцию, обратную к умножению (так же, как вычитание — операция, обратная к сложению). А именно, результат деления a/b равен c , если $c \cdot b = a$. К сожалению, это определение не всегда однозначно: таких чисел c иногда несколько, а иногда нет вообще. Например, если $m = 10$, то $6/2$ может дать как результат 3 (поскольку $3 \cdot 2 = 6$), так и результат 8 (поскольку $8 \cdot 2 = 6$; напомним, что все равенства в этой задаче записываются не для обычных чисел, а для чисел из \mathbb{Z}_m). А вот выражение $7/2$ вообще не может дать никакого результата: среди чисел $0, 1, \dots, 9$ нет такого числа x , что $x \cdot 2 = 7$.

Глеб Михайлович достал с полки какой-то старый калькулятор и объяснил, что этот калькулятор работает не с обычными числами, а как раз с \mathbb{Z}_m , причём $m = 2^{32} = 4\,294\,967\,296$. Сложение, умножение и вычитание он выполняет правильно. А вот при делении, если результат неоднозначен, калькулятор может с одинаковой вероятностью выдать любой из возможных ответов. Если же деление вообще не может дать никакого результата, на экране калькулятора появляется надпись «Error» (ошибка).

Затем Глеб Михайлович передал калькулятор ребятам как наглядное пособие. Когда калькулятор попал в руки к Алисе, она ввела число x и стала делить его на два, результат — ещё на два, и так далее. Через некоторое время на экране возникла надпись «Error». Алиса снова ввела x и попробовала ещё раз: действительно, калькулятор работал точно так, как сказал преподаватель.

Теперь Алисе интересно, какие числа могут получаться на калькуляторе, если ввести x , после чего ноль или более раз совершить деление на два. Она составила список чисел y_1, y_2, \dots, y_n , которые ей было бы интересно получить из x . Для каждого из этих n чисел выясните, возможно ли это.

Формат входных данных

В первой строке входных данных записаны два целых числа x и n — начальное число и количество интересующих Алису чисел ($0 \leq x \leq 4\,294\,967\,295$, $1 \leq n \leq 50$). Во

второй строке заданы n целых чисел y_1, y_2, \dots, y_n — числа, которые интересуют Алису ($0 \leq y_i \leq 4\,294\,967\,295$).

Формат выходных данных

Выведите n строк. Если число y_i может получиться из x после нуля или более делений на два в кольце вычетов по модулю 2^{32} , выведите в i -й строке слово «Yes», иначе выведите в ней «No».

Примеры

| <code>division.in</code> | <code>division.out</code> |
|--------------------------|---------------------------|
| 16 5 | Yes |
| 4 3 2 32 16 | No |
| | Yes |
| | No |
| | Yes |
| 123456789 3 | No |
| 123 456 789 | No |
| | No |

Пояснения к примерам

В первом примере из числа 16 могут получиться, например, числа 16, 8, 4, 2 и 1. Могут получиться и некоторые другие числа, но 3 и 32 получить не удастся.

Во втором примере из числа 123 456 789 нельзя получить никакое другое число: после первого же деления калькулятор выведет «Error».

Задача С. Цветной дом

Имя входного файла: house.in
Имя выходного файла: house.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче нужно выяснить цвет крыши дома по цветам полов и потолков всех этажей.

На окраине одного небольшого городка стоит недавно построенный цветной дом. В этом доме n этажей и $n + 1$ горизонтальная перегородка. Самая нижняя перегородка — это пол первого этажа, вторая снизу перегородка служит потолком первого этажа и полом второго, и так далее. Последняя перегородка — потолок верхнего этажа — является также крышей дома. Каждая горизонтальная перегородка этого дома со всех сторон покрашена в какой-то один цвет.

В этом доме также есть лифт с n кнопками. Каждая кнопка отправляет лифт на какой-то определённый этаж, разные кнопки — на разные этажи. Однако кнопки не подписаны, поэтому какая из них ведёт на какой этаж, неизвестно.

Исследуя цветной дом, мальчик Коля оказался в лифте. Он произвёл серию из n наблюдений: нажал по одному разу каждую из n кнопок, дождался, пока лифт приедет на соответствующий этой кнопке этаж и откроет двери, после чего записал себе в блокнот цвет пола и потолка на этаже, на который он попал. Для краткости Коля записывал вместо цветов целые числа так, что одному и тому же цвету соответствовало одно и то же число, а различным цветам — разные числа.

Придя домой, Коля внезапно понял, что забыл, какого цвета крыша у цветного дома. Он помнит только, что цвет пола самого нижнего этажа и цвет крыши различны. Пользуясь колышками записями, помогите ему восстановить цвет крыши.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 100$) — количество этажей в доме. Следующие n строк содержат по два целых числа каждая — записи колиных наблюдений. Первое из чисел в строке — цвет пола какого-то этажа, а второе — цвет потолка этого этажа. Соседние числа в этих строках разделены пробелами. Поскольку кнопки в лифте не подписаны, порядок этажей в этих записях может быть произвольным. Номера цветов, использованные Колей, лежат в диапазоне от 1 до 10 000 включительно.

Гарантируется, что колины записи корректны, то есть существует цветной дом, в котором описанная серия из n наблюдений могла дать такие результаты.

Формат выходных данных

В первой строке выходного файла выведите одно целое число, соответствующее цвету крыши цветного дома.

Примеры

| house.in | house.out |
|------------------------|-----------|
| 2 1 2 2 3 | 3 |
| 3 2 7 3 5 5 2 | 7 |

Пояснения к примерам

В первом примере дом состоит из двух этажей. Пол первого этажа имеет цвет 1, потолок первого и пол второго — цвет 2, а потолок второго и крыша дома — цвет 3.

Во втором примере дом состоит из трёх этажей. Пол первого этажа имеет цвет 3, потолок первого и пол второго — цвет 5, потолок второго и пол третьего — цвет 2. Наконец, потолок третьего этажа и крыша дома имеют цвет 7. Этажи в записях Коли перечислены в следующем порядке: третий, первый, второй.

Задача D. Новый дворец императора

Имя входного файла: palace.in
Имя выходного файла: palace.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Император Куско хочет построить летний дворец. Главный архитектор уже принёс план нового дворца — прямоугольник из $h \times w$ клеток, где отмечены клетки, которые займёт дворец. Но тут советница Ёзма сообщила императору, что на предполагаемом месте постройки обнаружена деревня! Жители деревни, встревоженные вестью о стройке, просят императора поставить дворец так, чтобы не пришлось сносить их хижины.

У Куско было хорошее настроение, и он повелел исполнить просьбу жителей. Главному архитектору принесли план деревни: прямоугольник из $m \times n$ клеток, где отмечены клетки, в которых стоят хижины. Теперь следует, двигая план дворца вдоль линий сетки, совместить его с планом деревни так, чтобы ни одна клетка дворца не оказалась совмещена с клеткой, в которой стоит хижина. Поскольку в архитектуре дворца важны стороны света, поворачивать планы не следует.

Помогите главному архитектору найти любое положение дворца, при котором не придётся сносить хижины, или выясните, что это невозможно.

Формат входных данных

В первой строке входных данных записаны два целых числа h и w — размер плана дворца ($1 \leq h \leq 10$, $1 \leq w \leq 10$). Следующие h строк содержат по w символов и задают план дворца. Каждый из этих символов — это либо точка («.»), обозначающая пустое место, либо звёздочка («*»), обозначающая часть дворца. Гарантируется, что из любой клетки дворца можно попасть в любую другую, двигаясь только между клетками дворца, имеющими общую сторону. Также гарантируется, что в первой и последней строке плана дворца, а также в первом и последнем его столбце, найдётся хотя бы одна звёздочка.

В следующей строке записаны два целых числа m и n — размер плана деревни ($h \leq m \leq 20$, $w \leq n \leq 20$). Следующие m строк содержат по n символов и задают план деревни. Каждый из этих символов — это либо точка («.»), обозначающая пустое место, либо маленькая буква «икс» («x»), обозначающая хижину. Последняя строка, как и все остальные, заканчивается переводом строки.

Формат выходных данных

Если дворец нельзя поставить, не снося хижины, выведите строку «NO». Иначе выведите строку «YES», а дальше — m строк по n символов в каждой. Каждый из этих символов должен быть либо точкой («.»), обозначающей пустое место, либо звёздочкой («*»), обозначающей часть дворца, либо маленькой буквой «икс» («x»), обозначающей хижину. Хижины должны стоять ровно там же, где они находятся на плане деревни. Дворец должен выглядеть точно так же, как на заданном плане, но может быть подвинут по горизонтали и вертикали на произвольное количество клеток. Если

существует несколько возможных расположений дворца, выведите любое из них.

Примеры

| palace.in | palace.out |
|--|--------------------------------|
| 2 3 *** **. 3 5 .x.x. ...x ...x. | YES .x.x. .***x .***. |
| 4 3 *** *.* **. .*. 4 5 .x... x..x. .x..x ...x. | NO |

Задача Е. Зоны патрулирования

Имя входного файла: `patrol-zones.in`
Имя выходного файла: `patrol-zones.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Прямой участок границы разделён на n участков километровой длины, некоторые из которых считаются непроходимыми. Нужно разбить все проходимые участки на зоны патрулирования. Каждая зона патрулирования должна состоять из двух или трёх проходимых участков. Кроме того, расстояние между любыми двумя участками одной зоны патрулирования в километрах должно быть строго меньше, чем количество участков в ней. Расстоянием между участками считается расстояние между ближайшими их точками. В частности, расстояние между соседними участками равно нулю.

Сколько существует способов разбить проходимую часть границы на зоны патрулирования?

Формат входных данных

Первая строка ввода содержит целое число n — длина границы в километрах ($1 \leq n \leq 50$). Вторая строка описывает границу и состоит из n символов из набора {«.» (точка), «X» (большая буква икс)}. Символ «.» соответствует проходимым участкам границы, а символ «X» — непроходимым.

Формат выходных данных

Выведите одно число — количество способов разбить границу на зоны патрулирования.

Примеры

| <code>patrol-zones.in</code> | <code>patrol-zones.out</code> |
|------------------------------|-------------------------------|
| 5 | 4 |
| 5 ...X. | 1 |
| 9 ...X.X... | 2 |
| 6 .XX... | 0 |

Пояснения к примерам

В первом примере есть четыре способа разбить границу на две зоны патрулирования (обозначим их буквами А и В): ААВВ, ААВВ, ААВВВ и АВВВ.

Во втором примере способ всего один: ААВХВ.

В третьем примере есть всего два способа разбить границу на три зоны патрулирования А, В и С: АААХВХВСС и ААВХВХССС.

В четвёртом примере первый участок не может оказаться ни в какой зоне патрулирования, поэтому ни одного разбиения не существует.

Задача F. Период строки

Имя входного файла: `period.in`
Имя выходного файла: `period.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Строка S имеет период T , если

$$\exists n > 0 : S = T^n = \underbrace{TT \dots T}_n.$$

Вам дана строка S . Ваша задача — найти минимальную по длине T такую, что $S = T^n$ для некоторого $n \in \mathbb{N}$.

Формат входных данных

Строка S длиной от 1 до 10^6 символов.

Формат выходных данных

Единственное число — длина T .

Примеры

| <code>period.in</code> | <code>period.out</code> |
|------------------------|-------------------------|
| abaabaabaabaaba | 3 |

Задача G. Вариация Нима

Имя входного файла: `varnim.in`
Имя выходного файла: `varnim.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На столе лежат n кучек камней: a_1 камней в первой кучке, a_2 камней во второй, \dots , a_n в n -ой. Двое играют в игру, делая ходы по очереди. За один ход игрок может либо взять произвольное ненулевое количество камней (возможно, все) из одной любой кучки, либо произвольным образом разделить любую существующую кучку, в которой не меньше двух камней, на две непустые кучки. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

Формат входных данных

В первой строке задано целое число t — количество тестов ($1 \leq t \leq 100$). Следующие t строк содержат сами тесты. Каждая из них начинается с целого числа n — количества кучек ($1 \leq n \leq 100$). Далее следует n целых чисел a_1, a_2, \dots, a_n через пробел — количество камней в кучках ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите t строк; в i -й строке выведите «FIRST», если в i -м тесте при правильной игре выигрывает первый игрок, и «SECOND», если второй.

Пример

| <code>varnim.in</code> | <code>varnim.out</code> |
|------------------------|-------------------------|
| 3 | FIRST |
| 1 1 | SECOND |
| 2 1 1 | FIRST |
| 3 1 2 3 | |

Задача Н. Вечерняя прогулка

Имя входного файла: walk.in
Имя выходного файла: walk.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

После трудного дня герцог Фред решил отвести своего друга, барона Питера, на прогулку в сад. В саду у герцога много примечательных мест — пруд, беседка, трёхсотлетний дуб, посаженный ещё пра-пра-пра-...-прадедом нынешнего герцога... Между некоторыми из этих мест есть тропинки.

Герцог хочет провести в саду не менее t минут — иначе гость не успеет нагулять аппетит перед ужином. С другой стороны, если гулять слишком долго, то наступит ночь, похолодает, и престарелый барон может простудиться, поэтому не следует гулять по саду более T минут.

Герцог Фред знает, сколько времени займёт перемещение по каждой тропинке от одного её конца до другого. Он стал придумывать маршрут, по которому следует совершить прогулку. Для этого он сперва пронумеровал все интересные места в своём саду числами от 1 до N . Маршрут должен начинаться и закончиться в месте с номером 1 — входе в герцогский дом. Кроме того, маршрут должен проходить через все остальные примечательные места не более одного раза. И наконец, время прогулки, равное суммарному времени прохода по всем тропинкам маршрута, должно быть от t до T минут, включительно.

Размышляя о маршруте, герцог спросил себя, а сколько всего маршрутов, удовлетворяющих всем перечисленным ограничениям, существует. Тут герцог Фред вынужден был констатировать, что его способностей к вычислению явно не хватает, чтобы ответить на этот вопрос. Помогите герцогу посчитать, сколько всего существует таких маршрутов, и... как знать, может быть, вы из подмастерья станете его придворным математиком.

Формат входных данных

В первой строке входного файла заданы целые числа N , M , t и T через пробел ($2 \leq N \leq 10$, $0 \leq M \leq 45$, $1 \leq t \leq T \leq 10\,000$). Следующие M строк содержат по три числа u_i v_i l_i каждая ($1 \leq u_i, v_i \leq N$, $1 \leq l_i \leq 1000$); эти три числа означают, что в саду есть тропинка между примечательными местами с номерами u_i и v_i , и чтобы по ней пройти в любую сторону, требуется l_i минут. Между любыми двумя примечательными местами может быть не более одной тропинки. В саду нет тропинок, соединяющих какое-то примечательное место с ним самим.

Формат выходных данных

Выведите в выходной файл одно число — количество маршрутов, удовлетворяющих всем ограничениям, поставленным герцогом. Маршруты считаются различными, если последовательности примечательных мест, посещаемых на этих маршрутах, различаются.

Примеры

| walk.in | walk.out |
|-------------------------------------|----------|
| 2 1 5 7 1 2 3 | 1 |
| 2 1 3 5 1 2 3 | 0 |
| 3 2 1 10 1 2 1 2 3 1 | 1 |
| 3 3 1 10 1 2 1 2 3 2 3 1 3 | 4 |