

Задача А. Снова ИИ

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Стёпа и Катя работают в Яндексе и создают новый искусственный интеллект для научно-исследовательских целей. ИИ работает с целыми числами, особенно с числами от 1 до N . Раскрыть предназначение ИИ они пока не готовы, но известно, что для него важно свойство *обратимости* чисел: так они называют тот факт, что $N - (N - X) = X$ для всех X от 1 до N .

Вы решили на шаг опередить разработки Яндекса и создать аналогичный ИИ, который тоже будет работать с числами от 1 до N , но будет сосредоточен на операциях умножения и деления чисел. Для этого вы определили своё свойство *суперобратимости*. Будем говорить, что целое число X ($1 \leq X \leq N$) *суперобратимо* для N , если $N \operatorname{div} (N \operatorname{div} X) = X$, где div — операция целочисленного деления с округлением вниз.

К сожалению, не все числа оказались суперобратимы. Но нельзя терять время! Нужно скорее продолжить работу, а для этого нужно определить, сколько целых чисел от 1 до N являются суперобратимыми. Возможно, придётся даже перебрать несколько значений N_i , пока вы не найдёте подходящее.

Формат входных данных

В первой строке дано целое число Q — количество чисел N_i , которые вы планируете рассмотреть ($1 \leq Q \leq 10$).

Далее идут Q строк. В каждой строке дано целое число N_i ($1 \leq N_i \leq 10^{18}$).

Формат выходных данных

Выведите Q строк. В i -й строке должно быть записано единственное целое число — количество суперобратимых чисел для N_i .

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--------------------|-----------------------|
| 1 | 34 | $N_i \leq 10^5$ | |
| 2 | 34 | $N_i \leq 10^9$ | 1 |
| 3 | 32 | $N_i \leq 10^{18}$ | 1, 2 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 | 1 |
| 1 | 2 |
| 2 | 2 |
| 3 | 6 |
| 12 | |

Задача В. Как пройти в столовую

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

Петя поступил на ФИИТ УрФУ и всё ещё не запомнил, как ориентироваться в учебном корпусе — матмехе. После второй пары Петя бесцельно слоняется по коридорам, которые представляют из себя сплошной лабиринт. Корпус имеет форму прямоугольника ширины W и высоты H , разделённого на клетки 1×1 . Клетки бывают четырёх типов:

- Пустая клетка;
- Клетка со стеной;
- Клетка с ключом;
- Клетка с запертой дверью.

Как назло уборщицы регулярно начинают мыть пол, поэтому если Петя начнёт двигаться в одном направлении, то он не остановится, пока не упрётся в препятствие. Препятствиями являются граница корпуса, стена, а также запертая дверь, если у Пети нет ключа.

По пути Петя может собирать ключи. Проходя мимо ключа, Петя моментально подбирает его, не останавливаясь. Он может носить с собой неограниченное количество ключей. Любой ключ подходит к любой двери. Если Петя упирается в запертую дверь, он моментально открывает её ключом и продолжает движение без остановки. Ключ остается в замочной скважине двери. Если же у Пети нет с собой ни одного ключа, то запертые двери для него не отличаются от стен.

Также известно, что во время пар декан открывает все двери. Поэтому во время пар на матмехе нет ни одной запертой двери и ни одного ключа. Во время перемены можно найти ключи и/или запертые двери.

Помогите Пете не заблудиться и выбраться в столовую. Он знает, в какой клетке он начал свой путь, а также всю историю своих передвижений. Он сделал L шагов, каждый из шагов — перемещение в одном из четырех направлений: вверх, вправо, вниз или влево. Для каждого из шагов выведите, сколько клеток лабиринта он проскользил в этом направлении.

Формат входных данных

В первой строке даны три целых числа H , W и D — высота корпуса, ширина корпуса, а также число D , обозначающее время; если $D = 0$, то дело происходит во время пары, иначе — на перемене ($1 \leq H, W \leq 10^5$, $H \cdot W \leq 10^5$, $0 \leq D \leq 1$).

Далее идут H строк, описывающих корпус матмеха. Каждая строка состоит из W символов, не разделённых пробелами. Всего есть 5 возможных символов. Символ «.» (код 46) обозначает пустую клетку. Символ «#» (код 35) обозначает стену. Символ «К» обозначает клетку с ключом. Символ «L» обозначает клетку с запертой дверью. Символ «S» обозначает клетку, где начинает Петя — эта клетка считается пустой. Гарантируется, что есть ровно один символ «S».

Гарантируется, что если $D = 0$, то символы «К» и «L» в описании корпуса отсутствуют. Если $D = 1$, то есть хотя бы один символ «К» или «L».

В следующей строке дано целое число L — количество шагов, сделанных Петей ($1 \leq L \leq 10^5$).

В последней строке даны L символов, не разделённых пробелами. Всего есть 4 возможных символа, которые могут встречаться: «U», «R», «D», «L». Они обозначают направления вверх, вправо, вниз и влево соответственно. Корпус ориентирован так, как дан во вводе: первая строка является самой верхней, первый столбец является самым левым.

Формат выходных данных

Выведите L строк, в каждой строке выведите одно целое число. В i -й строке выведите количество клеток, которые Петя прошёл, сделав i -й шаг.

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 25 | $H \cdot W \leq 100, L \leq 100, D = 0$ | |
| 2 | 25 | $H \cdot W \leq 100, L \leq 100, D \leq 1$ | 1 |
| 3 | 25 | $H \cdot W \leq 10^5, L \leq 10^5, D = 0$ | 1 |
| 4 | 25 | $H \cdot W \leq 10^5, L \leq 10^5, D \leq 1$ | 1, 2, 3 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 4 0 | 2 |
| S..# | 2 |
| | 1 |
| #... | 1 |
| 6 | 3 |
| RDRULD | 0 |
| 1 6 1 | 2 |
| LKSKLL | 0 |
| 4 | 4 |
| RRLR | 4 |

Задача С. Baba is You

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 256 мегабайт |

Вы отправились исследовать таинственный мир, заполненный различными существами и предметами. Поскольку многие объекты в этом мире не были открыты ранее, вы дали каждому объекту название. Все названия состоят из заглавных латинских букв и имеют длину от 1 до 4 символов.



В ходе наблюдений вы заметили, что объекты со временем меняются. Это вы решили записать в журнал с помощью *правил*. Всего получилось N правил. Правило номер i гласит, что объект A_i со временем превратится в объект B_i . Такое правило записывается как « A_i IS B_i ». Заметим, что объекты A_i и B_i могут совпадать, в этом случае превращение всё равно происходит.

Сегодня вы вернулись в исследовательский центр и решили заняться теоретическими экспериментами. Вам дали план, состоящий из Q запросов. У вас нет при себе экземпляров изученных объектов, поэтому при выполнении запросов приходится опираться на записи в журнале. Каждый запрос задаётся одним из двух способов:

1. «? C_j ». Чтобы выполнить этот запрос, нужно провести эксперимент и определить, во что в конечном итоге превратится объект C_j согласно записанным в журнале правилам.
2. «+ C_j IS D_j ». Чтобы выполнить этот запрос, нужно добавить новое правило в журнал, согласно которому объект C_j превращается в объект D_j . Новое правило не отличается от старых, и оно начинает работать для всех последующих запросов.

Важная часть любой исследовательской деятельности — ведение отчёта об ошибках. В ходе проведения эксперимента может произойти ошибка одного из двух типов:

- «INFINITE LOOP»: эта ошибка происходит в том случае, если цепочка превращений объекта бесконечна, то есть объект будет превращаться в другой объект сколько угодно раз. Ошибка не возникает, если объект превращается сам в себя.
- «TOO COMPLEX»: эта ошибка происходит в том случае, если согласно записанным правилам объект одновременно должен превратиться в два разных объекта. При возникновении этой ошибки эксперимент нужно тут же прекратить. Ошибка не возникает, если объект превращается в один и тот же объект по двум различным правилам. Но ошибка возникает, если объект должен превратиться в себя и в другой объект.

Попробуйте для каждого запроса первого вида определить, во что превратится данный объект или какая ошибка возникнет в результате выполнения запроса. Обратите внимание, что в одном запросе не могут произойти две ошибки сразу.

Формат входных данных

В первой строке дано целое число N — изначальное количество правил ($1 \leq N \leq 50000$).

Далее идут N строк. Строка номер i имеет вид « A_i IS B_i » (без кавычек) и описывает правило, согласно которому объект A_i превращается в объект B_i .

В следующей строке дано целое число Q — количество запросов ($1 \leq Q \leq 50000$).

Далее идут Q строк. Строка номер j имеет вид « $? C_j$ » или « $+ C_j$ IS D_j » (без кавычек) и описывает запрос первого или второго вида соответственно. Запросы выполняются в том порядке, в котором они даны во входных данных. Гарантируется, что существует хотя бы один запрос первого вида.

Все объекты названы строками длины от 1 до 4, состоящими из заглавных латинских букв. Гарантируется, что не существует объекта «IS». Также в описании правил встречается слово «IS», написанное заглавными латинскими буквами, а в описаниях запросов встречаются символы «?» (код 63) и «+» (код 43).

Формат выходных данных

Для каждого запроса первого вида, если ошибка не возникает, выведите в отдельной строке название объекта, в который превратится данный объект. Если ошибка возникает, выведите в отдельной строке «INFINITE LOOP» для ошибки первого типа или «TOO COMPLEX» для ошибки второго типа (в обоих случаях без кавычек).

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|---|-----------------------|
| 1 | 20 | $N, Q \leq 50$, нет запросов на добавление, нет ошибок | |
| 2 | 18 | $N, Q \leq 50$ | 1 |
| 3 | 20 | $N, Q \leq 1000$ | 1, 2 |
| 4 | 22 | $N, Q \leq 50000$, нет ошибок | 1 |
| 5 | 20 | $N, Q \leq 50000$ | 1, 2, 3, 4 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Обратите внимание на необычные ограничения в подзадачах 1 и 4. В них гарантируется, что при выполнении превращений не возникает ошибок. В подзадаче 1 также гарантируется, что нет запросов второго вида.

Пример

| стандартный ввод | стандартный вывод |
|--|--|
| 3 BABA IS YOU BOX IS KEY KEY IS BOX | YOU YOU INFINITE LOOP TOO COMPLEX |
| 8 + BABA IS YOU ? BABA + YOU IS YOU ? BABA ? BOX + BOX IS BOX ? BOX ? LOVE | LOVE |

Задача D. Он вам не ферзь

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1.5 секунд |
| Ограничение по памяти: | 256 мегабайт |

В коридорах матмеха, где учатся студенты ФИИТ, есть много всего интересного: рояль на потолке, Хогвартс, оптические иллюзии на полу, качели, даже портал на платформу $9\frac{3}{4}$. А ещё есть шахматы на стене, в которые любят играть студенты Артемий и Михей. Во время каникул на матмехе начался ремонт, поэтому доску сняли.

Ребята не растерялись и решили нарисовать шахматы мелом на стене. За каникулы правила игры они подзабыли. Начертив мелом доску $M \times M$, они начали вспоминать, как должны быть расположены фигуры. Михей недолго думал и нарисовал на поле N белых королей — король ведь лучшая фигура! Артемий разбирался в шахматах побольше, и он знал, что лучшая фигура — это ферзь. Поэтому Артемий решил нарисовать на свободной клетке одного чёрного ферзя, который поставил бы под бой всех белых королей и при этом сам не был бы под боем.

По авторитетному мнению Михея, король может бить 8 смежных клеток, имеющих общую сторону или общий угол с той клеткой, на которой стоит сам король. Ферзь же полностью бьёт горизонталь, вертикали и обе диагонали, на которых стоит. При этом наши герои забыли, что ферзь не может прыгать через фигуры: по их мнению, если два короля стоят на одной линии и по одну сторону от ферзя, ферзь может бить их обоих.

Всего они провели T игр, каждый раз заново рисуя доску и фигуры. Зная, как Михей располагал своих королей, посчитайте количество позиций, куда Артемий мог поставить ферзя так, чтобы он не стоял под боем ни одного из королей и при этом все короли стояли под его боем.

Формат входных данных

В первой строке дано целое число T — количество игр ($1 \leq T \leq 10$).

Далее идут T блоков, каждый блок описывает очередную игру.

В первой строке блока даны два целых числа M и N — длина стороны поля и количество королей ($1 \leq M \leq 10^9$, $1 \leq N \leq 10^5$). Гарантируется, что Михей и Артемий сыграли не более одной игры с $N > 5$.

Далее идут N строк, в i -й из них записаны два целых числа x_i , y_i — номер столбца и номер строки, на которых стоит i -й белый король ($1 \leq x_i, y_i \leq M$). Гарантируется, что в каждой клетке стоит не более одного короля.

Формат выходных данных

Выведите T строк. В каждой строке выведите единственное целое число — количество позиций, куда Артемий может поставить чёрного ферзя в очередной игре.

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|----------------------------|-----------------------|
| 1 | 14 | $M \leq 100, N = 1$ | |
| 2 | 19 | $M, N \leq 100$ | 1 |
| 3 | 20 | $M, N \leq 1000$ | 1, 2 |
| 4 | 28 | $M, N \leq 10^5$ | 1, 2, 3 |
| 5 | 19 | $M \leq 10^9, N \leq 10^5$ | 1, 2, 3, 4 |

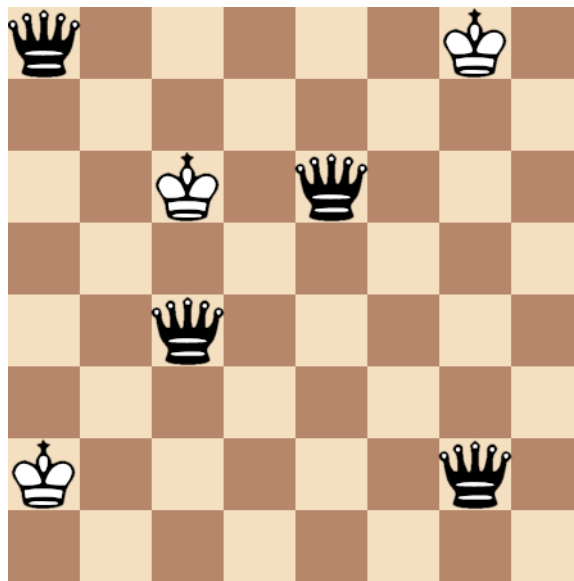
Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 | 4 |
| 8 3 | 0 |
| 1 2 | |
| 3 6 | |
| 7 8 | |
| 2 4 | |
| 1 1 | |
| 1 2 | |
| 2 1 | |
| 2 2 | |

Замечание

Иллюстрация к первой игре из примера, одновременно показаны все возможные позиции ферзя:



Задача Е. Сила есть, ум тоже нужен

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

В компании NAUMEN очень ценятся личные качества сотрудников. Основными личными качествами являются сила и ум. Недавно компания провела тестирование всех своих сотрудников, в результате которого были определены показатели силы и ума. По мнению исследователей, эти параметры позволят объективно оценить работоспособность каждого сотрудника.

Всего в компании N сотрудников, пронумерованных от 1 до N . Компания имеет иерархическую структуру. Чем меньше номер сотрудника, тем он главнее. Сотрудник номер 1 является директором, у него нет начальника. У всех остальных сотрудников есть начальник: у сотрудника номер i начальником является сотрудник номер P_i ($P_i < i$). Также принято считать, что $P_1 = 0$.

В результате исследования было определено, что i -й сотрудник имеет силу A_i и ум B_i . Эти показатели важны при решении проблем. Исследователи компании NAUMEN предложили следующую математическую модель проблемы: по их мнению, проблема — это пара натуральных чисел (x, y) . Первое число описывает физическую сложность проблемы, второе число — умственную. Считается, что сотрудник номер i может самостоятельно решить проблему (x, y) , если $A_i \geq x$ и $B_i \geq y$. Таким образом, учитываются как сила, так и ум.

Конечно, сотрудники не всегда решают проблемы поодиночке. Если у сотрудника в подчинении есть люди, он может делегировать любому своему непосредственному подчинённому эту проблему. Тот сотрудник, кому проблема была делегирована, может в свою очередь делегировать её одному из своих подчинённых, и так далее. Таким образом, если хотя бы один из подчинённых i -го сотрудника может решить проблему, то считается, что и i -й сотрудник может её решить.

Вам было поручено подведение итогов исследования. Для этого нужно для каждого сотрудника посчитать, сколько различных проблем он умеет решать. Обратите внимание, что раз параметры x и y в любой проблеме (x, y) — натуральные числа, то множество различных проблем, которые умеет решать сотрудник, всегда конечно.

Формат входных данных

В первой строке дано целое число N — количество сотрудников в компании NAUMEN ($1 \leq N \leq 50000$).

Далее идут N строк, описывающих сотрудников. В i -й строке даны три целых числа P_i, A_i, B_i — номер непосредственного начальника, показатель силы и показатель ума i -го сотрудника ($0 \leq P_i < i$, $1 \leq A_i, B_i \leq 10^6$; $P_i = 0$ только при $i = 1$).

Формат выходных данных

Выведите N строк. В i -й строке выведите одно целое число — количество различных проблем, которые умеет решать i -й сотрудник.

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 21 | $N \leq 5, A_i \leq 10, B_i \leq 10$ | |
| 2 | 26 | $N \leq 50, A_i \leq 100, B_i \leq 10^6$ | 1 |
| 3 | 23 | $N \leq 5000, A_i \leq 100, B_i \leq 10^6$ | 1, 2 |
| 4 | 14 | $N \leq 500, A_i \leq 10^6, B_i \leq 10^6$ | 1, 2 |
| 5 | 16 | $N \leq 50000, A_i \leq 10^6, B_i \leq 10^6$ | 1, 2, 3, 4 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 | 30 |
| 0 5 5 | 10 |
| 1 1 10 | 12 |
| 1 4 3 | |

Задача F. Трудный выбор

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Ваня учится на ФИИТ в УрФУ, где каждый семестр студенты делают командные проекты. У Вани много друзей в группе, но у каждого свои требования. Например, если позвать в команду Петю, то должен прийти и Алёша, иначе Петя не придёт. Ваня по данным требованиям смог выяснить, что у него есть целых N вариантов, кого пригласить. Каждый из способов он описал в виде величины A_i — количество друзей в выборке под номером i .

Но это было полбеды! Ведь после того, как он с друзьями соберёт команду, нужно будет выбрать того, кто будет представлять проект на защите. Все ребята любят программировать, и никто не хочет заниматься презентацией, поэтому они договорились, что защищает проект тот, кого выберут по считалочке.

Ваня, будучи организатором этого выбора, должен проводить этот расчёт. Чтобы выбрать ответственного по считалочке, он выстраивает людей по кругу и начинает считать от себя, считая по одному слову на человека. На кого попадёт последнее слово, тот и защищает проект. Иван знает Q считалочек. Считалочка номер j состоит из B_j слов. Он решил узнать, насколько удачной будет каждая из них. Для этого он ввёл понятие *хорошей* компании одногруппников — таковой она является, если в ней он не будет защищать проект при использовании данной считалочки.

Ваня хочет уже начать писать код для проекта вместо поиска идеальной считалочки, поэтому он обратился к вам, чтобы вы помогли ему вычислить ответ. Ему нужно для каждой считалочки найти минимальный номер *хорошей* группы.

Формат входных данных

В первой строке задано целое число N — количество групп из друзей Вани ($1 \leq N \leq 10^5$).

Во второй строке записаны N целых чисел A_i — размеры i -й группы ($1 \leq A_i \leq 10^7$).

В третьей строке дано целое число Q — количество считалочек ($1 \leq Q \leq 10^6$).

В следующих Q строках записано по одному целому числу B_j — количество слов в j -й считалочке ($1 \leq B_j \leq 10^7$).

Формат выходных данных

Выведите Q строк, где j -я строка содержит одно целое число — минимальный номер *хорошей* группы для считалочки под номером j или -1 , если таковой нет.

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 10 | $N, Q \leq 100; A_i, B_j \leq 10^3$ | |
| 2 | 30 | $N, Q \leq 10^3; A_i, B_j \leq 10^6$ | 1 |
| 3 | 16 | $N \leq 5000, Q \leq 10^5; A_i, B_j \leq 10^6$ | 1, 2 |
| 4 | 16 | $N \leq 10^5, Q \leq 5000; A_i, B_j \leq 10^6$ | 1, 2 |
| 5 | 15 | $N, Q \leq 10^5; A_i, B_j \leq 10^6$ | 1, 2, 3, 4 |
| 6 | 13 | $N \leq 10^5, Q \leq 10^6; A_i, B_j \leq 10^7$ | 1, 2, 3, 4, 5 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 | 2 |
| 1 2 5 | -1 |
| 3 | 1 |
| 3 | |
| 7 | |
| 12 | |

Замечание

Объяснение примера:

Пусть первая считалочка будет «Выступать будешь ты».

Если мы возьмём первую группу, то Ваня и Вася будут стоять по кругу. Считалочка будет воспроизводиться следующим образом: («Выступать», Ваня), («будешь», Вася), («ты», Ваня). Т.е. Ваня будет выступать с докладом.

Если возьмём вторую группу, то у нас будет круг: Ваня, Валя, Витя. В таком случае будет следующая ситуация: («Выступать», Ваня), («будешь», Валя), («ты», Витя) — в этом случае выступает не Ваня, а значит группа *хорошая* и она будет ответом.

Для второй считалочки для всех трёх групп в конце будет Ваня, а значит ответ -1 .

Для третьей считалочки первая группа уже *хорошая*.

Задача G. Ulearn

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Паша Вилкин, как и остальные студенты ФИИТ, проходит много разных курсов на платформе Ulearn.me. В последнее время у него скопилось много задачек на программирование, которые нужно успеть решить к дедлайну. Чтобы написать качественные решения и уложиться в срок, нужно пройти 4 этапа:

1. Написать код;
2. Отправить код на проверку автотестами;
3. Внести правки по результатам тестирования;
4. Отправить решение на код-ревью.

Первый и третий этапы требуют присутствия Паши и занимают время A . Для второго и четвертого этапов присутствие Паши не требуется, каждый из них занимает время B . Чтобы решение получилось качественным, весь процесс должен идти без остановки. Помогите Паше рассчитать минимальное время, необходимое для решения N задач.

Формат входных данных

В первой строке дано целое число N ($1 \leq N \leq 10^9$).
Во второй строке дано целое число A ($1 \leq A \leq 10^5$).
В третьей строке дано целое число B ($1 \leq B \leq 10^5$).

Формат выходных данных

Выведите единственное целое число — минимальное необходимое время для решения всех задач.

Система оценки

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--------------------------|-----------------------|
| 1 | 17 | $B < A, N \leq 10^5$ | |
| 2 | 15 | $B \leq A, N \leq 10^5$ | 1 |
| 3 | 16 | $B \leq 2A, N \leq 10^5$ | 1, 2 |
| 4 | 26 | $N \leq 10^5$ | 1, 2, 3 |
| 5 | 26 | $N \leq 10^9$ | 1, 2, 3, 4 |

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 3 1 | 15 |
| 2 2 2 | 10 |

Замечание

Иллюстрация к первому примеру:

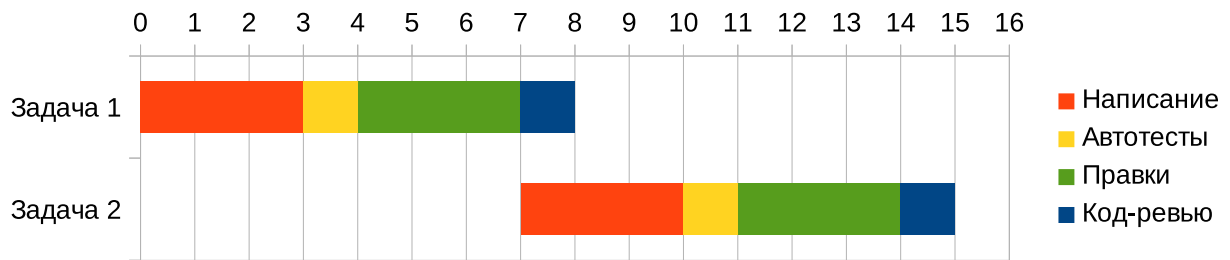


Иллюстрация ко второму примеру:

