

Задача A. long long

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вы наверняка знаете, что при решении задач (особенно на олимпиадах по программированию) нужно выбирать целочисленные типы данных с умом. Если используемый тип имеет слишком малый размер, при выполнении арифметических операций может произойти переполнение, и тогда результат вычислений окажется неверным.

Макс решает задачу, в которой требуется определить произведение целых чисел A и B . Любимый язык Макса — C++, а в этом языке самым ёмким целочисленным типом является long long — на компьютере Макса это 64-битное целое со знаком. 1 бит отводится на хранение знака числа, остальные 63 — на хранение самого числа; таким образом, максимальное значение, которое может иметь переменная типа long long, равно $(2^{63} - 1)$.

В своём решении Макс использовал long long везде, где только мог — для хранения чисел A , B и их произведения. Определите, действительно ли это поможет ему избежать переполнения.

Формат входных данных

Ввод содержит целые числа A и B ($0 \leq A, B \leq (2^{63} - 1)$).

Формат выходных данных

Выведите YES, если при вычислении произведения $(A \cdot B)$ произойдёт переполнение типа long long. В противном случае выведите NO.

Примеры

стандартный ввод	стандартный вывод
1000000000 1000000000	NO
100000000000 100000000000	YES

Задача В. Уборка снега

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Зима в Ульяновске обычно мягкая, с большим количеством осадков. Вот и на этот раз городской комитет по ЖКХ разрабатывает план уборки снега.

Всего в городе имеются N объектов, с которых нужно убирать снег, — улиц, дворов, площадей и так далее. Изначально ни на одном из объектов нет снега. Согласно нормативам, количество снега на i -м объекте ни в какой момент не должно превышать A_i миллиметров.

В распоряжении комитета по ЖКХ имеются M снегоуборочных машин, каждая из которых способна убрать B миллиметров снега. Назовём критическим тот объект, количество снега на котором ближе всего к максимально допустимому, а если таких объектов несколько — тот из них, который имеет минимальный номер. В начале каждого часа первая машина едет на критический объект и увозит с него снег, затем вторая машина едет на критический объект (он мог измениться или остаться прежним) и увозит с него снег, и так далее. Каждая машина в течение часа может сделать не более одной поездки.

Синоптики прогнозируют, что осадки будут выпадать в течение ближайших K часов; более конкретно, в начале j -го часа на весь город выпадет C_j миллиметров снега. Определите, позволит ли указанная стратегия работы снегоуборочных машин добиться того, чтобы ни на одном из объектов количество снега не превысило норму?

Формат входных данных

Первая строка содержит целое число N ($1 \leq N \leq 10^5$) — количество объектов, на которых осуществляется уборка снега.

Вторая строка содержит N целых чисел A_i ($1 \leq A_i \leq 10^6$) — максимально допустимое количество снега на каждом из объектов.

Третья строка содержит целые числа M и B ($1 \leq M \leq 1000$, $1 \leq B \leq 1000$) — соответственно количество снегоуборочных машин и количество снега, которое может убрать одна машина.

Четвёртая строка содержит целое число K ($1 \leq K \leq 1000$) — количество часов, в течение которых будут выпадать осадки.

Пятая строка содержит K целых чисел C_i ($0 \leq C_i \leq 1000$) — количество снега, выпадающее в каждый из часов.

Формат выходных данных

Если при указанном способе распределения машин ни на одном из объектов количество снега не превысит нормы, выведите число 0.

В противном случае выведите номер часа, на котором норма будет превышена.

Примеры

стандартный ввод	стандартный вывод
5 2 3 2 2 3 3 2 4 1 1 1 1	0
3 3 5 3 1 2 4 1 2 1 2	4

Задача С. Самый сложный предмет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Однажды Макс, едва пережив очередной тяжёлый экзамен, задался вопросом — какая из дисциплин, преподаваемых в университете, является наиболее сложной?

Листая учебный план, Макс выяснил, что для изучения некоторых предметов необходимо предварительно освоить другие предметы. Так, например, чтобы изучать «Технологии программирования», нужно сдать «Основы программирования», перед «Цифровой схемотехникой» следует пройти «Электротехнику и электронику», а для «Высокопроизводительных вычислений» требуются «Алгоритмы и структуры данных» и «Машинно-ориентированное программирование».

Макс предположил, что чем больше у предмета список предварительных требований, тем сложнее этот предмет. Более формально, сложность 0 имеют предметы без предварительных требований, сложность 1 — предметы, основывающиеся хотя бы на одном предмете сложности 0, сложность 2 — предметы, основывающиеся хотя бы на одном предмете сложности 1, и так далее.

Макс уже составил список предварительных требований для всех изучаемых предметов. Помогите ему определить предмет, сложность которого является максимальной.

Формат входных данных

Первая строка содержит целое число N ($1 \leq N \leq 10^4$) — количество учебных предметов.

Вторая строка содержит N различных слов S_i ($1 \leq |S_i| \leq 20$), состоящих из строчных латинских букв и знаков подчёркивания, — названия каждого из предметов.

Третья строка содержит целое число M ($0 \leq M \leq 10^5$) — количество требований.

Следующие M строк описывают предварительные требования. Каждая из них содержит слова A_j и B_j и $(A_j, B_j \in S)$ — соответственно название предмета, который изучается ранее, и название предмета, который изучается позднее.

Формат выходных данных

Выведите названия всех предметов, имеющих максимальную сложность, в алфавитном порядке.

Если сложность хотя бы одного из предметов не может быть определена, выведите Impossible.

Примеры

стандартный ввод
6 algorithms discrete_math higher_math history prog_basics prog_technologies 4 discrete_math algorithms prog_basics prog_technologies prog_basics algorithms higher_math discrete_math
стандартный вывод
algorithms
стандартный ввод
4 physics machine_learning electronics ai_systems 2 physics electronics machine_learning ai_systems
стандартный вывод
ai_systems electronics

Задача D. Масштабирование изображений

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дано чёрно-белое изображение, имеющее высоту H_1 пикселей и ширину W_1 пикселей. Требуется выполнить его масштабирование таким образом, чтобы результирующее изображение имело высоту H_2 пикселей и ширину W_2 пикселей. При этом возможны следующие варианты:

- Если $H_2 = K \cdot H_1$ и $W_2 = K \cdot W_1$, где K — целое, то выполняется увеличение изображения. В этом случае каждый пиксель исходного изображения должен быть заменён на квадрат $K \times K$ из пикселей того же цвета;
- Если $H_1 = K \cdot H_2$ и $W_1 = K \cdot W_2$, где K — целое, то выполняется уменьшение изображения. Эта операция является обратной к описанной выше: каждый квадрат $K \times K$ из пикселей исходного изображения должен быть заменён на один пиксель того же цвета, что и каждый пиксель квадрата. Если при этом квадрат $K \times K$ содержит пиксели разных цветов, то происходит ошибка «Bad image»;
- Если $H_1 = H_2$ и $W_1 = W_2$, то изображение воспроизводится без изменений;
- Наконец, во всех остальных случаях происходит ошибка «Bad size».

Напишите программу, которая позволит масштабировать изображения по указанным правилам.

Формат входных данных

Первая строка содержит целые числа H_1 и W_1 ($1 \leq H_1, W_1 \leq 100$) — соответственно высоту и ширину исходного изображения.

Следующие H_1 строк описывают исходное изображение. Каждая из них содержит W_1 символов '0' или '1' — пиксели изображения.

Следующая строка содержит целые числа H_2 и W_2 ($1 \leq H_2, W_2 \leq 100$) — размеры, которые должны быть получены после масштабирования.

Формат выходных данных

Если масштабирование возможно, выведите H_2 строк, каждая из которых содержит W_2 символов '0' или '1' — пиксели результирующего изображения.

В противном случае выведите текст ошибки.

Примеры

стандартный ввод	стандартный вывод
2 3 001 101 4 6	000011 000011 110011 110011
4 6 000011 001011 110011 110011 2 3	Bad image

Задача Е. Макс и своя игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Макс играет в «Свою игру» против самого Анатолия Вассермана. В игровом раунде знатокам предлагается N различных тем, в каждой из которых имеется M вопросов различной сложности. Самый простой вопрос каждой темы стоит 100 очков, следующий — 200, ..., i -й — $(100 \cdot i)$ очков.

Игра проходит следующим образом. Один из знатоков выбирает любой ещё не озвученный вопрос, ведущий зачитывает его, и первый знаток, нажавший на кнопку, может назвать ответ. Если ответ верен, знаток получает количество очков, равное цене вопроса, и может выбирать следующий вопрос. То, что происходит, когда ответ оказывается неверным, не будет нас интересовать в этой задаче.

Оба знатока — и Макс, и его соперник — настолько эрудированы, что могут правильно ответить на любой вопрос раунда. Макс имеет ещё более важное преимущество — у него безупречная реакция, и он всегда может нажать на кнопку раньше соперника. Прибавьте к этому то, что именно Макс выбирает вопрос первым — и вы поймёте, что его победа практически предрешена.

Тем не менее, интригу в игру вносят специальные вопросы, которые называются «Кот в мешке». Всего в раунде K «Котов в мешке», и заранее неизвестно, какие из вопросов ими являются. Если знаток выбрал подобный вопрос, он не может отвечать сам и обязан передать вопрос сопернику вместе с правом хода.

Макс очень беспокоится за свой результат и всегда продумывает худшие варианты. Помогите ему определить, какое минимальное количество очков он может заработать по результатам раунда.

Формат входных данных

Ввод содержит целые числа N , M и K ($1 \leq N, M \leq 10^5$, $0 \leq K \leq (N \cdot M)$) — соответственно количество тем, количество вопросов в каждой теме и количество «котов в мешке».

Формат выходных данных

Выведите одно целое число — минимальное количество очков, которое может заработать Макс.

Примеры

стандартный ввод	стандартный вывод
6 5 1	8500
1 3 3	100

Задача F. Огненный лабиринт

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Знаменитый искатель приключений Индиана Джонс попал в ловушку! Он оказался заперт в прямоугольной комнате размером $N \times M$, пол которой разделён на квадратные клетки 1×1 . По вертикали клетки комнаты нумеруются от 1 до N сверху вниз, по горизонтали — от 1 до M слева направо.

На каждой из клеток пола нарисована цифра от 1 до 9. Индиана Джонс моментально сообразил, что эти цифры означают не что иное, как период срабатывания встроенных в пол горелок: каждую секунду, номер которой делится на изображённое на клетке число, из отверстий в полу бьют мощные языки пламени, причиняющие одну единицу урона, если на них наступить.

Изначально (в первую секунду) Индиана Джонс стоит в клетке $(1; 1)$. В каждую следующую секунду он может либо остаться стоять в текущей клетке, либо перешагнуть в любую из клеток, имеющих общую сторону с текущей. В клетке $(N; M)$ проделан люк, ведущий на свободу.

Цель Индианы Джонса — добраться до выхода, получив по пути минимальное количество урона. Но ему надо действовать быстро, так как по прошествии T секунд вся комната самоуничтожится.

Подскажите ему, как выбраться из ловушки с наименьшими потерями.

Формат входных данных

Первая строка содержит целые числа N , M и T ($1 \leq N, M \leq 50$, $1 \leq T \leq 400$) — соответственно размеры комнаты и количество секунд, оставшееся до её разрушения.

Следующие N строк описывают клетки комнаты. i -я из них содержит строку из M цифр A_{ij} ($1 \leq A_{ij} \leq 9$), j -я из которых равна периоду срабатывания горелок в клетке $(i; j)$.

Формат выходных данных

Выведите одно целое число — минимальное количество единиц урона, которое может получить Индиана Джонс, оказавшись в клетке с выходом не позже секунды T . Обратите внимание, что Индиана Джонс может получить урон и в стартовой клетке, и в клетке с выходом.

Если Индиана Джонс не успеет добраться до выхода, выведите число -1.

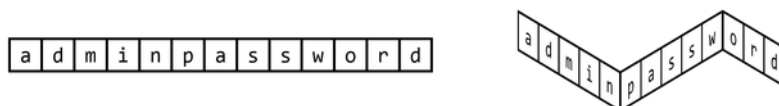
Примеры

стандартный ввод	стандартный вывод
3 4 8 3224 3243 1131	1
3 4 6 3224 3243 1131	2

Задача G. Пароль в снегу

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Перед ответственной олимпиадой Макс установил новый пароль для входа в тестирующую систему, состоящий из N символов. Этот пароль Макс записал на бумажке, а бумажку сложил гармошкой ширины M , как показано на рисунке, и положил в карман.



В день олимпиады Макс очень торопился, поэтому в какой-то момент сложенная бумажка выпала из кармана и упала в сугроб, так что K левых символов оказались в снегу. Когда Макс поднял и развернул бумажку, оказалось, что все попавшие в снег символы расплылись и стали нечитаемыми.



Но ещё не всё потеряно — определённую часть пароля Макс уже выучил наизусть. Тем не менее, некоторые символы он не запомнил и хочет выяснить, попали они в снег или нет. Помогите ему найти ответ на этот вопрос.

Формат входных данных

Первая строка содержит целые числа N , M и K ($1 \leq K \leq M \leq N \leq 10^{18}$) — соответственно длину пароля, длину сложенной гармошкой бумажки и количество левых символов сложенной бумажки, попавших в снег.

Вторая строка содержит целое число Q ($1 \leq Q \leq 10^5$) — количество запросов.

Третья строка содержит Q целых чисел A_i ($1 \leq A_i \leq N$) — номера символов, для которых Макс хочет узнать, оказались они в снегу или нет.

Формат выходных данных

Для каждого из номеров A_i выведите, не разделяя пробелами, 0, если соответствующий символ не попал в снег, либо 1, если попал.

Примеры

стандартный ввод	стандартный вывод
13 5 2 8 1 3 7 8 9 10 11 13	10001110
20 4 2 10 15 5 6 1 18 12 4 2 9 13	1001100110

Задача Н. Турнирная таблица

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Во время проведения соревнований по программированию тестирующая система хранит информацию обо всех отправленных на проверку решениях и автоматически формирует турнирную таблицу, которая показывает, какое место занимает каждый из участников. В этой задаче вам нужно помочь тестирующей системе и самостоятельно составить турнирную таблицу по имеющимся данным.

Тестирующая система хранит информацию о решениях в записях формата $T_i S_i P_i R_i$, где

- T_i — время отправки i -го решения от момента начала соревнования, указанное в минутах;
- S_i — имя участника, отправившего i -е решение;
- P_i — идентификатор (буква) задачи, для которой отправлено i -е решение;
- R_i — результат проверки i -го решения: '+' — решение засчитано, '-' — решение не засчитано.

По данной информации формируется содержание турнирной таблицы в соответствии со следующими правилами:

- Первый столбец таблицы имеет заголовок «R» и содержит места, занятые участниками:
 - Более высокое место занимает тот участник, который решил больше задач;
 - Если двое или более участников решили одинаковое количество задач, более высокое место занимает участник, имеющий меньшее штрафное время;
 - Если двое или более участников решили одинаковое количество задач и имеют одинаковое штрафное время, более высокое место занимает участник, раньше остальных отправивший последнее верное решение;
 - Если с учётом всех вышеуказанных критериев двое или более участников всё ещё имеют одинаковый результат, то они все занимают одинаковое место. При этом, если X участников заняли одинаковое место Y , то следующий за ними участник занимает место $(Y + X)$.
 - У участников, не решивших ни одной задачи, место не отображается, а сами они следуют в таблице в алфавитном порядке имён.
- Второй столбец таблицы имеет заголовок «Contestant» и содержит имена участников.
- Следующие столбцы таблицы имеют заголовки «A», «B» и так далее и содержат информацию об отправленных решениях по задачам:
 - Если участник решил задачу с первой попытки, то соответствующая ячейка должна содержать символ '+';
 - Если участник решил задачу после K неправильных попыток, то соответствующая ячейка должна содержать символ '+', за которым следует число K ;
 - Если участник не решил задачу, но отправил K неправильных попыток, то соответствующая ячейка должна содержать символ '-', за которым следует число K ;
 - В любом из упомянутых выше случаев ниже в ячейке должно располагаться время отправки последнего решения по задаче, записанное в минимальном подходящем формате — HH:MM, HHH:MM либо HHHH:MM;
 - Решения задачи, отправленные участником после первого засчитанного решения, не учитываются при составлении турнирной таблицы.

- Предпоследний столбец таблицы имеет заголовок «+» и содержит количество задач, решённых каждым из участников.
- Последний столбец таблицы имеет заголовок «Т» и содержит штрафное время каждого из участников. Штрафное время определяется как сумма минут (значений T_i), прошедших от начала соревнования до моментов отправки каждого засчитанного решения, плюс 20 минут за каждое ошибочное решение по каждой из решённых задач;
- Строки таблицы сортируются в порядке неубывания мест, занимаемых соответствующими участниками. Строки, соответствующие участникам, занявшим одинаковые места, сортируются в алфавитном порядке имён участников.

Форматирование турнирной таблицы осуществляется в соответствии со следующими правилами:

- Каждая строка таблицы должна иметь верхнюю и нижнюю границы, составленные из символов '#'. Высота строки заголовка должна равняться одному символу, всех остальных строк — двум символам;
- Каждая ячейка таблицы должна иметь левую и правую границы, составленные из символов '#'. Содержимое всех ячеек должно отделяться от левой и правой границ не менее чем одним пробелом. Ширина всех ячеек одного столбца должна быть одинаковой и минимально возможной (но различные столбцы могут иметь различную ширину);
- Содержимое первого, предпоследнего и последнего столбцов таблицы (кроме заголовков) выравнивается по правому краю соответствующих ячеек;
- Содержимое второго столбца таблицы (кроме заголовка) выравнивается по левому краю соответствующих ячеек;
- Заголовки столбцов таблицы, а также содержимое всех остальных столбцов выравниваются по центру соответствующих ячеек. Если при этом количество пробелов в ячейке нечётно, то пробелов слева должно быть на один меньше, чем справа.

Составьте турнирную таблицу, соблюдая все описанные требования.

Формат входных данных

Первая строка содержит целые числа N и M ($1 \leq N \leq 26$, $1 \leq M \leq 10^5$) — число задач и число отправленных решений.

Следующие N строк описывают решения. Каждая из них содержит число T_i , строку S_i , символы P_i и R_i ($0 \leq T_i \leq 10^5$, $1 \leq |S_i| \leq 20$, 'A' $\leq P_i \leq 'Z'$, $P_i \in \{'+', '-'\}$) — время отправки решения, имя участника, отправившего решение, идентификатор задачи и результат проверки.

Имена участников состоят только из строчных латинских букв и знаков подчёркивания. Идентификаторы задач принадлежат множеству первых N заглавных букв латинского алфавита. Строки, описывающие решения, упорядочены в порядке неубывания значений T_i .

Формат выходных данных

Выведите турнирную таблицу, составленную и оформленную в соответствии с правилами (см. примеры).

Примеры

стандартный ввод	стандартный вывод
4 21 15 rivest A - 47 cormen B - 52 rivest B + 118 stein B - 122 cormen A + 152 cormen C - 267 leiserson B - 320 stein D - 349 cormen B - 366 stein B - 366 leiserson C + 367 leiserson A + 392 rivest A - 487 stein D - 499 cormen B + 567 stein B - 599 stein B + 617 cormen D + 7028 cormen C - 9064 rivest A + 10026 stein D -	##### # R # Contestant # A # B # C # D # + # T # ##### # 1 # cormen # + # +2 # -2 # + # 3 # 1278 # # # # 02:02 # 08:19 # 117:08 # 10:17 # # # ##### # 2 # leiserson # + # -1 # + # # 2 # 733 # # # # 06:07 # 04:27 # 06:06 # # # # ##### # 3 # rivest # +2 # + # # # 2 # 9156 # # # # 151:04 # 00:52 # # # # # ##### # 4 # stein # # +3 # # -3 # 1 # 659 # # # # # 09:59 # # 167:06 # # # #####
2 6 5 korotkevich A - 7 korotkevich B - 12 korotkevich A + 14 mitrichev A - 20 korotkevich B - 26 mitrichev A -	##### # R # Contestant # A # B # + # T # ##### # 1 # korotkevich # +1 # -2 # 1 # 32 # # # # 00:12 # 00:20 # # # ##### # # mitrichev # -2 # # 0 # 0 # # # # 00:26 # # # # #####

Замечание

В системе Contester вы можете перейти к просмотру турнирной таблицы по ссылке, расположенной в левой части страницы с задачей.